CLOUDY WITH A CHANCE OF STARLIGHT:

COUPLING OF SMOOTHED PARTICLE HYDRODYNAMICS AND MONTE CARLO RADIATIVE TRANSFER FOR THE STUDY OF IONISING STELLAR FEEDBACK

Maya Atanasova Petkova

A Thesis Submitted for the Degree of PhD
at the
University of St Andrews

2018

Cloudy with a Chance of Starlight:

Coupling of Smoothed Particle Hydrodynamics and Monte Carlo Radiative Transfer for the Study of Ionising Stellar Feedback

Maya Atanasova Petkova

University of
St Andrews

This thesis is submitted in partial fulfilment for the degree of

Doctor of Philosophy (PhD)

at the University of St Andrews

July 2018

Candidate's declaration

I, Maya Atanasova Petkova, do hereby certify that this thesis, submitted for the degree of PhD, which is approximately 27,000 words in length, has been written by me, and that it is the record of work carried out by me, or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for any degree.

I was admitted as a research student at the University of St Andrews in September 2014.

I received funding from an organisation or institution and have acknowledged the funder(s) in the full text of my thesis.

Date Signature of candidate

Supervisor's declaration

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of PhD in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

Date Signature of supervisor

Permission for publication

In submitting this thesis to the University of St Andrews we understand that we are giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. We also understand, unless exempt by an award of an embargo as requested below, that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that this thesis will be electronically accessible for personal or research use and that the library has the right to migrate this thesis into new electronic forms as required to ensure continued access to the thesis.

I, Maya Atanasova Petkova, have obtained, or am in the process of obtaining, third-party copyright permissions that are required or have requested the appropriate embargo below.

The following is an agreed request by candidate and supervisor regarding the publication of this thesis:

Printed copy

No embargo on print copy.

Electronic copy

No embargo on electronic copy.

Date	Signature of candidate
Date	Signature of supervisor

Underpinning Research Data or Digital Outputs

Candidate's declaration

I, Maya Atanasova Petkova, understand that by declaring that I have original research data or digital outputs, I should make every effort in meeting the University's and research funders' requirements on the deposit and sharing of research data or research digital outputs.

Date	Signature of candidate

Permission for publication of underpinning research data or digital outputs

We understand that for any original research data or digital outputs which are deposited, we are giving permission for them to be made available for use in accordance with the requirements of the University and research funders, for the time being in force.

We also understand that the title and the description will be published, and that the underpinning research data or digital outputs will be electronically accessible for use in accordance with the license specified at the point of deposit, unless exempt by award of an embargo as requested below.

The following is an agreed request by candidate and supervisor regarding the publication of underpinning research data or digital outputs:

No embargo on underpinning research data or digital outputs.

Date	Signature of candidate
Date	Signature of supervisor

Abstract

Ionising radiation is present in a variety of astrophysical problems, and it is particularly important for shaping the process of star formation in molecular clouds, containing hot, high-mass stars. In order to account for the effects of ionising radiation within numerical models of star formation, we need to combine a hydrodynamics method with a radiative transfer method and obtain a radiation hydrodynamics scheme (RHD). In this thesis I achieve live radiation hydrodynamics by coupling the Smoothed Particle Hydrodynamics (SPH) code PHANTOM with the Monte Carlo Radiative Transfer (MCRT) code CMACIONIZE. Since SPH is particle-based and MCRT is grid-based, I construct an unstructured, Voronoi grid in order to establish a link between the two codes. In areas with large density gradients, a Voronoi grid based purely on the SPH particle positions achieves insufficient resolution, and therefore I propose a novel algorithm for inserting a small number of additional grid cells to improve the local resolution. Furthermore, the MCRT calculations require the knowledge of an average density for each Voronoi cell. To address this, I develop an analytic density mapping from SPH to a Voronoi grid, by deriving an expression for the integrals of a series of kernel functions over the volume of a random polyhedron. Finally, I demonstrate the validity of the live RHD through the benchmark test of D-type expansion of an H II region, where good agreement is shown with the existing literature. The RHD implementation is then used to perform a proof-of-concept simulation of a collapsing cloud, which produces high-mass stars and is subsequently partially ionised by them. The presented code is a valuable tool for future star formation studies, and it can be used for modelling a broad range of additional astronomical problems involving ionising radiation and hydrodynamics.

Acknowledgements

There are many people without whom the path to the end of my PhD would have been much harder to walk. Looking back I can recall many acts of help, kindness and guidance for which I feel grateful.

The first wave of gratitude is for my astronomy mentors from my home town in Bulgaria. I would like to thank Veselka Radeva, Dimitar Slavov, Eva Bojurova and Zahari Donchev for the many hours of wonder and shared passion for the night sky and the Universe.

During my time in St Andrews I have received a lot of academic support and encouragement. I thank Ian Bonnell for being my PhD supervisor and for helping me find my own research direction. I would also like to thank Bert, Guillaume, Duncan, Will and Gerardo for the many helpful conversations that we have had, as well as Daniel Price and Jim Dale for their guidance. I am especially grateful to Max who has been my go-to person for anything PhD, and who has been beside me at every step of the way.

Outside of my work, I thank my parents for supporting me and trusting me to make my own decisions, regardless of how otherworldly they may have seemed at times. And I thank baba Dori who taught me to always try to do my best. I thank Plamen for being a steady presence of encouragement, and Dan who provided loving care through times of doubt and internal pressure. Also, I would like to thank Inna for the many late night conversations over the years.

Finally, I would like to thank everybody from STAARTango for being part of my life. My days and nights would have been much duller without you!

This thesis received funding from the European Research Council for the FP7 ERC advanced grant project ECOGAL. The presented work used the DiRAC Complexity system, operated by the University of Leicester IT Services, which forms part of the STFC DiRAC HPC Facility (www.dirac.ac.uk). This equipment is funded by BIS National E-Infrastructure capital grant ST/K000373/1 and STFC DiRAC Operations grant ST/K0003259/1. DiRAC is part of the National E-Infrastructure.

Research data underpinning this thesis are available at http://dx.doi.org/10.17630/a3f5b723-1c49-4464-9d4d-ffbf85bf65b5

Contents

List of Figures

List of Tables

In the darkest clouds in space, known to me and you
Hot young stars get formed and proceed to shining through.
But can the light of newborn stars
Ionise and tear up the clouds?
Yes, it can. And my computer simulations show it too.

#phdlimerick

Chapter 1

Introduction

Our galaxy, the Milky Way, is a disk-like structure containing stars and clouds of gas and dust. The clouds are mostly concentrated in the galaxy's spiral arms, where they are compressed and stretched due to collisions and tidal forces. As in all other gas-rich galaxies, many of the clouds within the Milky Way will gravitationally collapse over time, leading to the birth of new stars and the gradual depletion of gas. Astronomers have been studying this process, and while we now have a good understanding of how individual stars are born, there are still some open questions related to star formation on larger scales, where environmental factors within the star-forming clouds play an important role.

In this chapter I will overview the properties of the primary stellar birth sites, known as giant molecular clouds, and the numerical efforts of modelling star formation on that size scale. Furthermore, I will discuss the effects that the newborn stars exert on their environment through the light that they emit, and I will introduce the goal of this thesis of improving some of the numerical models of star formation by including better treatment of stellar radiation.

1.1 A recipe for a star

1.1.1 Numerical models

The study of star formation has evolved as a combination of observations, theory and numerical models. Since a typical cloud takes millions of years to collapse to the size of a star, we cannot observe the full duration of the process. Instead we can look for snapshots of star formation in our galaxy. By identifying objects, such as pre-stellar cloud cores and young stellar objects, we can observe the different stages of star formation and study their properties. To complement the observations, theory-driven numerical simulations can follow the time evolution of a star-forming cloud, taking into account the various physical processes that the cloud would be experiencing.

Both the observations and numerical models of star formation have their limitations. Young, still forming stars are obscured by thick layers of gas and dust, and cannot be observed directly. Their properties are instead inferred through measurements of the gas and dust emission. On the other hand, numerical simulations are constrained by the available computing power, both in terms of the resolution that they can achieve, and also in the accuracy with which they model physical processes. Both of these considerations have made the numerical methods particularly useful for studying single star formation, which I will overview in the following section.

The process of star formation, however, extends far beyond the individual stars and spans many size scales. Most stars are born in groups or clusters, in which the individual members share a common birth environment and gravitational potential. On even larger scales these clusters are embedded into giant molecular clouds (GMCs) which can have different structure and environmental conditions in different regions, resulting in differences in star formation rate and mass distribution of the newborn stars. Furthermore, the GMCs are located in the spiral arms of galaxies and are built and shaped by the large-scale galactic dynamics. This makes the individual act of star formation interlinked with even the most global galactic processes, and our numerical models should ideally be able to account for this link.

In reality we are not able to resolve individual stars when modelling entire galaxies. Nowadays, however, we can achieve sufficient resolution to follow the individual star formation on GMC scale (Bonnell et al. (2011), Krumholz et al. (2011)). This is a crucial shift within the scientific field, as it allows us to explore how the different environments and the stars themselves affect the ongoing process of star formation. These developments call for the need to include realistic physical conditions within the GMCs, as well as the feedback effects of the existing stars on their environment, alongside the physics that has to do with individual star formation.

1.1.2 Single-star formation

The core underlying idea of the theory of star formation is that stars are born as a result of the gravitational collapse of clouds. This idea dates back to Newton in 1692 (as quoted by Larson (2003)), and is almost as old as the first formulation of the gravitational attraction itself. The formation of a star hence begins with a cloud which becomes unstable, and for which gravity can counteract the effects of all other forces that prevent the collapse.

For a cloud of given density and temperature, Jeans (1902) showed that gravitational collapse can begin only if the cloud is larger than a characteristic size scale (see also Jeans (2009)). This size scale is known as Jeans length and can be written as:

$$\lambda_J = \pi^{1/2} c (G\rho)^{-1/2}, \tag{1.1}$$

where G is the gravitational constant, ρ is the density of the cloud that has temperature T and mean particle mass m, and $c = (kT/m)^{1/2}$ is the sound speed (Larson (2003)).

Using the Jeans length, we can also arrive at the minimum cloud mass required for a gravitational collapse, which is known as the Jeans mass and is given by:

$$M_J \propto c^3 G^{-3/2} \rho^{-1/2}. \tag{1.2}$$

Jeans' analysis was performed for an infinite cloud of uniform density in which gravity was only counteracted by thermal pressure, and his work has been criticised for its mathematical inconsistencies (Binney and Tremaine (1987)). More rigorous analysis by Larson (1985), however, studied gravitational instabilities in different cloud geometries, such as infinite plane-parallel sheets and cylinders, and with different equations of state of the gas. His findings agree with the critical mass emerging from Jeans' analysis up to the numerical coefficient, thus demonstrating that the concepts of Jeans length and mass are valuable estimates for gravitational instabilities.

Once a cloud becomes unstable under gravity the gas begins to infall towards the cloud's centre. As this happens, the density and pressure in the central region increase, and an outward pressure gradient develops in the cloud, slowing down the collapse of the outer layers. As a result, the inner regions infall faster than the outer ones, which creates a centrally peaked density profile, with $\rho \propto r^{-2}$ close to the centre (Larson (1973), Tohline (1982), Foster and Chevalier (1993)). This effect leads to a very small fraction of the cloud's mass forming the initial star. The star then accumulates most of its mass later on, through subsequent accretion of the gas envelope which surrounds it (Larson (2003)).

In the early stages of the gravitational collapse the density is still low and the cloud can cool very efficiently by emitting thermal photons. The collapse is therefore approximately isothermal and the cloud temperature remains close to 10K (Low and Lynden-Bell (1976), Rees (1976)). Later on, when the central column density exceeds a critical value, the cooling becomes inefficient due to dust absorption, and the central region increases its temperature dramatically. The pre-stellar core undergoes brief hydrostatic phases before it finally begins the fusion of hydrogen and can be considered a star.

The presence of cloud rotation flattens the collapsing gas into a disk, and it is presently believed that young stars accumulate their mass via an accretion disk (Norman et al. (1980), Narita et al. (1984)). Similarly, magnetic fields also flatten the gas on a plane perpendicular to the field lines, and they can significantly slow down the overall gravitational collapse (Mouschovias (1991)).

For completeness, we should note that the above star formation theory has been developed and established for low-mass stars. High-mass stars are hotter, and hence their accretion can be disrupted by the substantial stellar feedback effects in the form of radiative heating, radiation pressure or ionisation. Some authors have found that a star can reach at most a mass of 10 $M_\odot$ when accreting from a spherically infalling envelop (Yorke and Krügel (1977), Wolfire and Cassinelli (1987)). Numerical models including cloud rotation and circumstellar disks, have produced higher stellar masses of up to about 30 $M_\odot$, however the star formation process was found to be inefficient and requiring very massive initial cloud cores (Yorke and Sonnhalter (2002)). Furthermore, high-mass stars are primarily observed in the central regions of stellar clusters and they almost always appear in binary or multiple systems (Preibisch et al. (1999), Duchêne and Kraus (2013)). This environmental bias suggests that the formation of massive stars, and especially those over 10–30 $M_\odot$, may happen through protostellar interactions and collisions (Bonnell and Bate (2002)).

1.2 Birthplaces

1.2.1 Giant molecular clouds

GMCs are large (~ 100 pc) structures, which are positioned along the spiral arms of the galaxy. They appear mostly dark with bright patches where they are illuminated by embedded stars. GMCs have complex, clumpy structure and density spanning many orders of magnitude (see Figure 1.1). Larger dense areas in these clouds are called clumps and are believed to be the birthplaces of clusters of stars. The smallest cloud structures with the most concentrated mass are referred to as cores. A core can potentially undergo a gravitational collapse and form a single star. We probe the densities of the more concentrated parts of the clouds by measuring the strength of the molecular emission lines of NH_3, N_2H^+, HCO^+, HCN etc., which serve as a cooling mechanism in these environments.

Additionally to the complex density profiles, GMCs have large, and in most places supersonic, velocities, as evidenced by the commonly observed line broadening due to the Doppler effect (Larson (2003)). These motions within the clouds typically have velocity dispersions which increase with the size of the region over which they are observed, and are attributed to turbulence (Larson (1981), Heyer and Brunt (2004)). Possible triggering mechanisms for these velocities include large-scale cloud motions due to the galactic potential and stellar feedback in the form of stellar radiation, stellar winds and supernovae.

The temperature structure of GMCs is also non-homogeneous, with the dense cloud regions being cold (< 100 K), and being surrounded by warm, low density gas (~ 8000 K). The two distinct temperature regions can co-exist in a stable pressure balance, following the two-phase interstellar medium (ISM) solution of Field et al. (1969). The cold phase (100 K) of the ISM consists of neutral molecular and atomic gas, and it contains most of the cloud mass, while the warm phase (8000 K) is a mixture of neutral and ionised gas and occupies a large volume. Later work by McKee and Ostriker (1977) has proposed that a third, hot phase of the ISM, with a temperature of $\sim 10^6$ K, co-exists with the other two, and is caused by supernova heating. Due to the focus of this thesis, I will review the ionised portion of the warm gas in more detail in the next section.

Finally, GMCs provide valuable sites for computing star formation statistics, as they are typically hosting a large number of newly formed stars. In the past few decades we have been able to identify individual young stellar objects (YSOs), thanks to the launch of the *Spitzer Space Telescope*, the *Herschel Space Observatory* and the *Hubble Space Telescope*. These large sets of YSOs have allowed us to find the star formation rate in different clouds, and the mass distribution of young stars, known as the initial mass function (IMF) (e.g. Hillenbrand (1997)).

The IMF represents the relative occurrence of different stellar masses at the time that the stars form. As a whole, low-mass stars are more numerous than high-mass stars, and this relation was first quantified by Salpeter (1955). He described the number of stars per unit mass interval as proportional to $M_*^{-2.35}$, where M_* is the stellar mass. This power law shape for the IMF was found to be constrained on both ends of the mass range. At low stellar masses of about 0.1–0.3 $\mathrm{M}_\odot$, the IMF peaks and then decreases as the mass decreases further (Da Rio et al. (2012)). On the high-mass side of the distribution, there is a cut-off at 150 $\mathrm{M}_\odot$, beyond which no stars exist (Zinnecker and Yorke (2007)). A major question within the scientific community is whether or not the shape of the IMF is universal. While some authors have found no significant observational evidence for environment-dependent IMF (Bastian et al. (2010)), others have argued that the IMF varies between different galaxies (Cappellari et al. (2012)). Typically, however, the IMF is assumed to be universal, and it is used for calculating the star formation rate in galaxies.

The star formation rate (SFR) is defined as the gas mass that gets converted into stars per unit time (typically in solar masses per year). We can obtain the SFR in two different ways — by counting stars of a given age within our galaxy and its satellites (Hillenbrand (1997)), or by assuming the IMF and relating the light emitted from further away galaxies to the number of stars (Overzier et al. (2011)). The SFR has long been found to relate to the cold gas surface density, Σ, via the power law SFR $\propto \Sigma^{1.5}$, known as the Schmidt-Kennicut relation (Schmidt (1959), Kennicutt (1998), Kennicutt and Evans (2012)). This relation was shown to hold on a broad range of scales from small star-forming regions to entire galaxies, however it exhibits signs of environment-dependent scatter (e.g. Momose et al. (2010)).

Apart from computing the SFR and the IMF, identifying individual YSOs gives us spatial information about how they are grouped, and has helped us identify stellar clusters within GMCs (Beerer et al. (2010)). There is strong observational evidence that YSOs appear in high concentrations around the areas of ionised gas which form around hot, young stars (Beerer et al. (2010), Thompson et al. (2012)). This link between ongoing star formation and ionised regions demonstrates again the need for larger scale star formation models to include ionising stellar feedback.

1.2.2 H II regions

The common name for the areas in space that consist primarily of ionised hydrogen (H^+) is H II regions. Their presence in the Milky Way was detected in the late 19th century by observing their emission lines, however it was not until the 1930s when astronomers started to realise how numerous these objects were (Struve (1937), Struve et al. (1938), Struve and Elvey (1939)). Their primary observed feature, the Hα emission line, originates from hydrogen atoms whose electrons de-excite from the $n = 3$ to the $n = 2$ energy level. This transition occurs when free electrons are recaptured by H^+ ions, and hence it is indicative of the presence of continuously recombining hydrogen.

Soon after the revelation of substantial amounts of ionised hydrogen in our galaxy, a theoretical model of the structure of H II regions was developed by Strömgren (1939). In his pioneering work, Strömgren painted a picture in which hot, high-mass stars are surrounded by spherical regions of fully ionised hydrogen, with sizes varying from a few parsecs to hundreds of parsecs. In these regions ionisation is balanced by recombination events where electrons are recaptured by the H^+ ions. These spherical regions are now referred to as Strömgren spheres, and their sizes are given by the Strömgren radius:

$$R_{St} = \left(\frac{3Qm_p^2}{4\pi\alpha_B\rho_o^2} \right)^{1/3} . \tag{1.3}$$

In the above, Q is the number of ionising photons per second emitted by the central star, m_p is the hydrogen mass, α_B is the recombination coefficient, and ρ_o is the density of hydrogen surrounding the star. I will review how this equation is derived as I introduce the topic of radiative transfer.

Strömgren also demonstrated that the transition from fully ionised to neutral hydrogen happens in a shell so thin that it could be treated as a sharp discontinuity, referred to as the ionisation front.

Building on Strömgren's static model of an H II region, Kahn (1954) studied the dynamics of an ionisation front when the central star switches on. At early times the high density of ionising photons causes the ionisation front to expand supersonically, as hydrogen atoms further and further from the star become ionised. This stage is called R-type expansion and continues until photoionisation balances recombination in the cloud. Thereafter the expansion can only continue through pressure gradients from the hot, ionised gas. The latter expansion mechanism is referred to as D-type expansion, and the transition between the two physical regimes happens at the Strömgren radius. During the D-type expansion the ionisation front moves subsonically for the ionised hydrogen, but supersonically for the neutral hydrogen and is preceded by a shock.

These simple models do not take into account 3D effects or the presence of clumps and other structures in the neutral gas that we see in observations, however they help us build a physical intuition for the processes occurring in an H II region. They also serve as valuable benchmarks for more complex numerical models, and hence I will return to the concepts of R-type and D-type expansion later on in Chapter 5.

Nowadays numerous H II regions have been identified in the Milky Way and the surrounding spiral and irregular galaxies. Since H II regions are associated with young, hot stars (of O and B types), which have lifetimes of the order of millions of years (as opposed to billions of years for cooler stars), H II regions are viewed as signatures of recent and ongoing star formation.

1.3 Ionising stellar feedback in numerical models

1.3.1 Ionising feedback and star formation

So far we have seen that the ionising radiation of massive stars, which creates H II regions, is commonly present in star-forming clouds and it affects the gas dynamics. When it comes to affecting the process of star formation, this type of stellar feedback has been argued to both induce it (Elmegreen and Lada (1977)) and inhibit it under different circumstances (Whitworth (1979), McKee et al. (1984), Matzner (2002)).

One way in which star formation can be triggered is via the collect-and-collapse mechanism, first theoretically formulated by Elmegreen and Lada (1977). The mechanism affects the material which is swept up in a shock by the expansion of an H II region. The shell of gas within the shock can reach sufficiently high densities in order to fragment and collapse gravitationally. This type of star formation has been extensively studied with numerical models, which show that the masses of the newly produced stars depend on the thickness of the shocked shell (Wünsch et al. (2010), Dale et al. (2011)). Whitworth et al. (1994a) and Whitworth et al. (1994b) found that the stars formed in this way would be preferentially more massive, which in turn could create another H II region triggering star formation, and result in a self-propagating high-mass star formation.

Another way in which star formation can be triggered is by making pre-existing cloud cores unstable under gravity as they get compressed by an expanding H II region. This mechanism is known as radiation-driven implosion, and has been shown to occur in numerical models (Kessel-Deynet and Burkert (2003), Mellema et al. (2006), Bisbas et al. (2011)). Walch et al. (2013) have studied the effect of the morphology of the swept-up cold gas on the resulting star formation, and found that gas with low fractal dimension results in the formation of clusters, whereas high fractal dimension gas produces individual stars or multiple systems.

Observationally, there are some signs of triggered star formation around H II regions. YSOs have been found in pillars carved out by ionising radiation in regions of the Eagle Nebula (Sugitani et al. (2002)), Cygnus X (Beerer et al. (2010)), and others. Apart from physical proximity to the H II regions, newly formed YSOs need to be observed to be younger than the ionising sources by an appropriate amount in order to claim that it was a triggered event. Such age gradients have been observed in some star-forming regions, making a strong claim of feedback-triggered star formation (Choudhury et al. (2010)). Additionally, a concentration of YSOs has been reported around *Spitzer* mid-infrared bubbles, indicating triggered star formation in expanding shells (Thompson et al. (2012)).

On the other hand, H II regions can have a negative effect on star formation by dispersing the surrounding gas. Within simulations, this has been observed both on small scales, with ionisation destroying the circumstellar disks around massive stars (Hollenbach et al. (2000)), and on larger scales, with H II regions blowing apart the molecular clouds they are embedded in (Walch et al. (2012), Dale et al. (2012)). When it comes to observations, we see many clouds with visibly disrupted morphologies within the Milky

Way. Furthermore, velocity measurements suggest that many of them undergo expansion, consistent with cloud dispersal due to stellar feedback (e.g. O'dell et al. (1993), Walborn et al. (2007)).

1.3.2 Dynamic models of ionising feedback

We have seen that the ionising stellar feedback appears commonly in star-forming regions, and it can have profound impact on their dynamics and the process of star formation. I will now focus on how we incorporate ionising feedback into star formation models.

The process of star formation is modelled via hydrodynamics codes, which simulate the gas dynamics and the gravitational collapse. Depending on the numerical method which is used, hydrodynamics codes can be grid-based (e.g. Teyssier (2002)), particle-based (Lucy (1977) and Gingold and Monaghan (1977)), or moving mesh (e.g. Springel (2010)). On the other hand, the interactions of ionising radiation with matter are modelled with radiative transfer methods. Both of these numerical approaches are reviewed in detail in Chapter 2. In order to incorporate the stellar radiative feedback into star formation models, astronomers have combined various radiative transfer techniques with hydrodynamics, creating radiation hydrodynamics (RHD) schemes.

Since the hydrodynamical models of star formation are complex and computationally costly on their own, computing power has been the main limiting factor of using RHD. Therefore the earlier efforts of achieving RHD have been using radiative transfer methods with approximate accuracy and lower cost. Krumholz et al. (2009) used flux-limited diffusion (FLD) method in a RHD scheme to study the formation of high mass stars via accretion. Later on, Kuiper et al. (2010) implemented a hybrid radiative transfer treatment combining FDL and ray tracing. The hybrid scheme improves the accuracy of the radiative transfer done only with FDL, while at the same time it decreases the computing time (Kuiper and Klessen (2013)).

More recently, RHD using Monte Carlo radiative transfer (MCRT) has been successfully achieved in combination with grid hydrodynamics (Harries (2015), Vandenbroucke and Wood (2018)) and moving mesh hydrodynamics (Vandenbroucke and Wood (2018)). This is a significant breakthrough in the field, since MCRT performs accurate radiative transfer despite its substantial computing time. This recent development has been the inspiration for my thesis, which aims to combine MCRT with a particle-based hydrodynamics model, namely the smoothed particle hydrodynamics (SPH).

Presently, MCRT has been applied to the outputs of SPH simulations, after they had already finished running (e.g. MacLachlan et al. (2015)). This post-processing approach can be of value if we want to find out which parts of a cloud would become ionised by stars, however, it does not allow for the stellar feedback to affect the dynamics of the collapsing gas. In contrast, a live RDH scheme would perform the SPH and MCRT calculations together, in an integrated manner, and it would recreate ionisation-driven gas dynamics. Achieving live RHD is the goal of this thesis, and its realisation would allow for a variety of astrophysical problems to be addressed.

1.4 Thesis outline

In this thesis I wish to address one of the less understood aspects of star formation, namely how the ionising radiative feedback of young, hot stars affects the ongoing star formation in a cloud. In order to do so I use both hydrodynamics and radiative transfer models simultaneously and present my work towards developing a novel radiation hydrodynamics scheme.

Chapter 2 introduces the topics of fluid dynamics and radiative transfer, and overviews the numerical methods that are commonly employed to model them. Two of these methods are of particular interest — Smoothed Particle Hydrodynamics (SPH), used for modelling the gravitational collapse of star-forming clouds, and Monte Carlo Radiative Transfer (MCRT), which allows us to simulate the effects of stellar light on the irradiated gas. Combined together, SPH and MCRT can form a live radiation hydrodynamics scheme, and the remaining chapters of this thesis follow the steps that I took to achieve live radiation hydrodynamics.

Chapter 3 addresses the main incompatibility between SPH and MCRT, which comes from the fact that SPH is particle-based and MCRT is grid-based. In order to bridge the gap between the two codes, I construct a Voronoi tessellation, which is a grid structure consisting of irregular polyhedra, built around a set of generating sites. This allows for the SPH density structure to be mapped onto a grid, which the MCRT code can use for its calculations. In this chapter I discuss the choice of generating sites for the Voronoi tessellation, and suggest an algorithm for improving the resolution in regions of large density gradient, where a more simplistic choice of a Voronoi tessellation proves insufficient.

In Chapter 4 I address the question of how to accurately map the SPH density onto a Voronoi grid. The solution that I present uses the integral of an SPH kernel function inside the volume of a grid cell and links it to the average cell density through the SPH formalism (Petkova et al. (2018)). This approach is first analytically derived, and then numerically implemented and tested. Furthermore, I improve its computational performance by pre-computing and tabulating its functional form.

Finally, in Chapter 5 I describe how I used my work from Chapters 3 and 4 in order to assemble a live radiation hydrodynamics scheme using the SPH code Phantom (Price et al. (2017)) and the MCRT code CMacIonize (Vandenbroucke and Wood (2018)). I demonstrate the validity of the scheme by applying it to the benchmark test of D-type expansion of an H II region (Bisbas et al. (2015)). I also perform a proof-of-concept live radiation hydrodynamics simulation of a star-forming cloud, analogous to the work of Dale et al. (2012), in which naturally created high-mass stars ionise their surroundings and affect the ongoing star formation.

Figure 1.1: Hα image of the star forming region Cygnus X, and the H II region containing the North America and the Pelican Nebulae (left). *Courtesy MDW Hydrogen-Alpha Sky Survey (mdwskysurvey.org).*

Chapter 2

Numerical Modelling of Matter and Light

In this chapter I will introduce the topics of fluid dynamics and radiative transfer, and describe how they are numerically modelled. I will specifically focus on the methods of Smoothed Particle Hydrodynamics (SPH) and Monte Carlo Radiative Transfer (MCRT), as they have been the focus of my work. Finally, I will discuss how fluid dynamics and radiative transfer can be combined to create more comprehensive star formation models, and I will formulate the steps necessary to achieve the overarching goal of my thesis of combining SPH and MCRT and running them simultaneously.

2.1 Modelling star formation: Hydrodynamics and SPH

2.1.1 Fluid dynamics

In order to model the gravitational collapse of molecular clouds into stars, we need to have a formulation of the physics that governs this process. Since clouds behave like fluids, we can use the fluid equations in order to describe their motions under a set of forces.

Let us consider our (cloud) fluid which occupies a certain volume and is characterised by a set of parameters (such as density, pressure, temperature etc.) which may differ from one point in space to another. We can break down the volume into fluid elements which are small enough that the physical parameters can be considered constant within each fluid element. There are two ways in which we can do this breakdown, depending on the reference frame that we choose to adopt. One way is to define each fluid element as an entity which is fixed in space. The fluid can then flow in and out of this fluid element, carrying with itself momentum and energy. This is known as the Eulerian description of a fluid. The alternative approach would be to consider fluid elements which are carried along the flow of the fluid, and which are constantly rearranging and reorganising themselves. This description is know as the Lagrangian one.

The Lagrangian and the Eulerian formulations of a fluid are equivalent, and any equation written by using one of the descriptions can be transformed into the other one. To demonstrate how this is done, let us consider some quantity of the fluid, A. If we want to know how A changes with time at a fixed position (i.e. under the Eulerian description), we need to find the partial time derivative $\frac{\partial A}{\partial t}$. If, however, we want to know how A changes in time in a Lagrangian fluid element, we can write the Lagrangian derivative as:

$$\frac{\mathrm{D}A}{\mathrm{D}t} = \frac{\partial A}{\partial t} + \mathbf{v} \cdot \nabla A, \tag{2.1}$$

where $\mathbf{v}$ is the fluid velocity (Clarke and Carswell (2007)).

Using the above formulation, we can now introduce the three fluid equations which ensure that mass, momentum and energy are conserved. Under the Eulerian description we can write them respectively as (adapted from Choudhuri (1998), Clarke and Carswell (2007) and Price (2012)):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0; \tag{2.2}$$

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = -\frac{1}{\rho}\nabla P + \mathbf{a}_f; \tag{2.3}$$

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u + \frac{P}{\rho}\nabla \cdot \mathbf{v} = \dot{Q}_h. \tag{2.4}$$

In the above, ρ is the fluid mass density, P is the pressure and u is the specific internal energy.

By using equation 2.1 we can adapt the above to the Lagrangian formulation and get:

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} + \rho\nabla\cdot\mathbf{v} = 0; \tag{2.5}$$

$$\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = -\frac{1}{\rho}\nabla P + \mathbf{a}_f; \tag{2.6}$$

$$\frac{\mathrm{D}u}{\mathrm{D}t} - \frac{P}{\rho^2}\frac{\mathrm{D}\rho}{\mathrm{D}t} = \dot{Q}_h. \tag{2.7}$$

The vector $\mathbf{a}_f$ stands for the net acceleration induced by forces other than pressure gradients, and it depends on the physical processes that are significant for the problem of interest. In the case of modelling star formation we will most commonly use $\mathbf{a}_f = \mathbf{g}$, where $\mathbf{g}$ is the gravitational acceleration. In fluids where viscosity has an important effect, $\mathbf{a}_f$ can also contain a viscous term, given by $\frac{\mu}{\rho}\nabla^2\mathbf{v}$, where μ is the viscosity coefficient. Similarly, magnetic field effects or radiation pressure can also contribute towards $\mathbf{a}_f$.

The term $\dot{Q}_h$ in the energy equations stands for net heating and cooling , which can include local change of heat due to thermal conduction, radiation, viscous heating etc. If the temperature in our fluid is constant everywhere and unchanging, then there will be no heat flow between different points in space, and $\dot{Q}_h = 0$. Such fluid is referred to as being isothermal. If, however, temperature gradients are present within the fluid, we can have, for example, $\dot{Q}_h = \frac{1}{\rho}\nabla\cdot(K\nabla T)$ due to thermal conductivity, where T is the fluid temperature and K is the conductivity coefficient.

Equation of state

If both $\mathbf{a}_f$ and $\dot{Q}_h$ are zero, the equations of fluid dynamics contain six quantities (if we count the three components of $\mathbf{v}$), which have their time evolution described by five equations (if we break down the momentum equation into its three components). This means that we need to have at least one more equation in order to describe the time evolution of a fluid. This missing link is the equation of state and it provides a relationship between P, ρ, and the fluid temperature, T.

There are many different equations of state, depending on the behaviour of the fluid of interest. If the fluid can be approximated to an ideal gas (which is the case for most astrophysical systems), we can write its equation of state as:

$$P = \frac{k_B}{\bar{m}}\rho T, \tag{2.8}$$

where $k_B = 1.38 \times 10^{-23}$ J/K is the Boltzmann constant, and $\bar{m}$ is the mean particle mass within the gas (Clarke and Carswell (2007)). The specific internal energy, u, is then proportional to T, and can be written as $u = \frac{3}{2}\frac{k_B}{\bar{m}}T$ for atomic gas.

There are some special cases of fluids for which the equation of state is barotropic, i.e. P depends only on ρ (Clarke and Carswell (2007)). One such case is when the fluid is isothermal and then $P \propto \rho$. Another special case is when the fluid is adiabatic and $P \propto \rho^\gamma$, where γ is the adiabatic index. In both of these cases we do not need the energy equation in order to evolve the fluid in time.

Gravity

As previously stated, when modelling star formation we will commonly assume that $\mathbf{a}_f = \mathbf{g}$. In order to include gravity in the treatment of the fluid we need another equation that can tie it to the rest of the physical quantities.

Since gravity is a conservative force, i.e. one for which the work done in a closed loop is zero ($\oint \mathbf{F} \cdot \mathrm{d}\mathbf{l} = 0$, see Clarke and Carswell (2007)), we can express it in terms of a scalar potential. This means that we can define a gravitational potential, Ψ, such that:

$$\mathbf{g} = -\nabla\Psi. \tag{2.9}$$

Poisson's equation then allows us to link the local gravitational potential and the local density by:

$$\nabla^2\Psi = 4\pi G\rho, \tag{2.10}$$

where $G = 6.67 \times 10^{-11}$ m^3 kg^{-1} s^{-2} is the gravitational constant.

With this extra equation in mind, we are now in a position to model the hydrodynamics of star formation.

2.1.2 Numerical hydrodynamics

For complex fluid flows with non-symmetric geometries, such as in the case of star forming clouds, the equations presented in Section 2.1.1 are typically solved numerically, and the time evolution of the fluid properties is tracked via hydrodynamics codes. These codes use a discretisation of the simulated volume following either Lagrangian or Eulerian fluid elements. The different types of numerical schemes have certain advantages and disadvantages, which I will now present and discuss.

Grid codes

Grid-based codes, also known as finite volume schemes, model the fluid dynamics by following the Eulerian description. They divide the simulation volume into cells, which have fixed positions in space. The cells can be evenly spaced, or constructed via an adaptive mesh refinement scheme, which increases the resolution in certain areas by breaking down some of the existing cells into smaller subcells (Fryxell et al. (2000), Teyssier (2002)). The fluid dynamics is modelled by calculating fluxes of physical parameters between neighbouring cells and numerically solving the differential equations governing the parameters' time evolution. Various approaches have been adopted for solving these equations, and the development of second order schemes has allowed for the accurate treatment of shocks, which are important for star formation and stellar feedback problems (van Leer (1979)).

The downside of the grid codes is that they are fixed in geometry and in size of the simulation volume, which requires us to have prior knowledge of the system that is modelled. Nevertheless, due to their accuracy, grid codes are very widely used for studying astronomical systems.

Particle codes

Particle codes follow the Lagrangian formalism by breaking down the fluid into a finite set of particles. The particles are then moved incrementally with the local fluid velocity, and can form complex geometries, naturally clustering and increasing the resolution at areas of high density.

The method of Smoothed Particle Hydrodynamics (SPH) was first put forward by Lucy (1977) and Gingold and Monaghan (1977) as an attempt to tackle non-spherically symmetric problems in astronomy. Such scenarios required 3D simulations, which were computationally unachievable at the time with the finite volume (i.e. grid-based) schemes that were being used. SPH transforms the fluid equations into summations over particle parameters, which are quick and easy to compute. These summations are approximations and have errors associated with them, however the SPH formalism conserves the total mass, momentum and angular momentum to machine accuracy.

Agertz et al. (2007) found that the Smoothed Particle Hydrodynamics method was unable to correctly model scenarios where large density, temperature and pressure gradients were present. Further improvements of the method, however, have addressed these issues and have demonstrated that SPH can be used for the modelling of shocks and instabilities (see Price (2008) and Hopkins (2013)).

Moving mesh codes

In recent years, moving mesh codes were developed as a hybrid of the Lagrangian and Eulerian fluid formalism (Springel (2010), Duffell and MacFadyen (2011), Vandenbroucke and De Rijcke (2016)). These codes use particles, which move with the fluid, but also an unstructured grid is constructed for each particle configuration in order to compute fluxes between neighbouring particles. The unstructured grid is called a Voronoi tessellation and it will be introduced in more detail in Chapter 3. By adopting this approach, the moving mesh codes aim to combine the best parts of both the grid and particle-based methods. The main down side is the algorithmic complexity and high computing cost.

2.1.3 Smoothed Particle Hydrodynamics

SPH is the hydrodynamics method which has been used for the work presented in this thesis. As previously mentioned, it solves the Lagrangian fluid equations (2.5, 2.6, 2.7), by expressing them in terms of summations. Therefore, to understand the approach in more detail I will first introduce the SPH summations, and I will then continue with overviewing some contemporary code features.

SPH formalism

In order to represent a continuous fluid with a set of N discrete particles, we need a way of interpolating the physical quantities describing the fluid at each point in space, or otherwise we will end up with unphysical gaps in the fluid. This interpolation is a key feature of SPH, and I will demonstrate how it is constructed (see Monaghan (1992)).

Let us start with some quantity A, which is defined for each point, $\mathbf{r}$, of the simulation volume. We can write $A(\mathbf{r})$ as the following integral over the whole simulation volume V:

$$A(\mathbf{r}) = \int_V A(\mathbf{r}')\delta(\mathbf{r} - \mathbf{r}')\mathrm{d}\mathbf{r}'. \tag{2.11}$$

In the above, $\delta(\mathbf{r} - \mathbf{r}')$ is the Dirac delta function, which equals zero for $\mathbf{r} \neq \mathbf{r}'$ and infinity otherwise. The delta function also has the property

$$\int_V \delta(\mathbf{r} - \mathbf{r}')\mathrm{d}\mathbf{r}' = 1. \tag{2.12}$$

Instead of using the Dirac delta in the previous expression we can approximate it by a kernel function, $W(\mathbf{r} - \mathbf{r}', h)$, which is also peaked at $\mathbf{r} = \mathbf{r}'$, however it has a finite width of the peak, h. The kernel function is also normalised, so that its integral over all space equals one. Then we can rewrite the expression for $A(\mathbf{r})$ as

$$\begin{aligned} A(\mathbf{r}) &\approx \int_V A(\mathbf{r}')W(\mathbf{r} - \mathbf{r}', h)\mathrm{d}\mathbf{r}' && (2.13) \\ &= \int_V A(\mathbf{r}')\frac{W(\mathbf{r} - \mathbf{r}', h)}{\rho(\mathbf{r}')}\rho(\mathbf{r}')\mathrm{d}\mathbf{r}', && (2.14) \end{aligned}$$

where $\rho(\mathbf{r}')$ is the density at $\mathbf{r}'$.

If we now assume that we have a set of N particles, and we have a list of properties assigned to each particle, then we can interpolate A by

$$A(\mathbf{r}) \approx \sum_{b=1}^{N} \frac{A_b m_b}{\rho_b} W(\mathbf{r} - \mathbf{r}_b, h). \tag{2.15}$$

In the above m_b and ρ_b are the mass and the density of particle b, and A_b is the value of property A for the same particle.

We can now replace A with any property of interest and evaluate it. Note that from now on for simplicity we will use an equal sign when writing equations containing SPH sums, however these expressions are always approximations. In the most trivial case we can write the density $\rho(\mathbf{r})$ as

$$\rho(\mathbf{r}) = \sum_{b=1}^{N} m_b W(\mathbf{r} - \mathbf{r}_b, h). \tag{2.16}$$

The fluid equations, however, do not only contain physical parameters, but also some gradients of them, e.g. ∇P, and we need to establish how to handle gradients under the SPH formalism.

Starting from the SPH form of $P(\mathbf{r})$:

$$P(\mathbf{r}) = \sum_{b=1}^{N} \frac{P_b m_b}{\rho_b} W(\mathbf{r} - \mathbf{r}_b, h), \tag{2.17}$$

the naive form of the gradient can be expressed as:

$$\nabla P(\mathbf{r}) = \sum_{b=1}^{N} \frac{P_b m_b}{\rho_b} \nabla W(\mathbf{r} - \mathbf{r}_b, h). \tag{2.18}$$

In particular, we need the gradient of P at the location of a particle a:

$$(\nabla P)_a = \sum_{b=1}^{N} \frac{P_b m_b}{\rho_b} \nabla_a W_{ab}, \tag{2.19}$$

where ∇_a is the derivative with respect to $\mathbf{r}_a$, and $W_{ab} = W(\mathbf{r}_a - \mathbf{r}_b, h)$.

Then, by using equation 2.6 and disregarding $\mathbf{a}_f$, we have:

$$\frac{\mathrm{D}\mathbf{v}_a}{\mathrm{D}t} = -\frac{1}{\rho_a} \sum_{b=1}^{N} \frac{P_b m_b}{\rho_b} \nabla_a W_{ab}, \tag{2.20}$$

The above expression, however, does not conserve the total linear or angular momentum of the system, which is a problem in a hydrodynamical numerical simulation (see Monaghan (1992)).

Instead we can use that

$$\frac{1}{\rho}\nabla P = \nabla\left(\frac{P}{\rho}\right) + \frac{P}{\rho^2}\nabla\rho, \tag{2.21}$$

and get that:

$$\begin{aligned}\frac{\mathrm{D}\mathbf{v}_a}{\mathrm{D}t} &= -\sum_{b=1}^{N}\frac{P_b m_b}{\rho_b^2}\nabla_a W_{ab} - \frac{P_a}{\rho_a^2}\sum_{b=1}^{N} m_b \nabla_a W_{ab} &(2.22)\\ &= -\sum_{b=1}^{N} m_b\left(\frac{P_b}{\rho_b^2} + \frac{P_a}{\rho_a^2}\right)\nabla_a W_{ab}. &(2.23)\end{aligned}$$

Then the time derivative of the total momentum is:

$$\frac{\mathrm{D}}{\mathrm{D}t}\sum_{a=1}^{N} m_a \mathbf{v}_a = -\sum_{a=1}^{N}\sum_{b=1}^{N} m_a m_b\left(\frac{P_b}{\rho_b^2} + \frac{P_a}{\rho_a^2}\right)\nabla_a W_{ab} = 0. \tag{2.24}$$

The above expression is zero due to the overall symmetry of its components, together with the fact that $\nabla_a W_{ab} = -\nabla_b W_{ab}$, and it demonstrates that the total momentum is conserved.

In a similar fashion, other derivatives are carefully constructed, so that they obey conservation laws. For a more detailed discussion see Monaghan (1992) and Price (2012).

Contemporary code features

There have been many improvements in the area of SPH numerical modelling since the 1970s, and here I will mention only the ones which are most relevant to the work described in this thesis.

First I will introduce the concepts of particle neighbours and variable smoothing length. I have previously mentioned that the kernel function, $W(\mathbf{r}-\mathbf{r}', h)$, is used to approximate the Dirac delta function. In doing so, W is chosen, such that it has compact support, i.e. its value becomes zero when $|\mathbf{r}-\mathbf{r}'|$ is greater than a certain distance, which is expressed in terms of the smoothing length, h. This means that we can think of all other particles which are within a radius of this critical distance from a certain particle, as the particle's neighbours. While in the early days of SPH all particles had the same smoothing length at a given time, it was later decided that h should be spatially varied in order to reflect the local particle number density. Therefore, nowadays SPH particles have their individual smoothing lengths, so that each particle has roughly the same number of neighbours. Mathematically, h can be found by iterating the equations

$$\rho(\mathbf{r}_a) = \sum_{b=1}^{N} m_b W(\mathbf{r}_a - \mathbf{r}_b, h_a); \qquad h(\mathbf{r}_a) = \eta \left(\frac{\rho_a}{m_a} \right)^{1/3}, \tag{2.25}$$

where η is a unitless parameter, typically assumed to be $\eta = 1.2$ (Springel and Hernquist (2002), Price and Monaghan (2007)).

Another important feature of modern SPH codes is the creation of sink particles (Bate et al. (1995)). A sink particle replaces a group of gas particles which are gravitationally bound to one another and are undergoing a collapse, such as in the case of star formation. Once a sink particle has been created, it interacts with the other particles only gravitationally, and it can accrete additional gas particles if criteria related to proximity, energy and angular momentum of the gas are met. The incorporation of sink particles has created a great computational advantage, since gas particles trapped in tight orbits around collapsed, dense regions would require very short time steps (and hence long computing times) in order to be propagated correctly.

Finally I will briefly explain how gravity is computed within SPH. Unlike most of the other forces affecting fluids, gravity has a long range of action, and therefore each particle affects each other particle gravitationally. To compute these forces directly would be a slow task, and therefore SPH codes use tree algorithms (Appel (1985), Springel et al. (2001)). A tree (connected, non-cyclic graph) is a hierarchical structure, which can organise the particles according to their spatial distribution (see Figure 2.1). It consists of a number of nodes with links between them, and in this case each node corresponds to a box of the simulation volume (Clark et al. (2012)). The tree is built from the "root" node, which contains all particles. Each (parent) node is then divided into a number of daughter nodes, containing parts of the volume of the parent node. Through this division the parent becomes linked to its daughters. This process is repeated until the "youngest" generation of nodes ("leaves") contain only single SPH particles. The construction of this tree structure allows for particles which are clustered together to be approximated to a point mass when the gravitational force is computed for a far away particle, and this speeds up the gravity calculations significantly.

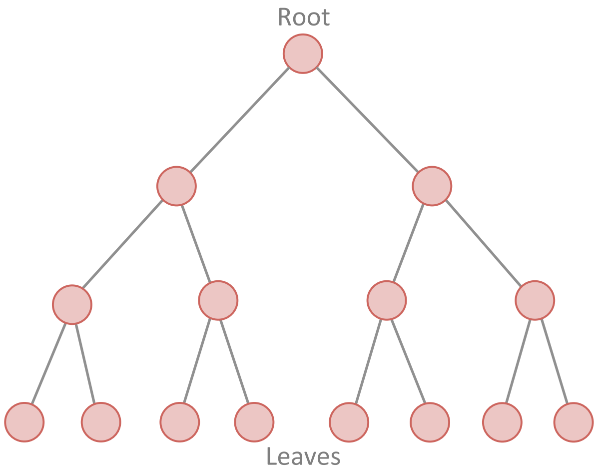

Figure 2.1: A schematic of a tree graph with the root and the leaves labelled.

With this we have reviewed how the process of star formation is modelled via the hydrodynamics of SPH, and we can now move on to discussing the modelling of stellar radiation.

2.2 Modelling radiation: Radiative transfer and MCRT

2.2.1 Radiative transfer

In order to review how radiation is modelled numerically, I will first introduce some basic concepts from the area of radiative transfer (see Rybicki and Lightman (1979) and Osterbrock and Ferland (2006) for more details).

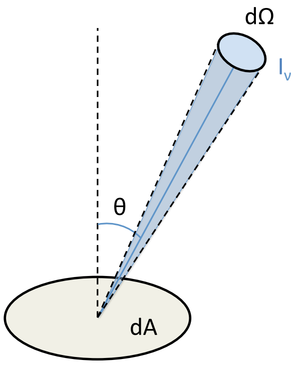

Figure 2.2: A schematic of a light ray with specific intensity I_ν and frequency ν, coming from an angle θ, spanning a solid angle $\mathrm{d}\Omega$, and hitting an area element $\mathrm{d}A$.

Let us start by considering the very common astronomical scenario of light hitting a surface element of area dA (see Figure 2.2). We can think of this surface element as part of a telescope detector, a pupil, or any arbitrary surface that we can define. If we are interested in the amount of energy that this incident light brings to the surface element, we could parameterise the problem by separating the light into different rays, each with its own specific intensity (I_ν), frequency (ν), direction (θ) and area of the sky (dΩ) that it spans. The energy contribution from each ray of light within a time period of dt can be written as:

$$\mathrm{d}E_\nu = I_\nu \cos(\theta)\mathrm{d}A\mathrm{d}t\mathrm{d}\nu\mathrm{d}\Omega. \tag{2.26}$$

We can now find the total energy by integrating the above expression. In astronomy, however, we are typically interested in other quantities, such as the mean intensity:

$$J_\nu = \frac{1}{4\pi}\int I_\nu \mathrm{d}\Omega, \tag{2.27}$$

or the flux:

$$F_\nu = \int I_\nu \cos(\theta)\mathrm{d}\Omega. \tag{2.28}$$

Here it is important to mention that the quantity I_ν is independent from the distance to the source, and is constant along a ray in the case when there is no extra absorption or emission of light happening between the source and the chosen surface element. The inverse square law that we expect for the flux ($F \propto \frac{1}{r^2}$; r – distance to source) therefore occurs from the fact that the specific intensity of a source is integrated over the angular size of the source, which in its turn depends on the inverse distance squared.

Some additional quantities that we can also derive using I_ν are the energy density and the radiation pressure, given respectively by:

$$u_\nu = \frac{1}{c}\int I_\nu \mathrm{d}\Omega = \frac{4\pi}{c}J_\nu, \tag{2.29}$$

and

$$p_\nu = \frac{1}{c}\int I_\nu \cos^2(\theta)\mathrm{d}\Omega. \tag{2.30}$$

From the above equations it is apparent that in order to model how energy is transported via radiation, all we need to know is the value of the specific intensity everywhere. This is a trivial task when our sources (i.e. stars) exist in vacuum, however I_ν is harder to find in the more realistic case where the sources are surrounded by material which can absorb and emit photons.

We can write the increment of energy added to a ray within a volume element dV as:

$$\mathrm{d}E_\nu = j_\nu \mathrm{d}V\mathrm{d}t\mathrm{d}\nu\mathrm{d}\Omega. \tag{2.31}$$

In the above j_ν is the emission coefficient, which can contain contributions from thermal and non-thermal emission, as well as light scattered into the ray of interest. We can then write the change in the specific intensity at a point s along the ray as:

$$\mathrm{d}I_\nu(s) = j_\nu \mathrm{d}s. \tag{2.32}$$

Similarly, energy can be lost from the ray due to scattering and absorption, and the change in the specific intensity is given in terms of the extinction coefficient (a_ν) by:

$$\mathrm{d}I_\nu(s) = -a_\nu I_\nu \mathrm{d}s. \tag{2.33}$$

From equations 2.32 and 2.33 we arrive at the equation of radiative transfer (ERT):

$$\frac{\mathrm{d}I_\nu}{\mathrm{d}s} = j_\nu - a_\nu I_\nu. \tag{2.34}$$

Note that in the above equation there is no time dependence and this is because we have implicitly assumed that the radiation field around us is at a steady state. This is a reasonable assumption if we are modelling a system in which the variation of the optical properties of the medium change on a time scale that is much longer than the light propagation time. In this thesis I will focus on time-independent problems only. Being able to write the specific form of the ERT for a given problem of interest and then solve it, is the essence of performing radiative transfer.

2.2.2 Ionisation

In this section we will consider the way ionising radiation is expressed within the area of radiative transfer. Let us assume that we have hot (high mass) stars surrounded by gaseous medium, consisting of pure hydrogen. What we want to find in this case is which parts of the medium would be ionised by the stars. Any such system would rapidly reach a state of ionisation balance in which the number of ionisation events will counterbalance the number of recombination events at each point in space. This idea can be expressed mathematically by (Osterbrock and Ferland (2006)):

$$n_0 \int_{\nu_0}^{\infty} \frac{4\pi J_\nu}{h\nu} \sigma_\nu \mathrm{d}\nu = n_p n_e \alpha_A(T), \tag{2.35}$$

where n_0, n_p and n_e are the local number densities of neutral hydrogen, ionised hydrogen (or protons) and electrons, respectively. Furthermore, σ_ν is the ionisation cross section of hydrogen, $\alpha_A(T)$ is the total recombination coefficient (i.e. including hydrogen atoms recapturing electrons to any of the bound energy levels), which is function of the local temperature, T, and h is the Planck constant. In the above we have only included photons of frequencies above ν_0, which corresponds to the ionisation energy of hydrogen, $h\nu_0 = 13.6eV$.

It is important to note that as a recombination event occurs, a photon is also emitted. When the recaptured electrons occupy any of the excited states of hydrogen, the photons that are emitted have frequencies $\nu < \nu_0$, so for our purposes we can disregard them. Recombination events to the ground state, however, emit ionising photons on their own. This leads us to a natural separation between the stellar ionisation field (of photons coming from the stars) and the diffuse field (of photons originating from recombination events).

In order to obtain the ionisation structure we need to formulate the ERT (see 2.34). First, we can separate the specific intensity into a stellar term ($I_{\nu s}$) and a diffuse term ($I_{\nu d}$):

$$I_\nu = I_{\nu s} + I_{\nu d}. \tag{2.36}$$

Since the stellar term consists only of ionising light that originates from the stars, the ERT for it is simply:

$$\frac{\mathrm{d}I_{\nu s}}{\mathrm{d}s} = -a_\nu I_{\nu s}, \tag{2.37}$$

where $a_\nu = n_0\sigma_\nu$. The above equation has the solution

$$I_{\nu s}(\tau_\nu) = I_{\nu s}(0)e^{-\tau_\nu}, \tag{2.38}$$

with τ_ν being the optical depth, defined as

$$\tau_\nu(s) = \int_{s_0}^{s} a_\nu \mathrm{d}s' = \int_{s_0}^{s} n_0\sigma_\nu \mathrm{d}s'. \tag{2.39}$$

The term $I_{\nu s}(0)$ is either zero if there is no star along the line of sight of the ray, or is equal to the brightness of the stellar surface.

The ERT for the diffuse term is more complex and can be written as:

$$\frac{\mathrm{d}I_{\nu d}}{\mathrm{d}s} = -n_0\sigma_\nu I_{\nu d} + j_\nu. \tag{2.40}$$

This equation has a solution of the form:

$$I_{\nu d}(\tau_\nu) = I_{\nu d}(0)e^{-\tau_\nu} + \int_0^{\tau_\nu} e^{-(\tau_\nu - \tau'_\nu)} \frac{j_\nu}{n_0\sigma_\nu}\mathrm{d}\tau'_\nu. \tag{2.41}$$

The term $\frac{j_\nu}{n_0\sigma_\nu}$ is known as the source function and we can sometimes approximate it to the Planck function, given by:

$$B_\nu(T) = \frac{2h\nu^3}{c^2}\frac{1}{\exp(h\nu/kT) - 1}, \tag{2.42}$$

where c is the speed of light. This approximation holds for physical systems in which the particles follow Maxwellian distribution before and after recombination.

In principle we could now obtain any diffuse field radiative transfer quantity provided that we knew the temperature and density at any point of the hydrogen gas surrounding the stars. However, since the local temperature depends on the local ionisation state, the temperature may need to be obtained iteratively.

In order to better describe what is meant by the local ionisation state of the medium we can express the fraction of hydrogen which is ionised at any point in space. We will denote this as the ionic fraction $f_i = \frac{n_p}{n_0+n_p}$. For a pure hydrogen medium we have that $n_e = n_p$, and then by using equation 2.35 we can solve for f_i.

Strömgren radius

If, like Strömgren, we consider a single star placed in a uniform density cloud, and we are interested in the question how far away from the star the cloud will stop being ionised, we can derive this distance (Strömgren radius) from the above formalism. First, we need to notice that j_ν is related to the number of recombinations to the ground level by :

$$\int_{\nu_0}^{\infty} \frac{4\pi j_\nu}{h\nu} \mathrm{d}\nu = n_p n_e \alpha_1(\mathrm{H}^0, T), \tag{2.43}$$

where $\alpha_1(T) < \alpha_A(T)$ is the recombination coefficient to the ground level.

Since we consider a cloud which is larger than the whole ionised region, we can assume that none of the photons are escaping, and therefore we have:

$$\int \frac{4\pi j_\nu}{h\nu} \mathrm{d}V = \int \frac{4\pi J_{\nu d}}{h\nu} n_0 \sigma_\nu \mathrm{d}V. \tag{2.44}$$

Next, we assume that this relation holds locally as well. This is called "on-the-spot" approximation and yields

$$J_{\nu d} = \frac{j_\nu}{n_0 \sigma_\nu}. \tag{2.45}$$

By noticing that $J_\nu = J_{\nu s} + J_{\nu d}$ in equation 2.35, and substituting 2.45 in 2.35, we get:

$$n_0 \int_{\nu_0}^{\infty} \frac{4\pi J_{\nu s}}{h\nu} \sigma_\nu \mathrm{d}\nu = n_p n_e \alpha_B(T), \tag{2.46}$$

with $\alpha_B(T) = \alpha_A(T) - \alpha_1(T)$.

We can express $J_{\nu s}$ from equation 2.27 as

$$J_{\nu s} = \frac{1}{4\pi} \int I_{\nu s} \mathrm{d}\Omega = \frac{R^2}{4r^2} I_{\nu s}, \tag{2.47}$$

where R is the radius of the star and r is the distance from it. The above expression assumes that $r >> R$, which is a reasonable assumption considering the sizes of H II regions.

Substituting the above together with equation 2.38 into equation 2.46, we get:

$$\frac{R^2 n_0}{r^2} \int_{\nu_0}^{\infty} \frac{\pi I_{\nu s}(0)}{h\nu} \sigma_\nu e^{-\tau_\nu} \mathrm{d}\nu = n_p n_e \alpha_B(T), \tag{2.48}$$

Finally, we will integrate both sides of 2.48 over volume, giving

$$\int_{r=0}^{\infty} 4\pi R^2 n_0 \int_{\nu=\nu_0}^{\infty} \frac{\pi I_{\nu s}(0)}{h\nu} \sigma_\nu e^{-\tau_\nu} \mathrm{d}\nu \mathrm{d}r = \int_0^{\infty} 4\pi n_p n_e \alpha_B(T) r^2 \mathrm{d}r. \tag{2.49}$$

Let us consider the left hand side of the equation. The only parameters which vary with r are n_0 and τ_ν. From earlier we can express $\frac{\mathrm{d}\tau_\nu}{\mathrm{d}r} = n_0\sigma_\nu$, which means that

$$\int_0^{\infty} n_0 \sigma_\nu e^{-\tau_\nu} \mathrm{d}r = \int_0^{\infty} e^{-\tau_\nu} \mathrm{d}\tau_\nu = 1. \tag{2.50}$$

What is left of the left hand side is now

$$4\pi R^2 \int_{\nu_0}^{\infty} \frac{\pi I_{\nu s}(0)}{h\nu} \mathrm{d}\nu = \int_{\nu_0}^{\infty} \frac{L_\nu}{h\nu} \mathrm{d}\nu = Q, \tag{2.51}$$

where L_ν is the stellar luminosity at ν, and Q is the total number of ionising photons emitted by the star per unit time.

On the right-hand-side we can notice that $n_p = n_e \approx 0$ when $r > R_{St}$ with R_{St} being the Strömgren radius. Conversely, $n_p = n_e \approx n_H$, where n_H is the total number density of hydrogen atoms and ions. Both of these observations lead to the following expression for the right-hand-side:

$$\int_0^\infty 4\pi n_p n_e \alpha_B(T) r^2 \mathrm{d}r = \frac{4}{3}\pi R_{St}^3 n_H^2 \alpha_B. \tag{2.52}$$

Combining equations 2.51 and 2.52 we can express the Strömgren radius as:

$$R_{St} = \left(\frac{3Q}{4\pi\alpha_B n_H^2}\right)^{1/3} \tag{2.53}$$

2.2.3 Modelling radiative transfer

As I stated in the previous section, we can obtain any desired quantity in radiative transfer if we know the specific intensity, I_ν. Solving for the specific intensity, however, can be non-trivial for any setup which is non-uniform or non-spherically symmetric. Furthermore, even if we could find I_ν, it is a six-dimensional quantity (it depends on three spatial coordinates, two directional coordinates, and frequency). This makes it beyond the scope of any analytic treatment for scenarios where we are modelling a complex, clumpy medium, such as a star forming cloud. For that reason radiative transfer is typically modelled numerically, using a variety of algorithms, such as diffusion, ray tracing and Monte Carlo techniques.

Similarly to the hydrodynamics models, numerical radiative transfer requires a way of describing the density structure of the gaseous medium surrounding the sources of light. Both particle and grid-based discretisations of space have been employed by some of the radiative transfer methods, as they evolved alongside the hydrodynamics.

The goals of the radiative transfer models can be divided into two categories. The first one is to obtain values for parameters, such as temperature, ionic fraction, radiation pressure etc., at each particle or cell within the simulated volume. These calculations are particularly useful when the radiative transfer method is coupled with a hydrodynamics scheme, which can incorporate the radiative effects into the gas or dust dynamics. The second category is to do with producing synthetic images, which can serve as a link between numerical models and observations of nebulae.

I will now overview some of the most common approaches used to model radiative transfer and I will discuss how they compare to one another.

Diffusion method

The diffusion method approximates the propagation of light in an absorbent medium to a diffusive process, which can be described by an equation of the type (Lucy (1977)):

$$\mathbf{F} \propto -\frac{\sigma_{SB} T^3}{\kappa \rho} \nabla T, \tag{2.54}$$

with $\mathbf{F}$ being the radiative flux, σ_{SB} being the Stephan-Boltzmann constant, κ being the Rosalind mean opacity, and T and ρ being the local temperature and density of the cloud, respectively.

The diffusion method has a low computational cost and it has been used since the early days of Smoothed Particle Hydrodynamics by Lucy (1977), with its formulation being later improved by Brookshaw (1994). In the context of SPH the radiative diffusion was added as an extra term ($\sim -\frac{1}{\rho}\nabla \cdot \mathbf{F}$) to the energy equation and was used to model optically thick (with $\tau_\nu > 1$) protostars.

In optically thin regions (with $\tau_\nu < 1$) the diffusion method breaks down, as it can result in very large propagation speeds which violate causality. As a result, various authors have adopted flux limiters to correct for this issue (e.g. Levermore and Pomraning (1981)). Even with this modification, however, the flux-limited diffusion approach does not model optically thin regions accurately. In particular, it is unable to cast sharp shadows behind thick barriers illuminated by point sources.

Ray tracing method

The ray tracing method follows the path of rays originating from the sources and passing through the surrounding cloud. In some ways this method has the strongest resemblance to the theoretical description of radiative transfer. As we move along each ray we calculate the optical depth (τ_ν) to the current location and the attenuation of the stellar flux that it creates (Pawlik and Schaye (2008)).

The ray tracing method works for both optically thin and optically thick environments, however it requires careful sampling of the directions in which rays are propagated. In simulations containing complex, clumpy clouds, particles or cells (if adaptive grid is used) are clustered together in the denser regions. With high particle or cell density in some areas of the simulation volume, there is the risk that some of the particles/cells will not be crossed by any of the rays that are considered, resulting in parts of the simulation volume being omitted. In order to address this issue, more rays need to be generated through denser environments, increasing the computational cost of the method.

Monte Carlo method

Monte Carlo radiative transfer (MCRT) imitates the process of light-matter interaction in nature by emitting photon packets from the radiation sources and propagating them through the surrounding medium. A photon packet can undergo scattering and have its direction altered, or it could get absorbed and deposit its energy at a location. An absorbed photon packet is later reemitted at a new frequency corresponding to the local temperature of the cloud.

MCRT bypasses solving the equation of radiative transfer (see 2.34), however it computes accurate values for the radiative parameters of interest, and it produces correct shadows. Additionally, by keeping track of the photon packets escaping the simulation volume, the Monte Carlo method can be employed for creating synthetic images. The downside of MCRT is the Poisson noise arising from the statistical description of the stellar radiation, which can be improved by using a large number of photon packets to increase the signal-to-noise ratio. This need for more photon packets adds an extra computational cost of the method, however due to its highly parallelizable nature MCRT is broadly used for tackling complex problems.

Hybrid methods

Various hybrid combinations of the above methods have also been adopted for specific purposes. Flux-limited diffusion has been used together with ray tracing to model ionising stellar radiation in environments which have both optically thick and optically thin regions (Kuiper et al. (2010)). The ray tracing is used around the stars, until the light enters the denser regions of the cloud and could be accurately represented by flux-limited diffusion. MCRT is also sometimes combined with flux-limited diffusion for very dense and optically thick regions (Min et al. (2009)). In environments of high density the photon packets have very short propagation distances between consecutive scattering events and can become trapped in such environments, thus greatly contributing to the computational cost of the

Monte Carlo method. Finally, MCRT is often used alongside ray tracing for imaging purposes (Robitaille (2011)). The Monte Carlo approach is typically the one used for calculating the temperature profile of the cloud and then parallel rays are traced through the cloud in the direction of an imaging plane. This hybrid technique can greatly reduce the noise which is typically present in pure MCRT synthetic images.

2.2.4 Monte Carlo radiative transfer for photoionisation

The Monte Carlo method has been used since the 1940s to model the transport of, first, neutrons and, later on, photons through a medium. For the work in this thesis I will focus in particular on the Monte Carlo radiative transfer method, as applied to ionising radiation.

Probabilities

From a mathematical perspective, the Monte Carlo method is about statistical sampling of probability distributions. There are certain properties of astrophysical sources of light that we have statistical information about. For example, stars have known spectra that tell us how many photons of a certain frequency a star would be emitting relative to photons of other frequencies. If a spectrum is normalised so that the integral over its whole range equals one, then it becomes a probability function. When choosing a large number of photon packets that our star emits by sampling the probability distribution of the stellar spectrum, we ensure that the resulting radiative transfer scheme will lead to correct results. This general approach applies to all other statistical information that we have about the system, such as the preferred direction in which sources emit (for our purposes that would be isotropic), optical depth etc. Through the Monte Carlo method we will sample all of these independent properties in order to generate photon packets which will be emitted by sources and propagated through the surrounding medium until they get absorbed or they exit the simulation volume.

In practice, the sampling of probability distributions happens by casting a random number between 0 and 1 and relating it to the probability function $P(x)$ of a property x. Let us denote the random number by ξ. Note that it is a different random number each time. Then, if x can take values in the range between a and b, then $\int_a^b P(x)\mathrm{d}x = 1$, since $P(x)$ is a probability function. This means that we can always find $X \in [a, b]$, such that

$$\int_{a}^{X} P(x)\mathrm{d}x = \xi. \tag{2.55}$$

By inverting the above equation and solving it for X, we perform random sampling of the probability distribution for x.

Relating Monte Carlo to radiative transfer parameters

As with the hydrodynamics, the Monte Carlo radiative transfer (MCRT) method requires a discretisation of the simulation volume, and this can be done both via particles and grid cells, as previously discussed. MCRT has been performed on SPH particles directly by some authors (Forgan and Rice (2010), Lomax and Whitworth (2016)), however none of these approaches have been adapted for ionising radiation so far. Therefore, from now on we will only focus on the Monte Carlo method as applied to a grid.

In order to reproduce radiative transfer quantities with MCRT, we need to establish a connection between the two frameworks. Since MCRT handles photon packets, it can store information about their frequencies, and paths through the grid. These can be related to the mean intensity, $J_{\nu,i}$, in a grid cell i, by the following equation from Vandenbroucke and Wood (2018):

$$\int_{\nu_0}^{\infty} \frac{4\pi J_{\nu,i}}{h\nu}\sigma_{\nu}\mathrm{d}\nu \approx \frac{Q}{W_t V_i}\sum_{j} w_j l_{i,j}\sigma_{\nu_j}. \tag{2.56}$$

In the above, the right hand side contains a sum over all photon packets which pass through the cell. Each photon packet has a frequency ν_j, a path length of $l_{i,j}$ as it passes through cell i, and a weight w_j, which is determined by the luminosity of its source. The weights are included in order to ensure that low luminosity sources would emit sufficient numbers of photon packets, thus reducing the numerical noise of the Monte Carlo method. W_t is the sum of all photon packet weights in the simulation, and Q is the total number of photons emitted per second. Finally, V_i is the volume of the cell. By combining equation 2.35 with equation 2.56 we can then solve for the ionic fraction in each grid cell.

Similarly, we can relate MCRT to the force per unit volume due to deposition of momentum in each grid cell by:

$$\mathbf{f}_{rad,i} \approx \frac{Qn_{0,i}h}{W_t V_i c} \sum_j w_j \nu_j \sigma_{\nu_j} l_{i,j} \hat{\mathbf{l}}_{i,j}, \tag{2.57}$$

where $n_{0,i}$ is the number density of neutral hydrogen in cell i, $\hat{\mathbf{l}}_{i,j}$ is the unit vector along the direction of the path $l_{i,j}$, and c is the speed of light. The above expression has been adapted from Harries (2015) and disregards scattering events.

There are other properties that can also be computed with an MCRT photoionisation code, however the ones mentioned above are all that are necessary in order to perform radiation hydrodynamics, as I will discuss in the next section.

2.3 Radiation hydrodynamics

We have reviewed both fluid dynamics and radiative transfer, and the ways in which they can be modelled numerically. We can now combine them into the topic of radiation hydrodynamics, which is a fundamental part of the work in this thesis.

Simply stated, radiation hydrodynamics (RHD) is the combination of a hydrodynamics model with a radiative transfer scheme. The term is used quite broadly and can stand for one of two approaches of combining the above methods. One way is by applying a radiative transfer code to a snapshot of a hydrodynamics simulation after its completion. This is commonly done for producing synthetic images, or for a computationally cheaper way of finding what effect the radiation might have on the gaseous medium of interest, and is referred to as post-processing. The other approach involves performing the radiative transfer and the hydrodynamics simultaneously, so that they can communicate their calculations to each other during runtime. This is known as live RHD and achieving it using SPH and MCRT is the goal of my work.

From a theoretical standpoint, the way to include radiative transfer into the hydrodynamics calculations is through modifying the $\mathbf{a}_f$ and $\dot{Q}_h$ terms in equations 2.6 and 2.7, based on the radiative transfer calculations. $\mathbf{a}_f$ can be related to the radiation pressure, while $\dot{Q}_h$ depends on the ionisation fraction of each particle. For the work presented in this thesis I will only use the ionisation fraction and modify $\dot{Q}_h$. From a practical perspective, there are preliminary considerations to make when combining the particle-based SPH and the grid-based MCRT. Due to the different ways in which the two numerical models represent the density structure of a diffuse medium, we need to include an intermediate step of mapping the SPH dataset onto a grid structure. For this work I have chosen the grid to be a Voronoi tessellation, which I will introduce and discuss in the next chapter. Once a grid has been constructed, another important consideration is how to map the SPH densities onto the grid cells. Finally, after the MCRT is performed on the grid and it computes ionic fractions and thermal energies, these parameters then need to be mapped back onto the particles in order to be used by SPH.

In the following chapters I will address the preliminary considerations listed above and I will demonstrate a proof of concept of the live radiation hydrodynamics scheme, which I have assembled.

Chapter 3

Voronoi Tessellation

In this chapter I will introduce the concept of a Voronoi tessellation (grid), as a geometric structure that can be used to link Smoothed Particle Hydrodynamics (SPH) and Monte Carlo Radiative Transfer (MCRT). I will review its properties and construction algorithms, and I will then move to describing how MCRT can be performed on a Voronoi tessellation.

Additionally, I will discuss the issues related to resolving SPH particles with a Voronoi grid, and will propose methods for locally addressing these issues by using a higher or a lower number of Voronoi cells per SPH particle.

3.1 Voronoi tessellation: a geometric link between SPH and MCRT

The goal of this thesis is to achieve live radiation hydrodynamics with SPH and MCRT. To run MCRT on complex non-uniform density data (such as from a star-forming cloud), we must be able to assign different properties to different parts of the medium (temperature, density, chemical abundance, ionisation fraction etc.). This is commonly achieved by discretising space, i.e. by constructing a grid structure and mapping the SPH density distribution onto the grid.

The simplest way to achieve this is by using a Cartesian grid (or evenly spaced cylindrical or spherical grid), as it is trivial to build and store. Propagating photon packets through the grid structure boils down to being able to identify in which cell the photon packet currently is, at what point it is leaving the cell and in which cell is it going next.

The main downside of this approach is that SPH particles follow the flow of a fluid and can have very uneven arrangements, so in order to achieve good resolution we will need a lot of grid cells and most of them will not contain any SPH particles.

An improvement of the Cartesian grid is the adaptive mesh refinement (AMR) scheme (Kurosawa and Hillier (2001), Steinacker et al. (2002), Harries et al. (2004)). AMR is achieved by taking a coarse Cartesian grid and recursively breaking down each cell into subcells if more resolution is needed there. A very common type of AMR is the octree in which each refined cell is divided into 8 new cells. This method can drastically improve the memory required by a well resolved Cartesian grid and is also relatively easy to implement. If the cells are stored in a tree structure, the photon packets can be propagated by performing searches in the tree. A downside of this method is the fact that it still creates plenty of empty cells (in some cases up to 60% being empty, as reported by Camps et al. (2013)).

In recent years an alternative approach using Voronoi tessellations has been proposed by some authors (Brinch and Hogerheijde (2010), Camps et al. (2013), Vandenbroucke and Wood (2018)). A Voronoi tessellation is a discretisation of space into different cell structures organised around a set of generating sites (see Figure 3.1). Each generating site is assigned the points in space that are closer to itself than to any other generating site (Voronoi (1908), Dirichlet (1850)). Typically for MCRT purposes the generating sites are taken to be the SPH particle positions, and hence there is exactly one cell per particle. This is certainly more efficient in terms of particle numbers, and is the method that has been selected for the rest of my thesis work. A downside of this method is that it is more complex to implement in terms of grid construction, and I will review the Voronoi tessellation construction algorithms in Section 3.2. Additionally, the Voronoi cells take the shape of unstructured polyhedra, so tracking the photon packets is less intuitive. I will discuss how this is achieved in Section 3.3.

Finally, for completeness I should point out that there was a recently developed MCRT algorithm that does not map the SPH data onto a grid but propagates the photon packets directly through the particles (Lomax and Whitworth (2016)). This is an intriguing new way of doing things and it is worth considering it for a potential radiation hydrodynamics approach in the future.

3.2 Constructing Delauney triangulation and Voronoi tessellation

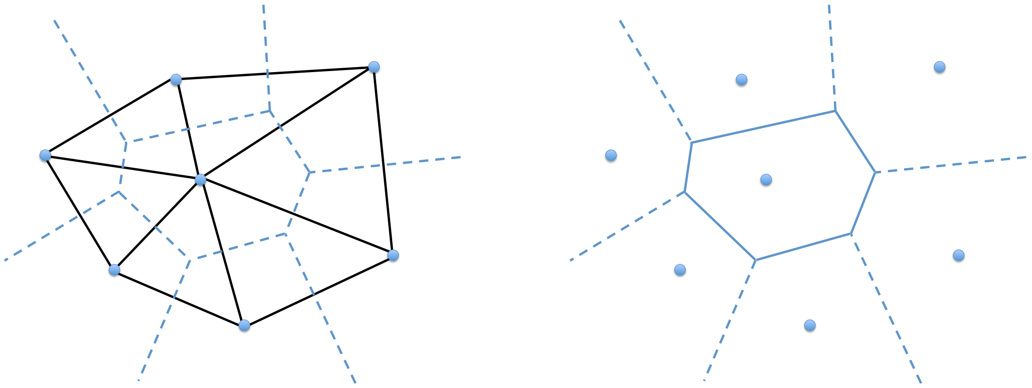

Figure 3.1: Delaunay triangulation *(left)* and Voronoi tessellation *(right)* of a set of seven generating sites in 2D.

One of the most common construction methods for a Voronoi tessellation was developed by Bowyer (1981) and Watson (1981). The Bowyer-Watson algorithm is not, in fact, focused on building the tessellation itself, but its dual graph, known as Delaunay triangulation (see Figure 3.1). We can get the Delaunay triangulation from a Voronoi tessellation when we connect with a line the generating sites sharing a Voronoi cell wall. Conversely, if we have constructed the Delaunay triangulation, we can obtain the Voronoi cell walls by bisecting each line of the triangulation. Both the Delaunay triangulation and the Voronoi tessellation for a given set of generating sites are unique, with the exception of some special degenerate cases of generating sites arrangement (e.g. if four or more generating sites are placed on a circle in 2D).

The Delaunay triangulation has the property that none of the circumcircles of its triangles, or circumspheres of its tetrahedra in 3D, contain any of the generating cites that are not part of that triangle or tetrahedron (see Figure 3.2). This property is what various algorithms use in order to construct the triangulation.

For my work I have written my own Fortran implementation of the Bowyer-Watson algorithm in 2D. This algorithm is incremental, so at each step it adds a new generating site to an already constructed and valid Delaunay triangulation (see Figure 3.3). The method starts by creating a super-triangle, which contains the positions of all of the generating sites and serves as an external boundary. A set of three points can be triangulated in only

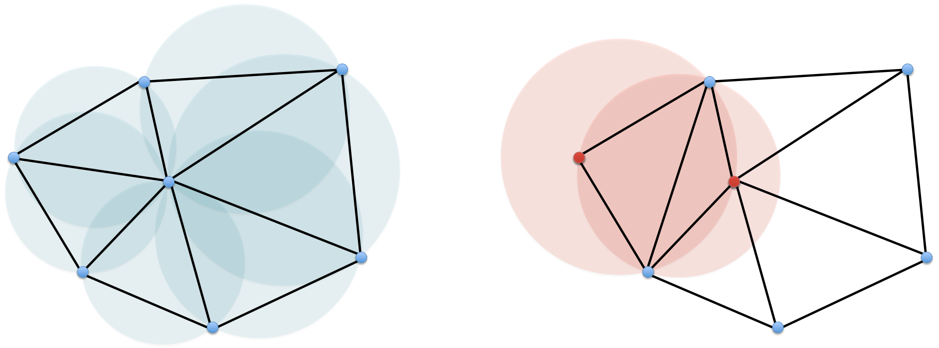

Figure 3.2: *Left:* the Delaunay triangulation for a set of seven generating sites in 2D. Note that none of the blue circumcircles contain generating sites external to their triangles. *Right:* A different triangulation of the same seven generating sites. This time two of the circumcircles contain generating sites (marked in red), which are external to their triangles.

one way, so by definition it is a valid Delaunay triangulation. The generating sites are then added one by one to the construction, following the steps:

1. Loop over all existing triangles and make a list of those whose circumcircles contain the newly added generating site.

2. Of all the sides of the triangles that are part of the above list, delete those that are repeated twice within the list.

3. Connect the generating sites of the removed triangles to the newly added generating site.

The Bowyer-Watson algorithm can also be performed in 3D, and I have adapted my own implementation for that accordingly. For further 3D applications, however, I decided against using my own code in favour of a more thoroughly tested and used, publicly available library.

An alternative Delaunay triangulation construction method has been proposed where we start from a triangulation containing all of the generating sites and we perform a series of triangle flips until we reach the Delaunay triangulation (see an example of triangle flip in Figure 3.2). This method has been shown to always be able to converge in 2D (Lawson (1972)), however in 3D only some initial triangulations can be evolved to reach the Delaunay triangulation (Joe (1991)).

Finally, we could construct the Voronoi tessellation directly without building the Delaunay triangulation first. This approach is adopted by the public C++ library VORO++

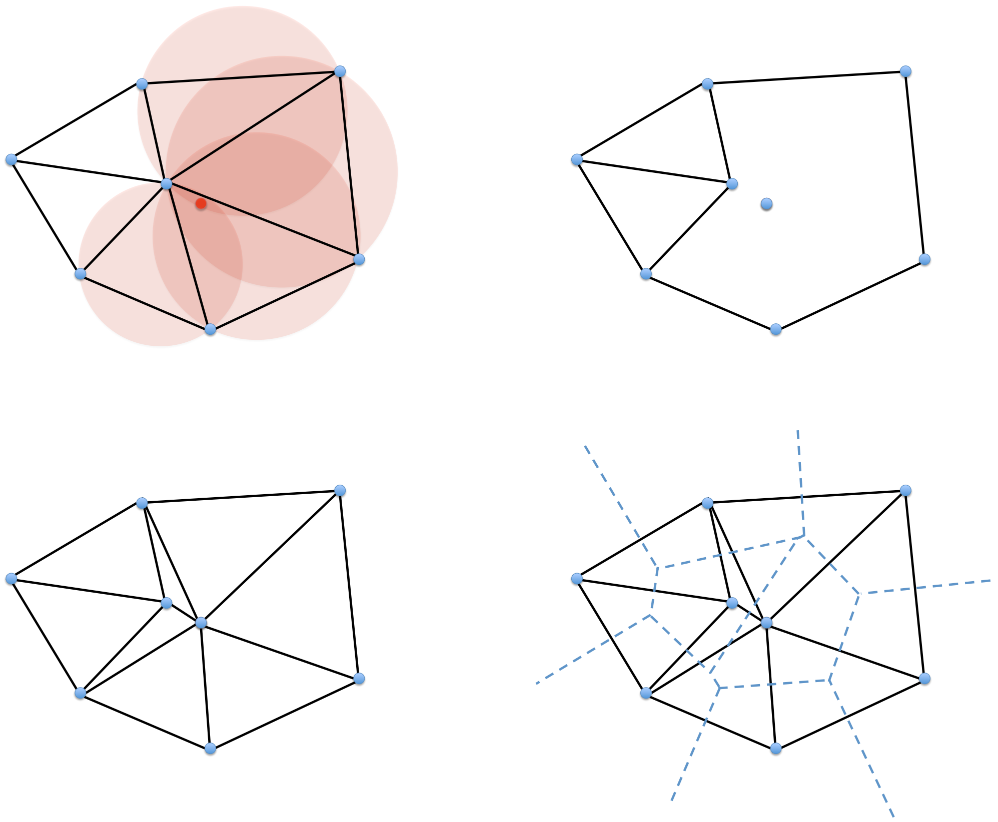

Figure 3.3: Illustration of the incremental Bowyer-Watson algorithm in 2D. *Top left:* insert a new generating site and find the circumcircles that contain it. *Top right:* delete all repeating sides of triangles whose circumcircles contain the new generating site. *Bottom left:* connect the generating sites of these triangles to the new generating site. *Bottom right:* we now have a new Delaunay triangulation with one extra generating site.

(Rycroft (2009)). In it each cell is computed individually by starting with a large box, which is then cut by the planes that bisect the distances to the nearby generating sites. The library performs the construction algorithm with efficiency and is easy to parallelise. This makes it suitable for handling large datasets, such as the outputs of SPH simulations. For these reasons, I have chosen to use it for my 3D applications.

3.3 MCRT on a Voronoi grid

One way of propagating photon packets through a Voronoi grid is by using the transport algorithm developed for SimpleX (Ritzerveld and Icke (2006)) and adapted in LIME (Brinch and Hogerheijde (2010)). In their method photon packets are only allowed to move along the lines of the Delaunay triangulation for the particular Voronoi grid. This makes the code faster because the paths are fixed and pre-computed, however they need to correct for path length approximations and the photon packet movements are restrictive.

The approach adopted for the rest of this thesis is the one presented in Camps et al. (2013), which allows the photon packets to propagate in all directions, similarly to the Cartesian grid propagation. I will now review how this is achieved.

Finding the cell in which the photon packet is conained

Finding the cell in which a photon packet is contained is equivalent to finding the nearest generating site to the photon packet's location and can be performed in many ways. This step is only needed when introducing a new photon packet to the simulation, and as such there is less need for it to be highly computationally optimised. Therefore, we could compute the distance from the photon packet to all of the generating sites and find the minimum.

Camps et al. (2013) adopt a method of dividing the Voronoi cells between a set of cuboidal boxes. For a given photon packet location we can trivially identify its box and compute the distances to all generating sites in the box, taking the minimum. This is, in fact, a very simple version of constructing a tree structure consisting of three levels. The root level is the entire simulation volume, which splits into the individual boxes (branches), and each box splits into its Voronoi cells (leaves).

This method can be further optimised by using a tree structure with larger number of levels (such as in Vandenbroucke and Wood (2018)). This optimisation may ultimately become significant if very large datasets are considered.

Propagating the photon packet through a cell

Let us consider a photon packet located at position $\mathbf{x_0}$ in cell i, which is moving in direction $\mathbf{n}$. The straight line path of the photon packet can be written as

$$\mathbf{L}(s) = \mathbf{x_0} + s\mathbf{n}, \tag{3.1}$$

where s is the distance travelled along the path.

The most crucial step in MCRT is finding the value of s that corresponds to a randomly sampled optical depth τ. Since all physical properties inside of a grid cell are assumed to be constant, we can make the following guess for s:

$$s = \frac{\tau}{a_i}, \tag{3.2}$$

where a_i is the extinction coefficient of cell i.

If our guess for s results in the point $\mathbf{L}(s)$ being in cell i, then we have successfully found the distance travelled that covers an optical depth of τ. If, however, the distance s brings the photon packet across a cell wall, and into a new cell (say, j), we only transport the photon packet to the cell wall. This will bring the photon packet a distance s_{ij} along its line path, which will correspond to an optical depth of $\tau_i = a_i s_{ij}$. Once the photon packet leaves cell i we set $\mathbf{x_0} = \mathbf{L}(s_{ij})$ and repeat the process with a new guess $s = (\tau - \tau_i)/a_j$. This process is repeated until the photon packet exits the simulation volume, or the optical depth of τ is reached.

In order to execute the above photon transport algorithm, we need to have a quick and easy way of recognising that we have crossed a cell wall. Let us assume that the positions of the generating sites of cells i and j are $\mathbf{p}_i$ and $\mathbf{p}_j$ respectively. Then the vector from $\mathbf{p}_i$ to $\mathbf{p}_j$ can be written as:

$$\mathbf{l}_{ij} = \mathbf{p}_j - \mathbf{p}_i. \tag{3.3}$$

Since a Voronoi cell wall bisects the distance between two neighbouring generating sites, the point

$$\mathbf{p}_{ij} = \frac{\mathbf{p}_i + \mathbf{p}_j}{2} \tag{3.4}$$

is located on the cell wall plane. This allows us to write the equation of the cell wall plane as:

$$\mathbf{l}_{ij} \cdot (\mathbf{x} - \mathbf{p}_{ij}) = 0, \tag{3.5}$$

where $\mathbf{x}$ is any point in the plane.

If we then set $\mathbf{x} = \mathbf{L}(s_{ij}) = \mathbf{x_0} + s_{ij}\mathbf{n}$, we can rearrange and find the following form for s_{ij}:

$$s_{ij} = \frac{\mathbf{l}_{ij} \cdot (\mathbf{p}_{ij} - \mathbf{x}_0)}{\mathbf{l}_{ij} \cdot \mathbf{n}}. \tag{3.6}$$

If s_{ij} is positive, then the photon packet is able to cross the plane of the cell wall while moving in direction $\mathbf{n}$. In order to determine which of the cell walls is crossed first, and hence which cell the photon packet enters next, it is sufficient to calculate the s_{ij} values for all of the cell walls of i and select the smallest positive distance.

The above algorithm is simple to implement and fast to execute. Additionally, it does not require any extra information to be stored about the Voronoi cells beyond the positions of the generating sites and a neighbours list for each cell, which makes it very suitable for performing MCRT.

3.4 Choosing the generating sites for a Voronoi grid

3.4.1 Resolution issues

The simplest way of choosing the set of generating sites for the construction of a Voronoi grid is by using the SPH particle positions. This automatically ensures that we will get more grid cells in the areas that contain more particles and hence we will obtain good overall resolution. In some cases, however, when large density gradients are present in the SPH dataset, this approach can be less desirable due to density artifacts.

In SPH, a region of high density consists of more particles, which have short smoothing lengths. Conversely, in a low density region we have fewer particles with large smoothing lengths. When two such regions are next to each other (which is a common occurrence in star formation simulations), we can get a high density, short smoothing length particle having a Voronoi neighbour which is of low density and large smoothing length. Since the wall of a Voronoi cell bisects the distance between two neighbouring generating sites, this can result in the high density particle being placed in a cell which is much larger than its smoothing length. Overall this results in a fuzziness at the boundary between a low density and a high density region (see Figure 3.5 and 3.6), as pointed out by Koepferl et al. (2016).

Not being able to achieve a sharp boundary between high and low density regions can result in significant numerical noise and even shadowing effects when performing MCRT.

For this reason I propose an alternative method for choosing the Voronoi grid generating sites.

3.4.2 Increasing the resolution

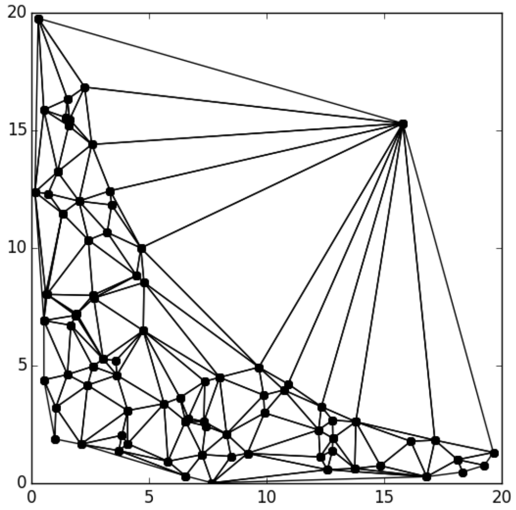

Figure 3.4: 2D Delaunay triangulation for a set of data with a large density gradient. We can notice the elongated triangles in the outskirts of the high density region with sides significantly longer than the typical distances between the tightly packed generating sites.

In order to improve the resolution of a Voronoi grid when large density gradients are present in the SPH dataset, I propose a simple algorithm for inserting a small number of extra generating sites where needed. The algorithm uses the Delaunay triangulation and follows these steps:

1. Loop over all Delaunay triangles (tetrahedra).

2. For each side (edge) of a triangle (tetrahedron) connecting generating sites i and j, calculate its length l_{ij}.

3. Compare l_{ij} to the threshold lengths d_i and d_j associated with i and j.

4. If $l_{ij}/2 > d_i$ add a generating site located on the side (edge) of the triangle (tetrahedron) at distance $2d_i$ from generating site i. The same applies for generating site j and the distance d_j.

5. Recompute all Voronoi cells affected by the generating site insertion.

The above algorithm ensures that the maximum distance from a Voronoi cell generating site to its cell wall is d_i, which can now be tied to the SPH particle smoothing length h_i.

I have implemented the cell insertion algorithm in 2D within my own Voronoi construction algorithm and in 3D by using some of the pre-existing methods of VORO++. In 2D, steps 4 and 5 are trivial since the construction algorithm is incremental and is designed to insert a new generating site into an already constructed tessellation. In 3D, VORO++ also allows for the easy addition of new generating sites since each cell is computed individually every time it is needed. The cell insertion algorithm is quick and easy to run, with computing time depending roughly linearly on the number of cells.

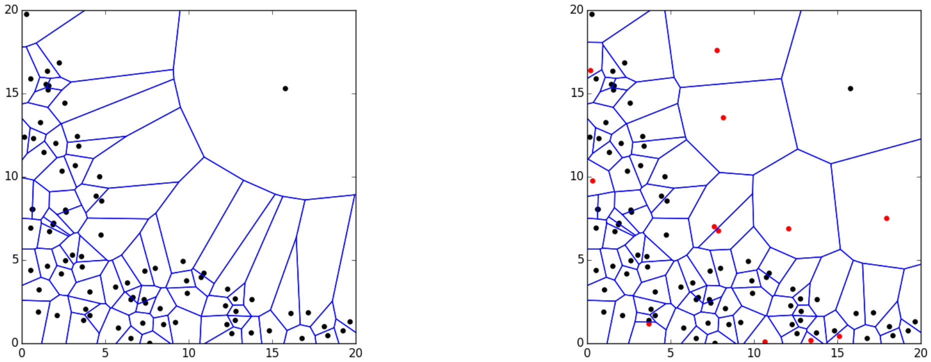

Figure 3.5: *Left:* 2D Voronoi grid of the dataset from Figure 3.4. *Right:* 2D Voronoi grid with added 12 extra generating sites (in red) by using the cell insertion algorithm described in Section 3.4.2.

In order to test the algorithm in 2D I have generated my own test case consisting of an arch of 80 tightly placed generating sites and a single generating site positioned away from them (see Figure 3.4). All of the generating sites were given smoothing length equal to the distance to their third closest neighbour. This only roughly resembles the way in which SPH smoothing lengths are calculated, however it is sufficient for a proof of concept test.

The outcome of the cell insertion algorithm for the 2D test is shown in Figure 3.5. It is easy to notice that originally the Voronoi cells at the boundary between the arch and the stand-alone generating site were too elongated, however by employing the algorithm with $d_i = h_i$, and inserting only 12 extra generating sites (shown in red) there is a significant improvement.

A similar test was then performed in 3D, this time on a section of an actual SPH dataset of a supernova explosion (Lucas et al. (in prep.)). This simulation contains shocks and complex density structures at many size scales which makes is a perfect test case. I have selected a region of large density gradient, containing 42568 particles out of 208155, and I have again assumed that $d_i = h_i$.

The outcome of the 3D cell insertion algorithm is shown in Figure 3.6. I have plotted a slice through the dataset at $z = -1$ for ease of visualisation, and the cells are colour coded in terms of their average density. In this case it is even easier to spot the density fuzziness at the boundary. Once again the cell insertion algorithm significantly improves the grid resolution by adding only 2202 (or about 5%) extra generating sites.

3.4.3 Using Lloyd's algorithm

An alternative to the cell-insertion algorithm would be to apply Lloyd's algorithm to the Voronoi tesselation built around the SPH particle positions (Lloyd (1982)). This is an iterative algorithm for regularising the Voronoi cell shapes, which is commonly used by the moving mesh hydrodynamics codes (see Springel (2010), Duffell and MacFadyen (2011), Vandenbroucke and De Rijcke (2016)). At each iteration of the algorithm we find the centroid of all Voronoi cells and we move all of the generating sites to the locations of the centroids of their respective cells. This naturally neutralises the cell elongation at the regions of high density gradient.

I have applied Lloyd's algorithm to the same 3D SPH dataset as the one used in Section 3.4.2. I used 5 iterations and the result is presented in Figure 3.7.

We can see that Lloyd's algorithm has a similar effect of helping to resolve the cloud boundary as the cell-insertion algorithm. In some ways Lloyd's algorithm has the advantage of producing cells with potentially fewer walls and vertices, which uses less memory and can speed up the mapping of SPH density onto the cells (see Chapter 4). An additional minor advantage is the fact that we would not need to increase the number of generating sites. Overall the algorithm is ideal for improving the resolution for the purpose of post-processing with MCRT.

The main disadvantage of Lloyd's algorithm is that the Voronoi tessellation needs to be reconstructed at each iteration. For large datasets this introduces a substantial extra computational cost, and in the case of performing live radiation hydrodynamics the Voronoi grid construction is sometimes one of the most time-consuming steps, as will be shown in Chapter 5.

Furthermore, it is unclear if using Lloyd's algorithm would not create blurring of the parameters (such as temperature and ionisation fraction) that are mapped back to the SPH particles during live radiation hydrodynamics. Unlike the cell-insertion algorithm which retains information of the particles' positions, the application of Lloyd's algorithm can lose some of this information and result in some particles having multiple cells that contain significant fractions of their mass. If as a result of MCRT these cells have vastly different temperatures (which is common occurrence in ionisation problems), then the temperature that is mapped back to the particle will have an in-between value, which might not be desirable. Further tests are therefore needed in order to establish the applicability of Lloyd's algorithm to live radiation hydrodynamics problems.

3.4.4 Using a Radical Voronoi tessellation

Another way of addressing the resolution issues around the high and low density boundaries is by altering the geometry of the individual Voronoi cells. Instead of bisecting the distance between neighbouring generating sites, we can instead split the distance with a given ratio.

Simply assigning weights to the generating sites to construct the cell walls at certain distance ratios is the simplest idea that comes to mind. Unfortunately this results in cells which are no longer polyhedra, not necessarily convex and may have gaps (Aurenhammer and Edelsbrunner (1984)), neither of which are desirable.

A better way to achieve a weighted outcome is by using a Radical Voronoi grid (also called Power diagram). Instead of altering the distance criterion to generating sites multiplicatively (i.e. choosing which cell a point in space belongs to by finding the minimum of the distances to generating sites divided by their weights), a Radical Voronoi grid changes it additively. This means that each generating site is assigned an individual radius (r_i for cell i), and a point in space belongs to the cell for which $d_i^2 - r_i^2$ is minimised, where d_i is the distance to the generating site (Voronoi (1908)). A grid structure following this property has polyhedral, non-overlapping cells, similar to those of a regular Voronoi grid (Aurenhammer (1987)). In fact, the regular Voronoi grid is the special case of a Radical Voronoi grid for which all r_i are equal.

I have tested the output of the Radical Voronoi grid algorithm included in VORO++

for the same dataset as the one used in Section 3.4.2. I have assumed the radius of each generating site to be equal to the smoothing length of its corresponding SPH particle, and the results are shown in Figure 3.8. As we can see, there is a small improvement over the unaltered, standard Voronoi grid from Figure 3.6, however the improvement is not as effective as adding extra generating sites. It is worth pointing out that a Radical Voronoi tessellation does not always have as many cells as the number of generating sites provided, since some generating sites may not be assigned any points in space. In this particular case 42530 cells were generated, as opposed to the expected 42568.

Finally, we need to address how an MCRT method could be performed onto a Radical Voronoi grid. In order to propagate photon packets through this grid structure we can follow the same procedure as in Section 3.3 with only the following modification in equation 3.4:

$$\mathbf{p}_{ij} = \frac{w_j \mathbf{p}_i + w_i \mathbf{p}_j}{w_i + w_j}, \tag{3.7}$$

where $w_i = l_{ij}^2 + r_i^2 - r_j^2$ and $w_j = l_{ij}^2 + r_j^2 - r_i^2$.

The ease with which an MCRT code can be adapted to use a Radical Voronoi grid, together with the fact that a grid construction algorithm has already been implemented in the library VORO++ makes this a potentially viable alternative to a regular Voronoi grid for certain datasets.

3.4.5 Lowering the resolution

While in some cases we find it necessary to locally increase the Voronoi grid resolution, there may be regions of the simulation where there is no need to resolve each SPH particle. Decreasing the number of generating sites reduces the computation time, which is always a limiting factor in numerical simulations.

We may want to lower the number of generating sites if we model radiation and we have part of the simulation volume which is very far away from the source and is sufficiently represented by a single cell. In order to locally lower the resolution, we can make use of the particle tree structure that is typically constructed and maintained in an SPH code, and employ the opening angle criterion which is used for computing gravitational forces (see Clark et al. (2012)). This can allow us to replace a group of neighbouring particles by the position of their central point, if their combined angular size, as seen from the radiation source, is smaller than a given value.

Another situation in which we may want to lower the resolution is if we are modelling ionising radiation and there is a region which is completely shielded by incoming rays. This is a common occurrence, however, these regions can be difficult to identify in advance.

Using the techniques described above, we can adjust the Voronoi grid for an optimal representation of an SPH dataset. As the next step after constructing the grid structure, we need to consider the question of how to map SPH properties, such as mass density, onto the grid cells. I have explored this question thoroughly in the following chapter.

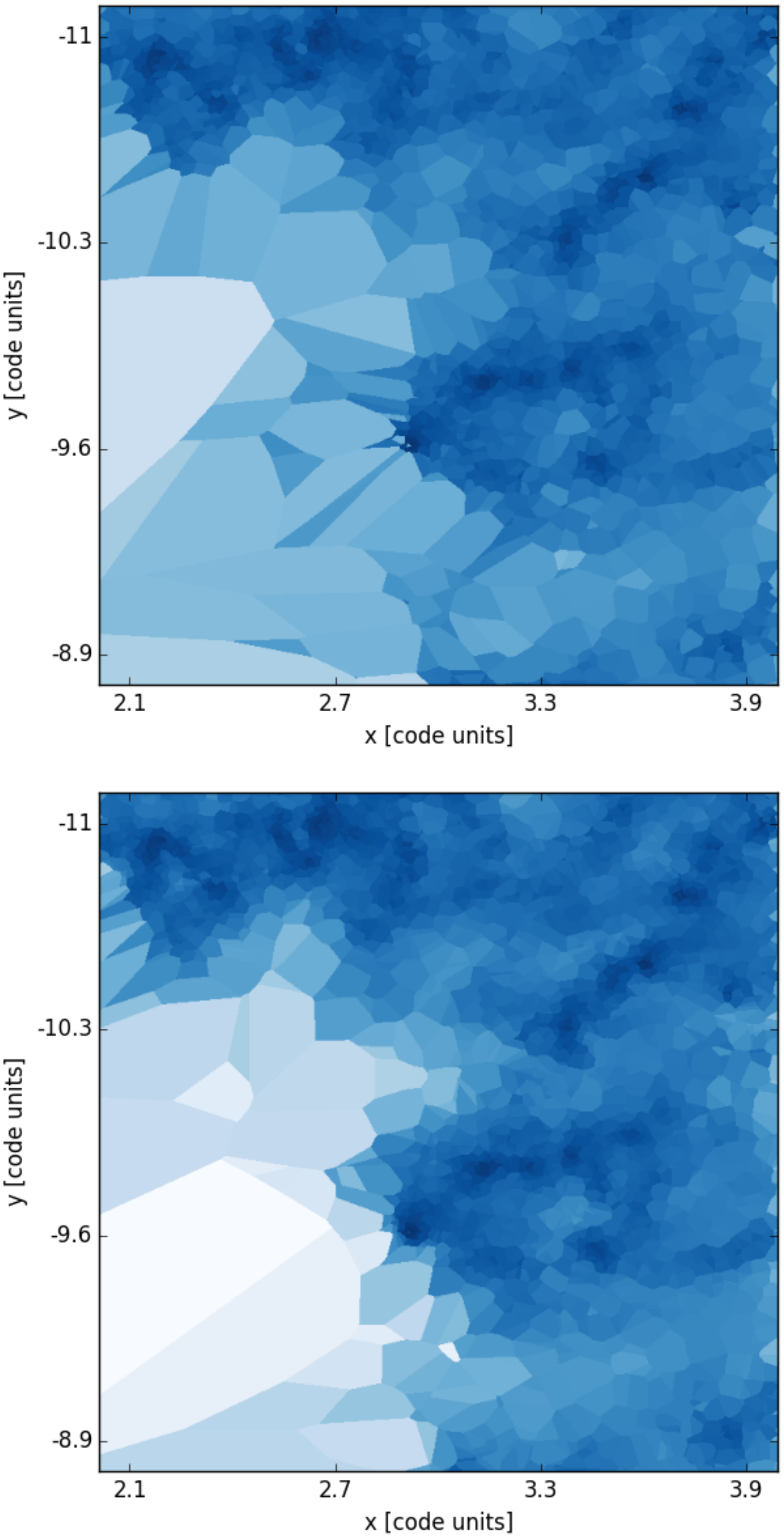

Figure 3.6: Slices through the 3D Voronoi grid representation of a part of an SPH dataset at $z = -1$. *Top:* the Voronoi grid was constructed using one cell per SPH particle. *Bottom:* the grid was modified by the cell-insertion algorithm described in Section 3.4.2. The colour corresponds to cell density with darker blue being higher density. The density was calculated via the method developed in Chapter 4 (see also Petkova et al. (2018)).

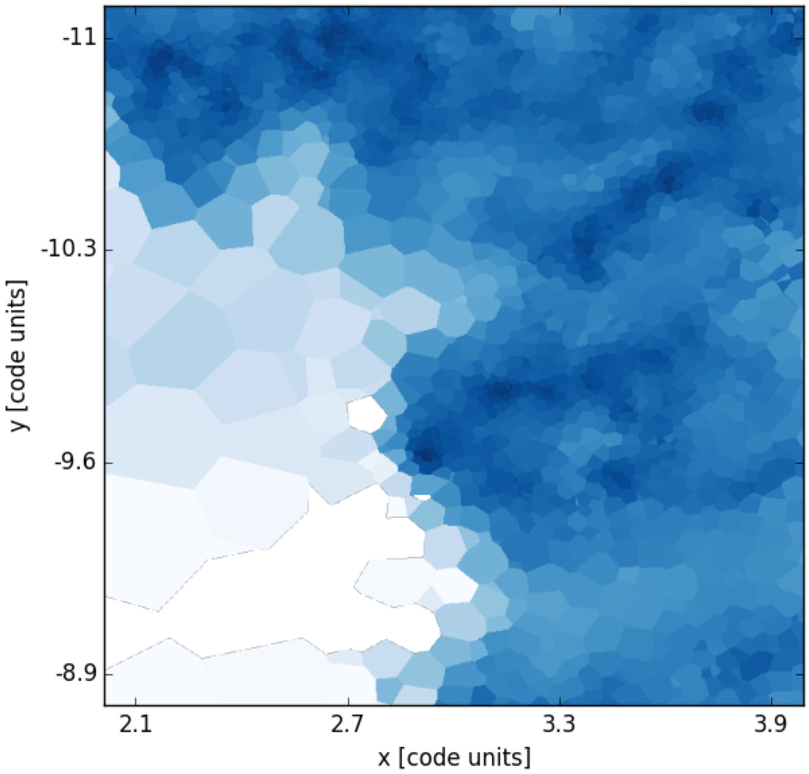

Figure 3.7: A slice through the 3D Voronoi grid representation of a part of an SPH dataset at $z = -1$. The same starting configuration was used as in Figure 3.6, followed by 5 iterations of Lloyd's algorithm. The colour corresponds to cell density with darker blue being higher density. The density was calculated via the method developed in Chapter 4 (see also Petkova et al. (2018)).

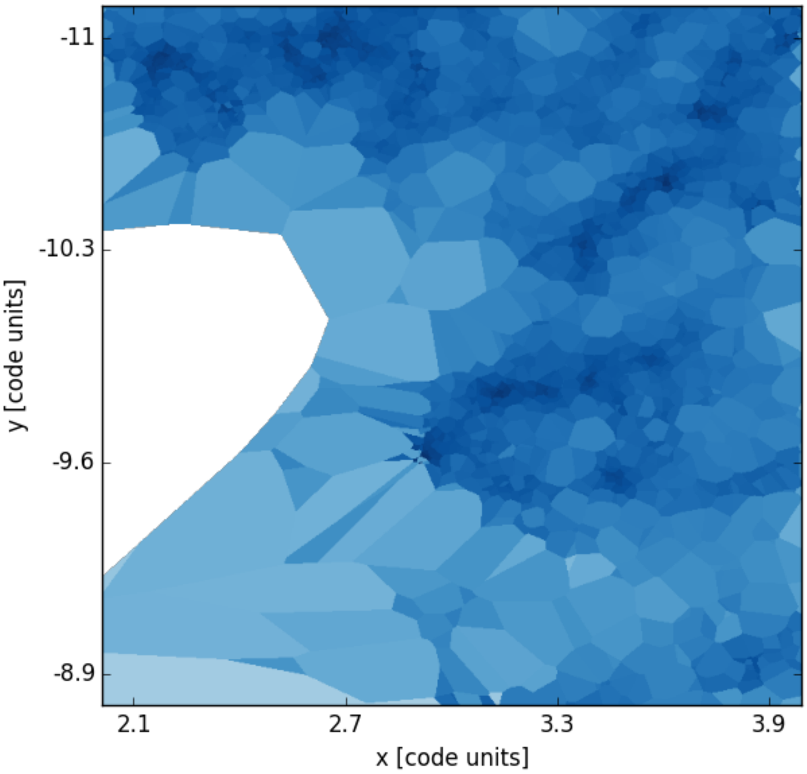

Figure 3.8: A slice through the 3D Radical Voronoi grid representation of a part of an SPH dataset at $z = -1$. The cell generating sites were assigned radii equal to the smoothing lengths of their corresponding SPH particles. Only 42530 cells could be constructed using this setup, as opposed to the 42568 used in the standard Voronoi grid (see Figure 3.6). The colour corresponds to cell density with darker blue being higher density. The density was calculated via the method developed in Chapter 4 (see also Petkova et al. (2018)).

Chapter 4

Mapping SPH Densities onto a Voronoi Grid

This chapter will present the methodology of how to map SPH particles onto a Voronoi grid for the purposes of either post-processing a hydrodynamics simulation with a radiative transfer code, or running live radiation hydrodynamics. In order to do so, I will explore the link between the SPH kernel function and the Voronoi cell density, by integrating the former over the area (in 2D) or volume (in 3D) of the cell. For computational efficiency, I will derive analytic expressions for these integrals, as presented in Petkova et al. (2018).

Furthermore, I will perform numerical tests of the derived solution, I will demonstrate how the results can be tabulated to further improve the performance, and I will discuss how to expand the applicability of the derivation to a broad range of SPH kernel functions.

4.1 Calculating densities of Voronoi grid cells

4.1.1 The SPH kernel function and Voronoi cell density

SPH uses discrete particles in order to represent the continuous density distribution of a fluid, as discussed in Chapter 2. The exact reconstruction of this continuous density profile from the particle positions is a major topic in SPH (Price (2012)). The density estimate at a given position $\mathbf{r}$ is typically determined by the masses and positions of local particles. Furthermore, particles that are closer to $\mathbf{r}$ have a greater effect on the the density $\rho(\mathbf{r})$, and this effect is quantified by the SPH kernel function, W, which peaks at a particle's position and decreases as a function of radius from the particle (see Figure 4.1). The density at $\mathbf{r}$ is therefore given by the sum over all (N) particles:

$$\rho(\mathbf{r}) = \sum_{j=1}^{N} m_j W(|\mathbf{r} - \mathbf{r}_j|, h_j), \tag{4.1}$$

where m_j is the mass of the j-th particle, $\mathbf{r}_j$ is its position, and h_j is known as the smoothing length and it indicates a length scale for the density influence of the j-th particle.

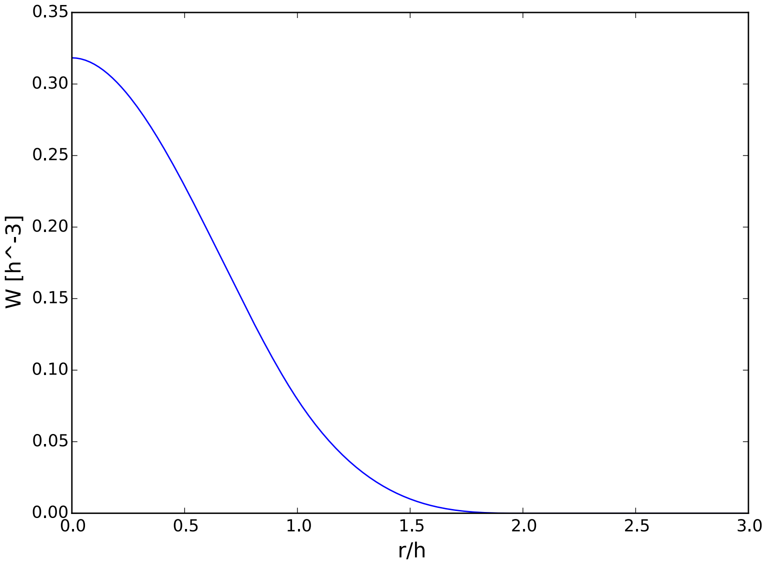

Figure 4.1: The 3D cubic spline kernel function as a function of radius (see equation 4.37).

Typically we would like only particles in the vicinity of $\mathbf{r}$ to contribute to the density estimate, and hence W is chosen to be a function with *compact support*, i.e. W becomes zero beyond a certain value of $|\mathbf{r} - \mathbf{r}_j|$, which is expressed in terms of h_j (Price (2012)). As an example, the cubic spline kernel function shown in Figure 4.1 becomes zero beyond a radial distance of $2h_j$.

When constructing a Voronoi grid, we need to assign a density value to each cell, in order to adequately represent the continuous density distribution contained in an SPH simulation. So far in the scientific community this task has been done in three different ways by:

1. Dividing the mass of an SPH particle by the volume of the cell surrounding it;

2. Computing the SPH density estimate at the centroid of the cell of interest;

3. Or sampling many points within a cell and averaging their SPH density estimates (Koepferl et al. (2016)).

Of the above, method 1 conserves the total mass through the density mapping, however it assumes that a particle is the sole contributor to the density of its associated cell, which produces different local density than the SPH estimator, especially since a particle's smoothing length and the size of its associated cell are unrelated quantities. This density inaccuracy has been observed and presented by Koepferl et al. (2016). Furthermore, this method can only be applied to situations when each SPH particle is mapped to a single cell, and I have already discussed why this might not be desired in Chapter 3.

Method 2 takes into account the SPH density estimates when approximating the Voronoi cell density, and hence is likely to produce more accurate local values than method 1. It also has the advantage of being applicable to setup with multiple cells per SPH particle. However, overall method 2 cannot ensure mass conservation, which affects the results of radiative transfer performed on the Voronoi grid.

Method 3, on the other hand, can ensure local and global accuracy, as well as enabling multiple cells per particle, provided that a sufficient number of points are sampled within each cell. While methods 1 and 2 are computationally quick to apply, method 3 takes a lot more time and computing power.

In this work I propose and develop a new density mapping approach as an alternative to all of the above. This approach employs the SPH kernel function and is presented in Petkova et al. (2018). Here I will go over it again, expanding on some details.

Let us consider the mass contained within the i-th Voronoi cell, M_i. We can obtain this mass by integrating the density inside the volume of the cell (V_i):

$$\begin{aligned} M_i &= \int_{V_i} \rho(\mathbf{r}')\mathrm{d}V' && (4.2)\\ &= \int_{V_i} \sum_{j=1}^{N} m_j W(|\mathbf{r}' - \mathbf{r}_j|, h_j)\mathrm{d}V' && (4.3)\\ &= \sum_{j=1}^{N} m_j \int_{V_i} W(|\mathbf{r}' - \mathbf{r}_j|, h_j)\mathrm{d}V'. && (4.4) \end{aligned}$$

By rearranging the expression for M_i, as shown above, we can think of the cell mass as a sum of "mass contributions" from the neighbouring particles. In order to obtain each mass contribution, we need to integrate W inside the volume of the cell and multiply it by the particle's mass, m_j. Once we compute M_i and divide it by V_i, we will have the average density of the cell.

Therefore the mathematical problem that I will be solving becomes how to integrate W inside the volume of a random polyhedron.

4.1.2 Ensuring mass conservation with the mapping

Since mass conservation is crucial for the purposes of performing radiative transfer on SPH data, here I will demonstrate that my proposed density mapping ensures it. The total mass of the Voronoi cells can be written as the following sum:

$$\begin{aligned} \sum_{i=1}^{N'} M_i &= \sum_{i=1}^{N'} \sum_{j=1}^{N} m_j \int_{V_i} W(|\mathbf{r}' - \mathbf{r}_j|, h_j)\mathrm{d}V' && (4.5)\\ &= \sum_{j=1}^{N} m_j \sum_{i=1}^{N'} \int_{V_i} W(|\mathbf{r}' - \mathbf{r}_j|, h_j)\mathrm{d}V', && (4.6) \end{aligned}$$

where N' is the number of cells.

A fundamental property of the kernel function is that it is normalised when integrated over all space, which can be expressed as:

$$\int_{V} W(|\mathbf{r}' - \mathbf{r}_j|, h_j)\mathrm{d}V' = 1. \qquad (4.7)$$

Since the Voronoi cells are non-overlapping and together they cover the full simulation volume, we can write that:

$$\sum_{i=1}^{N'} \int_{V_i} W(|\mathbf{r}' - \mathbf{r}_j|, h_j) \mathrm{d}V' = \int_V W(|\mathbf{r}' - \mathbf{r}_j|, h_j) \mathrm{d}V' = 1. \tag{4.8}$$

The above equations imply that $\sum_{i=1}^{N'} M_i = \sum_{j=1}^{N} m_j$, i.e. the sum of all cell masses equals the sum of all particle masses, and hence mass is conserved.

4.1.3 Setup of the mathematical problem

In order to develop a density mapping method which is both accurate and computationally fast, I will integrate the SPH kernel function analytically within the polyhedral volume of a Voronoi cell.

For simplicity let us drop the i and j indices from the previous equations and consider the case of a single SPH particle mapped onto a grid. Let us also choose a coordinate system, such that the particle is at the origin. The mathematical problem can then be expressed as finding I_V from:

$$I_V = \int_{\mathcal{V}} W(r) \mathrm{d}V, \tag{4.9}$$

where $\mathcal{V}$ is the space enclosed by the Voronoi cell of interest. Note that in this general case the cell may not necessarily contain the origin.

In order to integrate the above expression I have loosely followed the steps outlined in Mirtich (1996) with some modifications, which I added to account for the spherically symmetric nature of W and the flatness of the Voronoi cell walls. First, I will apply the Divergence theorem in order to express the volume integral I_V as a sum of surface integrals integrated over the cell walls. Second, for each surface integral, I_S, I will apply Green's theorem and write it as a sum of line integrals, following the contour of a wall's face. Finally, a line integral, I_L, will be integrated once again between the two vertices that define it to give us the final expression I_P, which when evaluated at each vertex of each edge of the polyhedron, and added up, gives us I_V.

For the specific derivation in this chapter I have chosen to use a cubic spline kernel function. Its simple to integrate form and its common use within the astronomical community (Price (2012)) made it the most suitable choice. The same integration, however, can be performed with a range of different kernel functions, and I will discuss this further at the end of the chapter.

4.2 Analytic integration of a 2D cubic spline kernel function

Before tackling the 3D integral of a kernel function, let us first explore the problem in 2D space. Here we also consider a particle at the origin of a coordinate system, but the Voronoi cell is a polygon instead of a polyhedron, and the problem of interest becomes:

$$I_S = \int_{\mathcal{A}} W(r)\mathrm{d}A. \tag{4.10}$$

In the above $\mathcal{A}$ is the area of the cell of interest and we can assume that a cell is defined by the coordinates of its vertices.

The cubic spline kernel function in 2D is given by:

$$W(r) = \frac{10}{7h^2\pi}\begin{cases} 1 - 1.5\left(\frac{r}{h}\right)^2 + 0.75\left(\frac{r}{h}\right)^3, & r \leq h; \\ 0.25\left(2 - \left(\frac{r}{h}\right)\right)^3, & h \leq r \leq 2h; \\ 0, & r \geq 2h. \end{cases} \tag{4.11}$$

4.2.1 Applying Green's Theorem

The difficulty in integrating equation 4.10 comes from not being able to set limits for the integral in any coordinate system due to the unspecified shape of the Voronoi cell. Instead of trying to perform the integration directly, we will apply Green's Theorem, which can be written as:

$$\int_{\mathcal{A}} \nabla \cdot \mathbf{H}\mathrm{d}A = \int_{\partial\mathcal{A}} \mathbf{H} \cdot \hat{\mathbf{m}}\mathrm{d}l. \tag{4.12}$$

In the above expression $\partial\mathcal{A}$ is the contour of the area of integration $\mathcal{A}$, and $\hat{\mathbf{m}}$ is the unit vector normal to that contour.

The benefit of employing Green's Theorem is that instead of integrating W over an area, we will construct a new function which will only need to be integrated along a line. This way we reduce the problem by one dimension.

The left hand side of equation 4.12 has the same form as the expression in equation 4.10, and hence in order to apply the theorem I will construct a function $\mathbf{H}$, such that $\nabla \cdot \mathbf{H} = W$.

For mathematical simplicity, and since W is azimuthally symmetric, I choose a vector function of the form $\mathbf{H} = H_r \hat{\mathbf{r}}$. The kernel function can then be expressed as:

$$W(r) = \frac{1}{r} \frac{\partial (r H_r)}{\partial r}. \tag{4.13}$$

By rearranging and integrating, I obtain the following solution for H_r:

$$H_r(r) = \frac{1}{r} \frac{5}{7h^2\pi} \begin{cases} r^2 - \frac{3}{4h^2} r^4 + \frac{3}{10h^3} r^5 + D_1, & r \leq h; \\ 2r^2 - \frac{2}{h} r^3 + \frac{3}{4h^2} r^4 - \frac{1}{10h^3} r^5 + D_2, & h \leq r \leq 2h; \\ D_3, & r \geq 2h, \end{cases} \tag{4.14}$$

where D_1, D_2 and D_3 are constants of integration.

In order to determine the values of the constants I will enforce continuity of $\mathbf{H}$ at $r = h$ and $r = 2h$, which creates the relations:

$$D_1 = D_2 + \frac{h^2}{10}; \qquad D_2 = D_3 - \frac{4h^2}{5}. \tag{4.15}$$

As a final constraint on the constants of integration I will use the normalisation property of W. Let us consider a circle of radius $r_{circle} > 2h$, which is centred at the origin. Since W drops to zero for $r \geq 2h$, W integrates to 1 within the selected circle, and so does $\nabla \cdot \mathbf{H}$ by construction. Therefore applying Green's theorem for the circle, we can write:

$$\int_{\mathcal{A}_{circle}} \nabla \cdot \mathbf{H} \mathrm{d}A = \int_{\partial \mathcal{A}_{circle}} \mathbf{H} \cdot \hat{\mathbf{m}} \mathrm{d}l = 1. \tag{4.16}$$

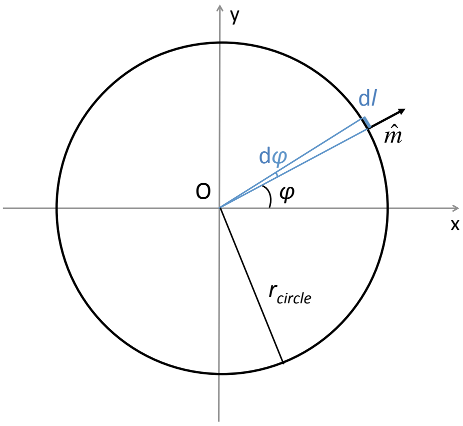

Figure 4.2: Setup for applying Green's Theorem to an integral inside of a circle with radius r_{circle}.

As shown in Figure 4.2, for the case of the circle $\hat{\mathbf{m}} = \hat{\mathbf{r}}$ and $\mathrm{d}l = r_{circle}\mathrm{d}\phi$. Therefore equation 4.16 becomes:

$$\int_{\partial\mathcal{A}_{circle}} \mathbf{H}\cdot\hat{\mathbf{m}}\mathrm{d}l = \int_{\phi=0}^{\phi=2\pi} H_r(r_{circle})r_{circle}\mathrm{d}\phi = \int_{\phi=0}^{\phi=2\pi} \frac{5}{7h^2\pi}D_3\mathrm{d}\phi = 1. \tag{4.17}$$

By combining equations 4.15 and 4.17 the constants of integration are found to be:

$$D_1 = 0; D_2 = -\frac{1}{10}h^2; D_3 = \frac{7}{10}h^2. \tag{4.18}$$

4.2.2 Contour integral along a line segment

Let us now return to the area and contour of the polygon of interest. A polygonal cell has line segments for walls, and hence the contour integral of a cell becomes a sum of integrals over line segments. For that reason I will focus on demonstrating how to apply Green's Theorem for a single line segment, $\mathcal{L}$:

$$I_L = \int_{\mathcal{L}} \mathbf{H}\cdot\hat{\mathbf{m}}\mathrm{d}l. \tag{4.19}$$

Without loss of generality we can rotate the coordinate system so that the cell wall of interest is perpendicular to the $\phi = 0$ ray. Let the orthogonal distance from the origin to the wall be r_0 (see Figure 4.3).

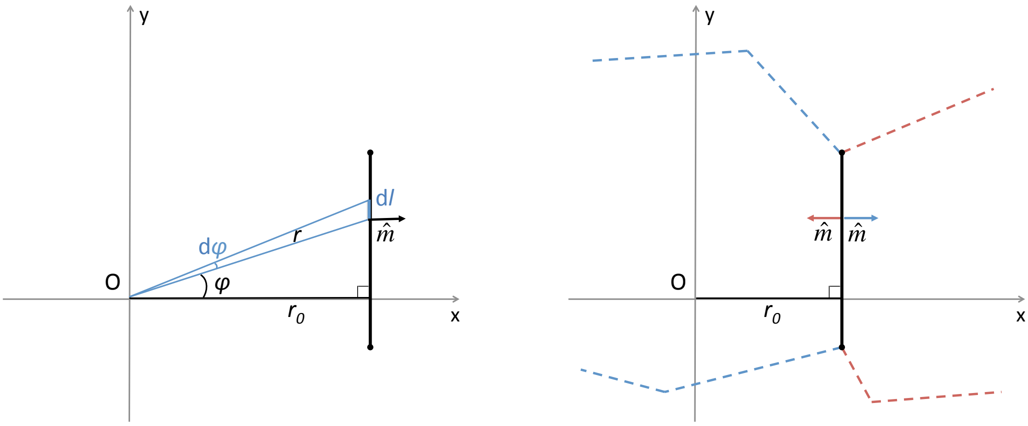

Figure 4.3: *Left*: setup for integrating **H** along a line segment. *Right*: schematic of the different signs of $\hat{m}$, depending on the orientation of the rest of the polygon.

The vector normal to the line segment can be written as:

$$\hat{\mathbf{m}} = \pm\hat{\mathbf{x}} = \pm(\cos\phi\hat{\mathbf{r}} - \sin\phi\hat{\boldsymbol{\phi}}), \tag{4.20}$$

where the sign depends on whether the rest of the cell sits on the left (+) or on the right (−) of the line segment (see Figure 4.3). Therefore the dot product from Green's Theorem becomes:

$$\mathbf{H}\cdot\hat{\mathbf{m}} = \pm H_r\cos\phi. \tag{4.21}$$

In order to express dl, I will use a vector representation given by:

$$\mathrm{d}\mathbf{l} = \mathrm{d}r\hat{\mathbf{r}} + r\mathrm{d}\boldsymbol{\phi}\hat{\boldsymbol{\phi}}. \tag{4.22}$$

Additionally, we can also write d**l** as:

$$\mathrm{d}\mathbf{l} = \mathrm{d}l\hat{\mathbf{y}} = \mathrm{d}l\sin\phi\hat{\mathbf{r}} + \mathrm{d}l\cos\phi\hat{\boldsymbol{\phi}}. \tag{4.23}$$

By comparing the $\hat{\boldsymbol{\phi}}$ terms we have that:

$$\mathrm{d}l = \frac{r\mathrm{d}\phi}{\cos\phi}. \tag{4.24}$$

Therefore for a line segment we need to integrate the following

$$\mathbf{H}\cdot\hat{\mathbf{m}}\mathrm{d}l = \pm H_r r\mathrm{d}\phi, \tag{4.25}$$

where

$$r = \frac{r_0}{\cos\phi}, \tag{4.26}$$

and

$$H_r r = \frac{5}{7h^2\pi}\begin{cases} \left(\frac{r_0}{\cos\phi}\right)^2 - \frac{3}{4h^2}\left(\frac{r_0}{\cos\phi}\right)^4 + \frac{3}{10h^3}\left(\frac{r_0}{\cos\phi}\right)^5, & \frac{r_0}{\cos\phi}\leq h;\\ 2\left(\frac{r_0}{\cos\phi}\right)^2 - \frac{2}{h}\left(\frac{r_0}{\cos\phi}\right)^3 + \frac{3}{4h^2}\left(\frac{r_0}{\cos\phi}\right)^4 - \frac{1}{10h^3}\left(\frac{r_0}{\cos\phi}\right)^5 - \frac{1}{10}h^2, & h\leq\frac{r_0}{\cos\phi}\leq 2h;\\ \frac{7}{10}h^2, & \frac{r_0}{\cos\phi}\geq 2h. \end{cases} \tag{4.27}$$

4.2.3 Calculating the line integral analytically

Let us for now only consider the positive sign of equation 4.25. The significance of the signs will be discussed in more detail later in this section.

We have now established a final integral which needs to be computed between two values of ϕ. In order to perform this integration, I will use the fact that $H_r r$ is a polynomial consisting of different powers of $\frac{1}{\cos\phi}$, and hence I will compute the following terms:

$$J_n = \int\frac{\mathrm{d}\phi}{\cos^n\phi}, \tag{4.28}$$

where $n\in\mathbb{Z}$, $n\geq 0$.

The expression J_0 is trivial and can be written as:

$$J_0 = \int\mathrm{d}\phi = \phi + C. \tag{4.29}$$

For the remaining even powers ($n = 2k$, $k\in\mathbb{N}$) we can use the following formula from Gradshteyn and Ryzhik (2007):

$$J_{2k} = \int \frac{\mathrm{d}\phi}{\cos^{2k}\phi} = \frac{\sin\phi}{2k-1}\left(\sec^{2k-1}\phi + \sum_{p=1}^{k-1} \frac{2^p(k-1)(k-2)...(k-p)}{(2k-3)(2k-5)...(2k-2p-1)}\sec^{2k-2p-1}\phi\right)+C. \tag{4.30}$$

Applied to the specific powers of interest, the formula leads to:

$$J_2 = \tan\phi + C; \tag{4.31}$$

$$J_4 = \frac{1}{3}\tan\phi\left(\sec^2\phi + 2\right) + C. \tag{4.32}$$

Similarly, the odd powers ($n = 2k+1,\ k \in \mathbb{N}$) can also be expressed with a formula from Gradshteyn and Ryzhik (2007), namely:

$$J_{2k+1} = \int \frac{\mathrm{d}\phi}{\cos^{2k+1}\phi} = \frac{\sin\phi}{2k}\left(\sec^{2k}\phi + \sum_{p=1}^{k-1} \frac{(2k-1)(2k-3)...(2k-2p+1)}{2^p(k-1)(k-2)...(k-p)}\sec^{2k-2p}\phi\right) + \frac{(2k-1)!!}{2^k k!}\ln\tan\left(\frac{\pi}{4}+\frac{\phi}{2}\right) + C. \tag{4.33}$$

This formula gives us the expressions for $n = 3$ and $n = 5$ to be:

$$J_3 = \frac{1}{2}\tan\phi\sec\phi + \frac{1}{2}\ln\tan\left(\frac{\pi}{4}+\frac{\phi}{2}\right) + C; \tag{4.34}$$

$$J_5 = \frac{1}{4}\tan\phi\sec\phi\left(\sec^2\phi + \frac{3}{2}\right) + \frac{3}{8}\ln\tan\left(\frac{\pi}{4}+\frac{\phi}{2}\right) + C; \tag{4.35}$$

The final solution is then:

$$I_P = \int H_r(r) r \mathrm{d}\phi = \frac{5r_0^2}{7h^2\pi} \begin{cases} J_2 - \frac{3}{4}\left(\frac{r_0}{h}\right)^2 J_4 + \frac{3}{10}\left(\frac{r_0}{h}\right)^3 J_5, & \cos\phi \geq \frac{r_0}{h}; \\ 2J_2 - 2\frac{r_0}{h} J_3 + \frac{3}{4}\left(\frac{r_0}{h}\right)^2 J_4 - \frac{1}{10}\left(\frac{r_0}{h}\right)^3 J_5 - \frac{1}{10}\left(\frac{r_0}{h}\right)^{-2} J_0, & \frac{r_0}{2h} \leq \cos\phi \leq \frac{r_0}{h}; \\ \frac{7}{10}\left(\frac{r_0}{h}\right)^{-2} J_0, & \cos\phi \leq \frac{r_0}{2h}. \end{cases} \tag{4.36}$$

Note that the J_n expressions all contain constants of integration. Similarly to before I have enforced continuity of the function when using a computer implementation of this solution, however the mathematical expressions of the constants of integration are too lengthy to be included here.

4.2.4 Geometric interpretation of the solution

Let us return to the statement that equation 4.10 is difficult to integrate due to inability to define integration limits, and consider what Green's Theorem does in order to bypass this issue.

When integrating $\mathbf{H}$ along a line segment, the answer we get is equivalent to integrating W inside the triangle formed by the line segment and the origin (see Figure 4.4). As an example, the integral of $\mathbf{H}$ along AB, as shown in Figure 4.4, is equivalent to integrating W over the area of AOB. The same is true for all other cell walls of ABCDE, and hence I_S is obtained by dividing the Voronoi cell into triangular segments and integrating them separately.

Furthermore, evaluating I_P at A is equivalent to integrating W over the area of AOP, thus the final step of our derivation splits the cell into even smaller triangles.

It is worth pointing out that the negative sign of equation 4.25 becomes relevant when the Voronoi cell does not contain the origin (i.e. the particle). Since $\hat{\mathbf{m}}$ points outwards from the cell walls, the integral of $\mathbf{H}$ along AB and BC in the example in Figure 4.4 is negative. This is necessary in order for the triangular breakdown of a cell to work in the general case.

When we integrate $\mathbf{H}$ along the segments CD, DE and EA, and add up the results

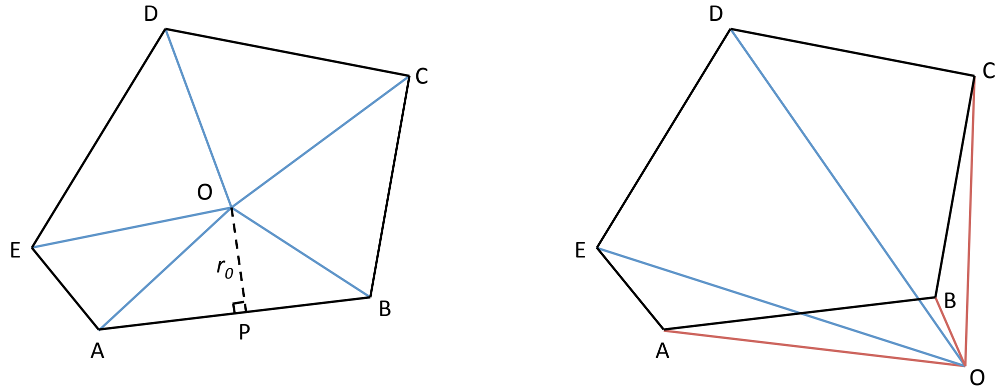

Figure 4.4: Breakdown of the integration area into triangles when the origin is internal (*left*) or external (*right*) to the polygonal cell.

we obtain the equivalent to integrating W over the area of AOCDE. Since the integral of $\mathbf{H}$ along AB and BC is negative, adding these two integrals to the rest is the same as subtracting the integral of W over AOCB from the integral of W over AOCDE. This gives us the equivalent of integrating W over the area of ABCDE, as desired.

4.3 Analytic integration of a 3D cubic spline kernel function

Having integrated W in the simpler, 2D case, we can now expand the derivation to 3D. The expression that we are trying to integrate is the one shown in equation 4.9.

Similarly to before I will use the cubic spline kernel, which in 3D is given by the following:

$$W(r) = \frac{1}{h^3\pi}\begin{cases} 1 - 1.5\left(\frac{r}{h}\right)^2 + 0.75\left(\frac{r}{h}\right)^3, & r \leq h; \\ 0.25\left(2 - \left(\frac{r}{h}\right)\right)^3, & h \leq r \leq 2h; \\ 0, & r \geq 2h. \end{cases} \tag{4.37}$$

4.3.1 Reducing the volume integral to a surface integral

As in the 2D case, here once again we can't perform the integration easily due to the difficulty in defining the integration limits. In order to progress with the solution first we will transform equation 4.9 into a surface integral, by using the Divergence Theorem:

$$\int_{\mathcal{V}} \nabla \cdot \mathbf{F} \mathrm{d}V = \int_{\partial\mathcal{V}} \mathbf{F} \cdot \hat{\mathbf{n}} \mathrm{d}S. \tag{4.38}$$

In the above $\hat{\mathbf{n}}$ is the unit vector normal to the surface of the volume of integration, and the surface itself is denoted by $\partial\mathcal{V}$.

It is easy to notice that the left hand side of equation 4.38 resembles equation 4.9. In order to apply the theorem, we will construct $\mathbf{F}$, such that $\nabla \cdot \mathbf{F} = W$, as we did in 2D. This is analogous to the relationship between charge and electric field in electrostatics.

Since W is spherically-symmetric, expressing $\mathbf{F}$ in spherical coordinates is the most suitable choice. Hence for $\mathbf{F}$ and its divergence we have:

$$\mathbf{F} = F_r\hat{\mathbf{r}} + F_\theta\hat{\boldsymbol{\theta}} + F_\phi\hat{\boldsymbol{\phi}}; \tag{4.39}$$

$$\nabla \cdot \mathbf{F} = \frac{1}{r^2}\frac{\partial(r^2 F_r)}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial(F_\theta \sin\theta)}{\partial\theta} + \frac{1}{\sin\theta}\frac{\partial F_\phi}{\partial\phi}. \tag{4.40}$$

Additionally, due to the spherical symmetry of the problem, we consider a function for which $F_\theta = F_\phi = 0$. This gives us the equation

$$\frac{1}{r^2}\frac{\partial(r^2 F_r)}{\partial r} = W(r), \tag{4.41}$$

which integrates to the following solution:

$$F_r(r) = \frac{1}{r^2}\frac{1}{h^3\pi}\begin{cases} \frac{1}{3}r^3 - \frac{3}{10h^2}r^5 + \frac{1}{8h^3}r^6 + C_1, & r \le h; \\ \frac{1}{4}\left(\frac{8}{3}r^3 - \frac{3}{h}r^4 + \frac{6}{5h^2}r^5 - \frac{1}{6h^3}r^6\right) + C_2, & h \le r \le 2h; \\ 0 + C_3, & r \ge 2h, \end{cases} \tag{4.42}$$

where C_1, C_2 and C_3 are constants of integration.

Similarly to the previous section, we enforce continuity of F_r at $r = h$ and $r = 2h$ to obtain the relationships:

$$C_1 = \frac{1}{60}h^3 + C_2; \qquad C_3 = \frac{4}{15}h^3 + C_2. \tag{4.43}$$

Finally, let us consider a sphere of radius $r_{sphere} \geq 2h$ and apply the Divergence Theorem to the integral of W over the volume of the sphere. For this setup we have that $\hat{\mathbf{n}} = \hat{\mathbf{r}}$, and $\mathrm{d}S = r_{sphere}^2 \sin\theta \mathrm{d}\theta \mathrm{d}\phi$.

Using the normalisation property of the kernel, and the fact that it is zero for $r \geq 2h$, we can write that:

$$\int_{\partial\mathcal{V}} \mathbf{F} \cdot \hat{\mathbf{n}} \mathrm{d}S = \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} F_r(r_{sphere}) r_{sphere}^2 \sin\theta \mathrm{d}\theta \mathrm{d}\phi = 1. \tag{4.44}$$

Equations 4.43 and 4.44 lead to the following solution for the constants of integration:

$$C_1 = 0; \quad C_2 = -\tfrac{1}{60}h^3; \quad C_3 = \frac{1}{4}h^3. \tag{4.45}$$

4.3.2 Calculating the surface integral on a plane

We have now transformed I_V into the surface integral given by equation 4.38. Since the surface of a polyhedron is a set of polygons, we will focus on integrating over only one of them. Namely, we wish to compute the integral

$$I_S = \int_{\mathcal{A}} \mathbf{F} \cdot \hat{\mathbf{n}} \mathrm{d}S, \tag{4.46}$$

where $\mathcal{A}$ is the region of space contained in a single polygonal wall.

Without loss of generality, we can choose the spherical coordinate system that $\mathbf{F}$ is defined in, so that the normal of the plane of the desired flat surface coincides with the $\theta = 0$ axis. Let the orthogonal distance from the origin to the plane be denoted by r_0.

We can express any surface element in spherical coordinates as

$$\mathrm{d}\mathbf{S} = \hat{\mathbf{n}} \mathrm{d}S = r^2 \sin\theta \mathrm{d}\theta \mathrm{d}\phi \hat{\mathbf{r}} + r \sin\theta \mathrm{d}r \mathrm{d}\phi \hat{\boldsymbol{\theta}} + r \mathrm{d}r \mathrm{d}\theta \hat{\boldsymbol{\phi}}, \tag{4.47}$$

and hence:

$$\mathbf{F}\cdot\hat{\mathbf{n}}\mathrm{d}S = \pm F_r r^2 \sin\theta \mathrm{d}\theta\mathrm{d}\phi. \tag{4.48}$$

In the above equation we have two options for the sign, depending on the orientation of the polyhedron. This is similar to equation 4.20 and Figure 4.3. For simplicity of notation we will only consider the positive sign in the derivations that are to follow.

Since we are integrating over θ and F_r is a function of $r(\theta)$, it would be necessary to pick a suitable integration variable. From the chosen coordinate system orientation we have the relationship:

$$r = \frac{r_0}{\cos\theta}. \tag{4.49}$$

Let $\mu = \cos\theta$, then $r = \frac{r_0}{\mu}$ and $\mathrm{d}\mu = -\sin\theta\mathrm{d}\theta$. This gives us the following expressions, which are simpler to work with:

$$\mathbf{F}\cdot\hat{\mathbf{n}}\mathrm{d}S = -F_r(\mu)\left(\frac{\mu}{r_0}\right)^{-2}\mathrm{d}\mu\mathrm{d}\phi; \tag{4.50}$$

$$F_r(\mu) = \left(\frac{\mu}{r_0}\right)^2 \frac{1}{h^3\pi}\begin{cases} \frac{1}{3}\left(\frac{\mu}{r_0}\right)^{-3} - \frac{3}{10h^2}\left(\frac{\mu}{r_0}\right)^{-5} + \frac{1}{8h^3}\left(\frac{\mu}{r_0}\right)^{-6}, & \mu \geq \frac{r_0}{h}; \\ \frac{1}{4}\left(\frac{8}{3}\left(\frac{\mu}{r_0}\right)^{-3} - \frac{3}{h}\left(\frac{\mu}{r_0}\right)^{-4} + \frac{6}{5h^2}\left(\frac{\mu}{r_0}\right)^{-5} - \frac{1}{6h^3}\left(\frac{\mu}{r_0}\right)^{-6} - \frac{h^3}{15}\right), & \frac{r_0}{2h} \leq \mu \leq \frac{r_0}{h}; \\ \frac{h^3}{4}, & \mu \leq \frac{r_0}{2h}. \end{cases} \tag{4.51}$$

4.3.3 Reducing the surface integral to a contour integral

We have now reduced the problem to integrating a function over the area of a polygon, which is similar to the 2D kernel integration. In order to represent the surface integral I_S as a contour integral we will once again use Green's Theorem, given by:

$$\int_{\mathcal{A}} \nabla \cdot \mathbf{H} \mathrm{d}A = \int_{\partial \mathcal{A}} \mathbf{H} \cdot \hat{\mathbf{m}} \mathrm{d}l, \tag{4.52}$$

where $\hat{\mathbf{m}}$ is the unit vector normal to the contour $\partial \mathcal{A}$ of the area of integration $\mathcal{A}$.

In order to apply the theorem, we need to construct function $\mathbf{H}$, such that $\nabla \cdot \mathbf{H} \mathrm{d}A = \mathbf{F} \cdot \hat{\mathbf{n}} \mathrm{d}S$. While $\mathbf{F}$ is defined as a three-dimensional vector function, $\mathbf{H}$ should be in two dimensions, and the spherically-symmetric nature of W suggests that we should define $\mathbf{H}$ in terms of polar coordinates:

$$\mathbf{H} = H_R \hat{\mathbf{R}} + H_\phi \hat{\boldsymbol{\phi}}; \tag{4.53}$$

$$\nabla \cdot \mathbf{H} = \frac{1}{R} \frac{\partial (R H_R)}{\partial R} + \frac{1}{R} \frac{\partial H_\phi}{\partial \phi}. \tag{4.54}$$

Let us set $H_\phi = 0$. The area element can be written as $\mathrm{d}A = R \mathrm{d}R \mathrm{d}\phi$, where ϕ is ensured to be the same as the three-dimensional coordinate used for $\mathbf{F}$ by aligning the coordinate systems' axes appropriately. This gives us the following expression:

$$\nabla \cdot \mathbf{H} \mathrm{d}A = \frac{\partial (R H_R)}{\partial R} \mathrm{d}R \mathrm{d}\phi. \tag{4.55}$$

From geometrical considerations we can show that $R = r \sin\theta = r_0 \tan\theta$, which leads to $\mathrm{d}R = r_0 \sec^2\theta \mathrm{d}\theta$. This allows us to rewrite the following:

$$\begin{aligned}
\mathbf{F} \cdot \hat{\mathbf{n}} \mathrm{d}S &= F_r r^2 \sin\theta \mathrm{d}\theta \mathrm{d}\phi && (4.56)\\
&= F_r \frac{R^2}{\sin^2\theta} \sin\theta \frac{\mathrm{d}R}{r_0 \sec^2\theta} \mathrm{d}\phi && (4.57)\\
&= F_r \frac{R^2}{r_0 \tan^2\theta} \sin\theta \mathrm{d}R \mathrm{d}\phi && (4.58)\\
&= F_r r_0 \sin\theta \mathrm{d}R \mathrm{d}\phi. && (4.59)
\end{aligned}$$

By combining equations 4.55 and 4.59, we obtain the integral:

$$H_R = \frac{1}{R}\int F_r r_0 \sin\theta \mathrm{d}R. \tag{4.60}$$

We will now write the above expression in terms of μ, so that we can integrate it more easily. We can notice that $\sin\theta\mathrm{d}R = r_0 \sec^2\theta \sin\theta\mathrm{d}\theta = -r_0\mu^{-2}\mathrm{d}\mu$, and with this modification the solution for H_R becomes:

$$H_R R = \frac{r_0^3}{h^3\pi}\begin{cases} \frac{1}{6}\mu^{-2} - \frac{3}{40}(\frac{r_0}{h})^2\mu^{-4} + \frac{1}{40}(\frac{r_0}{h})^3\mu^{-5} + \frac{B_1}{r_0^3}, & \mu \geq \frac{r_0}{h}; \\ \frac{1}{4}(\frac{4}{3}\mu^{-2} - (\frac{r_0}{h})\mu^{-3} + \frac{3}{10}(\frac{r_0}{h})^2\mu^{-4} - \frac{1}{30}(\frac{r_0}{h})^3\mu^{-5} + \frac{1}{15}(\frac{r_0}{h})^{-3}\mu) + \frac{B_2}{r_0^3}, & \frac{r_0}{2h} \leq \mu \leq \frac{r_0}{h}; \\ -\frac{1}{4}(\frac{r_0}{h})^{-3}\mu + \frac{B_3}{r_0^3}, & \mu \leq \frac{r_0}{2h}. \end{cases} \tag{4.61}$$

In the above expression B_1, B_2 and B_3 are the constants of integration, which can be functions of r_0 and h.

4.3.4 Deriving expressions for B_1, B_2 and B_3

Consider integrating $\mathbf{F}$ over the area of a circle, extending from $\mu = 1$ to $\mu = \mu_0$. Depending on the value of r_0 we would need to use different parts of the piecewise form of $\mathbf{F}$. This will result in the constants of integration B_1, B_2 and B_3 having different form depending on r_0, so that $\mathbf{H}$ gives answers consistent with those for $\mathbf{F}$.

In order to find expressions for them, let us start by considering $r_0 \geq 2h$. This means that $\frac{r_0}{2h} \geq 1 \geq \mu$, and we need to use only the third expression for $\mathbf{F}$, which gives us the following integral:

$$I_S = 2\pi\int_{\mu_0}^{1} F_r(\mu)\left(\frac{\mu}{r_0}\right)^{-2}\mathrm{d}\mu = \frac{1}{2}(1-\mu_0). \tag{4.62}$$

If we were to apply Green's theorem and use $\mathbf{H}$, then the following should give us the same answer for all values of μ_0:

$$I_S = \int \mathbf{H} \cdot \hat{\mathbf{m}} \mathrm{d}l = \int_0^{2\pi} H_R R \mathrm{d}\phi = \frac{2B_3}{h^3} - \frac{1}{2}\mu_0. \tag{4.63}$$

By comparing the coefficients of each term of the above polynomials, we get that $B_3 = \frac{h^3}{4}$.

Similarly, we then consider the case of $h \leq r_0 \leq 2h$. Here, we have that $\frac{r_0}{h} \geq 1 \geq \mu$, however, depending on the final integration value of μ_0, we would either use the second polynomial of $\mathbf{F}$ or a sum of the second and the third one. By calculating the integral in two different ways, as shown previously, we can then obtain expressions for the constants. If $\mu_0 \geq \frac{r_0}{2h}$, then we get an expression for B_2, and if $\mu_0 \leq \frac{r_0}{2h}$, we can express B_3.

In the third case, when $r_0 \leq h$, we have three possibilities ($\mu_0 \geq \frac{r_0}{h}$; $\frac{r_0}{2h} \leq \mu_0 \leq \frac{r_0}{h}$; $\mu_0 \leq \frac{r_0}{2h}$), which give rise to expressions for B_1, B_2 and B_3 respectively.

The final polynomial forms of B_1, B_2 and B_3 are as follows:

$$B_1 = \frac{r_0^3}{4}\left(-\frac{2}{3} + \frac{3}{10}\left(\frac{r_0}{h}\right)^2 - \frac{1}{10}\left(\frac{r_0}{h}\right)^3\right); \tag{4.64}$$

$$B_2 = \frac{r_0^3}{4}\begin{cases} -\frac{2}{3} + \frac{3}{10}\left(\frac{r_0}{h}\right)^2 - \frac{1}{10}\left(\frac{r_0}{h}\right)^3 - \frac{1}{5}\left(\frac{r_0}{h}\right)^{-2}, & r_0 \leq h; \\ \\ -\frac{4}{3} + \left(\frac{r_0}{h}\right) - \frac{3}{10}\left(\frac{r_0}{h}\right)^2 + \frac{1}{30}\left(\frac{r_0}{h}\right)^3 - \frac{1}{15}\left(\frac{r_0}{h}\right)^{-3}, & h \leq r_0 \leq 2h; \end{cases} \tag{4.65}$$

$$B_3 = \frac{r_0^3}{4}\begin{cases} -\frac{2}{3} + \frac{3}{10}\left(\frac{r_0}{h}\right)^2 - \frac{1}{10}\left(\frac{r_0}{h}\right)^3 + \frac{7}{5}\left(\frac{r_0}{h}\right)^{-2}, & r_0 \leq h; \\ \\ -\frac{4}{3} + \left(\frac{r_0}{h}\right) - \frac{3}{10}\left(\frac{r_0}{h}\right)^2 + \frac{1}{30}\left(\frac{r_0}{h}\right)^3 - \frac{1}{15}\left(\frac{r_0}{h}\right)^{-3} + \frac{8}{5}\left(\frac{r_0}{h}\right)^{-2}, & h \leq r_0 \leq 2h; \\ \\ \left(\frac{r_0}{h}\right)^{-3}, & r_0 \geq 2h. \end{cases} \tag{4.66}$$

4.3.5 Calculating the contour integral on a line

We have now reduced I_S to a contour integral, which consists of a sum of line integrals (i.e. along the edges of the polygonal wall). Similarly to before, we will only consider the integral of $\mathbf{H}$ over a single line segment, $\mathcal{L}$. The integral that we will focus on is given by:

$$I_L = \int_{\mathcal{L}} \mathbf{H} \cdot \hat{\mathbf{m}} \mathrm{d}l. \tag{4.67}$$

Without loss of generality, we can select the orientation of the coordinate system such that the $\phi = 0$ line is perpendicular to the line segment that we are interested in, just as we did in Section 4.2.2. Let the perpendicular distance to the line from the centre of the 2D polar coordinate system be denoted by R_0.

We then have the following expressions:

$$\hat{\mathbf{m}} = \pm \hat{\mathbf{x}} = \pm(\cos\phi \hat{\mathbf{R}} - \sin\phi \hat{\boldsymbol{\phi}}); \tag{4.68}$$

$$\mathbf{H} \cdot \hat{\mathbf{m}} = \pm H_R \cos\phi. \tag{4.69}$$

The above sign has been discussed in detail in Section 4.2.4, and here again we will assume that it is positive for simplicity of notation. In order to express $\mathrm{d}l$, we will use its vector form given by:

$$\mathrm{d}\mathbf{l} = \mathrm{dR}\hat{\mathbf{R}} + R\mathrm{d}\boldsymbol{\phi}\hat{\boldsymbol{\phi}}. \tag{4.70}$$

Alternatively, we also have that:

$$\mathrm{d}\mathbf{l} = \mathrm{d}l\hat{\mathbf{y}} = \mathrm{d}l \sin\phi \hat{\mathbf{R}} + \mathrm{d}l \cos\phi \hat{\boldsymbol{\phi}}. \tag{4.71}$$

By comparing the $\hat{\boldsymbol{\phi}}$ terms we can write that:

$$\mathrm{d}l = \frac{R\mathrm{d}\phi}{\cos\phi}. \tag{4.72}$$

And hence, for a linear segment, we have that:

$$\mathbf{H}\cdot\hat{\mathbf{m}}\mathrm{d}l = H_R R\mathrm{d}\phi. \tag{4.73}$$

4.3.6 Calculating the line integral analytically

In order to proceed, we require a kernel function which is integrable, in order to provide an analytical (or tabulated) form of this expression. Previously, we have expressed $H_R R$ as a function of μ and now we want to integrate it with respect to ϕ. In order to complete the integration we need to express μ as a function of ϕ or vice versa.

From geometrical considerations we have the following:

$$R = \frac{R_0}{\cos\phi} = r\sin\theta. \tag{4.74}$$

And hence

$$r = \frac{R_0}{\sin\theta\cos\phi}. \tag{4.75}$$

We also have that:

$$\mu = \frac{r_0}{r} = \frac{r_0\sin\theta\cos\phi}{R_0}. \tag{4.76}$$

By squaring both sides, substituting $\sin^2\theta$ for $1-\mu^2$, and rearranging, we obtain the relationship:

$$\mu = \frac{\frac{r_0}{R_0}\cos\phi}{\sqrt{1+\frac{r_0^2}{R_0^2}\cos^2\phi}}. \tag{4.77}$$

Since $H_R R$ is a polynomial consisting of different powers of μ (see equation 4.61), we need to integrate the following terms and insert them into the polynomial:

$$I_n = \int \mu^n \mathrm{d}\phi = \int \left(\frac{\frac{r_0}{R_0}\cos\phi}{\sqrt{1 + \frac{r_0^2}{R_0^2}\cos^2\phi}} \right)^n \mathrm{d}\phi, \tag{4.78}$$

where $n \in \mathbb{Z}$.

We can easily notice that I_0 is trivial and can be expressed as:

$$I_0 = \int \mathrm{d}\phi = \phi + C. \tag{4.79}$$

For the remaining even powers ($n = -2k$, $k \in \mathbb{N}$) of μ we can simplify as follows:

$$I_{-2k} = \int \left(1 + \frac{1}{\frac{r_0^2}{R_0^2}\cos^2\phi} \right)^k \mathrm{d}\phi. \tag{4.80}$$

Hence,

$$\begin{aligned} I_{-2} &= \phi + \int \frac{\mathrm{d}\phi}{\frac{r_0^2}{R_0^2}\cos^2\phi} && (4.81) \\ &= \phi + \left(\frac{R_0}{r_0}\right)^2 \tan\phi + C; && (4.82) \end{aligned}$$

$$\begin{aligned} I_{-4} &= \int \left(1 + \frac{2}{\frac{r_0^2}{R_0^2}\cos^2\phi} + \frac{1}{\frac{r_0^4}{R_0^4}\cos^4\phi} \right) \mathrm{d}\phi && (4.83) \\ &= \phi + 2\left(\frac{R_0}{r_0}\right)^2 \tan\phi + \frac{1}{3}\left(\frac{R_0}{r_0}\right)^4 \tan\phi(\sec^2\phi + 2) + C. && (4.84) \end{aligned}$$

For the odd powers ($n = 1$; $n = -3$; $n = -5$) we will express ϕ in terms of μ, as it follows from equation 4.78:

$$\mathrm{d}\phi = -\frac{R_0}{r_0} \frac{\mathrm{d}\mu}{(1-\mu^2)\sqrt{1 - \left(1 + \frac{R_0^2}{r_0^2}\right)\mu^2}}. \tag{4.85}$$

Starting with integrating the expression for $n = 1$, let $\alpha = \frac{R_0}{r_0}$, and then:

$$I_1 = \int \frac{-\alpha\mu \mathrm{d}\mu}{(1-\mu^2)\sqrt{1-(1+\alpha^2)\mu^2}} \tag{4.86}$$

Let $u = \sqrt{1-(1+\alpha^2)\mu^2}$. Then $\mathrm{d}u = -\frac{(1+\alpha^2)\mu\mathrm{d}\mu}{\sqrt{1-(1+\alpha^2)\mu^2}}$, and $1-\mu^2 = \frac{\alpha^2+u^2}{1+\alpha^2}$. This changes the expression for I_1 to:

$$I_1 = \int \frac{\alpha \mathrm{d}u}{\alpha^2+u^2} = \tan^{-1}\left(\frac{u}{\alpha}\right) + C. \tag{4.87}$$

Using the same substitution, the expressions for I_{-3} and I_{-5} can be written as follows (for more details, see Section 4.6):

$$\begin{aligned} I_{-3} = {} & \frac{\alpha(1+\alpha^2)}{4}\left(\frac{2u}{1-u^2} + \log(1+u) - \log(1-u)\right) \\ & + \frac{\alpha}{2}(\log(1+u) - \log(1-u)) + \tan^{-1}\left(\frac{u}{\alpha}\right) + C; \end{aligned} \tag{4.88}$$

$$\begin{aligned} I_{-5} = {} & \frac{\alpha(1+\alpha^2)^2}{16}\left(\frac{10u-6u^3}{(1-u^2)^2} + 3(\log(1+u) - \log(1-u))\right) \\ & + \frac{\alpha(1+\alpha^2)}{4}\left(\frac{2u}{1-u^2} + \log(1+u) - \log(1-u)\right) \\ & + \frac{\alpha}{2}(\log(1+u) - \log(1-u)) + \tan^{-1}\left(\frac{u}{\alpha}\right) + C. \end{aligned} \tag{4.89}$$

The final solution is hence given by:

$$I_P = \frac{r_0^3}{h^3\pi}\begin{cases} \frac{1}{6}I_{-2} - \frac{3}{40}\left(\frac{r_0}{h}\right)^2 I_{-4} + \frac{1}{40}\left(\frac{r_0}{h}\right)^3 I_{-5} + \frac{B_1}{r_0^3}I_0, & \frac{r_0}{h} \leq \mu; \\ \\ \frac{1}{4}(\frac{4}{3}I_{-2} - (\frac{r_0}{h})I_{-3} + \frac{3}{10}(\frac{r_0}{h})^2 I_{-4} - \frac{1}{30}(\frac{r_0}{h})^3 I_{-5} + \frac{1}{15}(\frac{r_0}{h})^{-3}I_1) + \frac{B_2}{r_0^3}I_0, & \frac{r_0}{2h} \leq \mu \leq \frac{r_0}{h}; \\ \\ -\frac{1}{4}(\frac{r_0}{h})^{-3}I_1 + \frac{B_3}{r_0^3}I_0, & \mu \leq \frac{r_0}{2h}. \end{cases} \tag{4.90}$$

In the applications of this method boundary conditions are applied to ensure continuity between the different regions of the function.

The method can be applied to any piecewise polynomial kernel and I will discuss this in more detail in Section 4.6 and Appendix A.

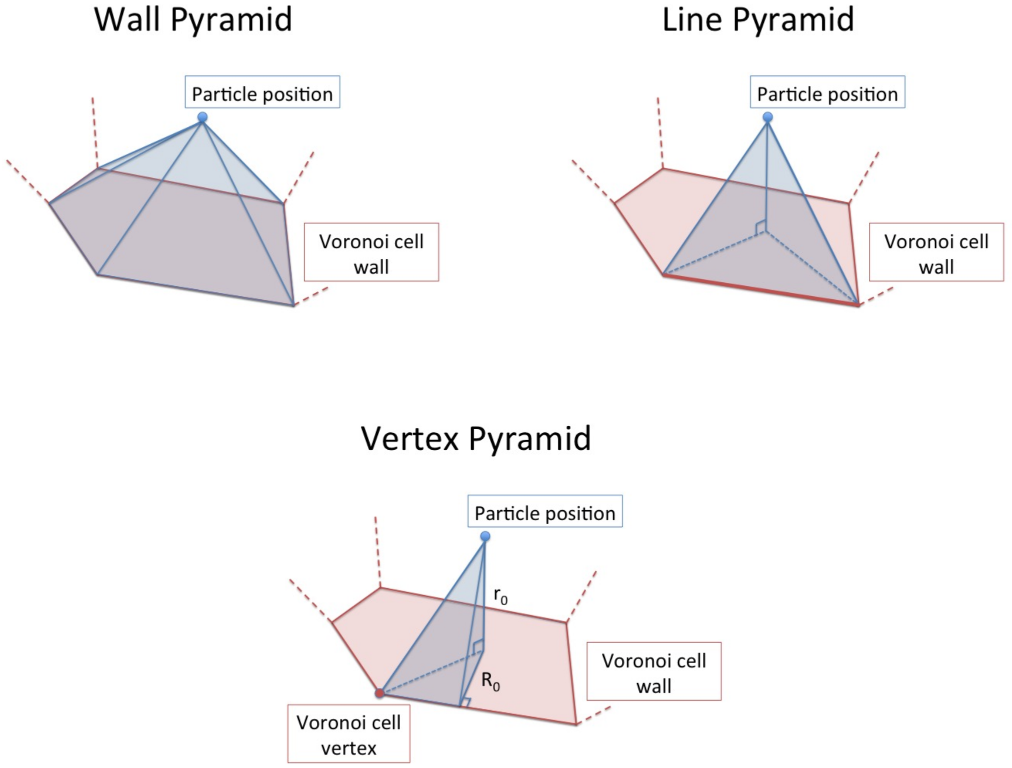

Figure 4.5: Geometric representation of I_S (*top left*), I_L (*top right*) and I_P (*bottom*). I_S, I_L and I_P equal the integral of the kernel function over the volume of each respective pyramid.

4.3.7 Geometric interpretation of the solution

Similarly to the 2D case, the process of obtaining the 3D integral of the kernel function has a clear geometric analogy. By applying the Divergence Theorem, we effectively split the cell volume into wall pyramids (see Figure 4.5). The integral of the kernel function over the volume of a wall pyramid is given by I_S. Furthermore, with the use of Green's theorem, we split each wall pyramid into line pyramids, and the integral of the kernel inside each of them equals I_L. Finally, each line pyramid can be broken down into two vertex pyramids, with the integral of the kernel function over the volume of a vertex pyramid being equal to I_P.

In Section 4.2.4 we discussed the scenario when the particle position was outside the cell, and we saw that some of the integrals would be included with a negative sign. The same idea can be generalised to 3D. When the particle position is outside of the polyhedral cell, some of the wall integrals will have negative signs arising from equation 4.48. This ensures that when all I_S expressions are added up they will give the integral of the kernel function over the volume of the cell.

4.4 Testing

4.4.1 Accuracy

After completing the analytic derivation of the kernel integral, I have created computer implementations of the density mapping in 2D and 3D. As a first step in the implementations, I have considered a single SPH particle mapped onto a Voronoi grid. To calculate the density of a cell in 3D, I have looped over each side of each wall, and computed r_0, R_0 and ϕ for each of the two vertices of the side. This allowed me to calculate the I_P expressions, and add them all up. The sum was then multiplied by the particle mass and divided by the cell volume in order to produce the average cell density. An analogous procedure was developed in 2D, and both of these implementations were applied to test cases (see Figure 4.6 and 4.7).

The results of the test cases in Figure 4.6 and 4.7 match the expected cell density distribution by visual inspection. Additionally, in both cases the total mass of the cells equals the mass of the SPH particle, as expected. To test the analytic integrals more

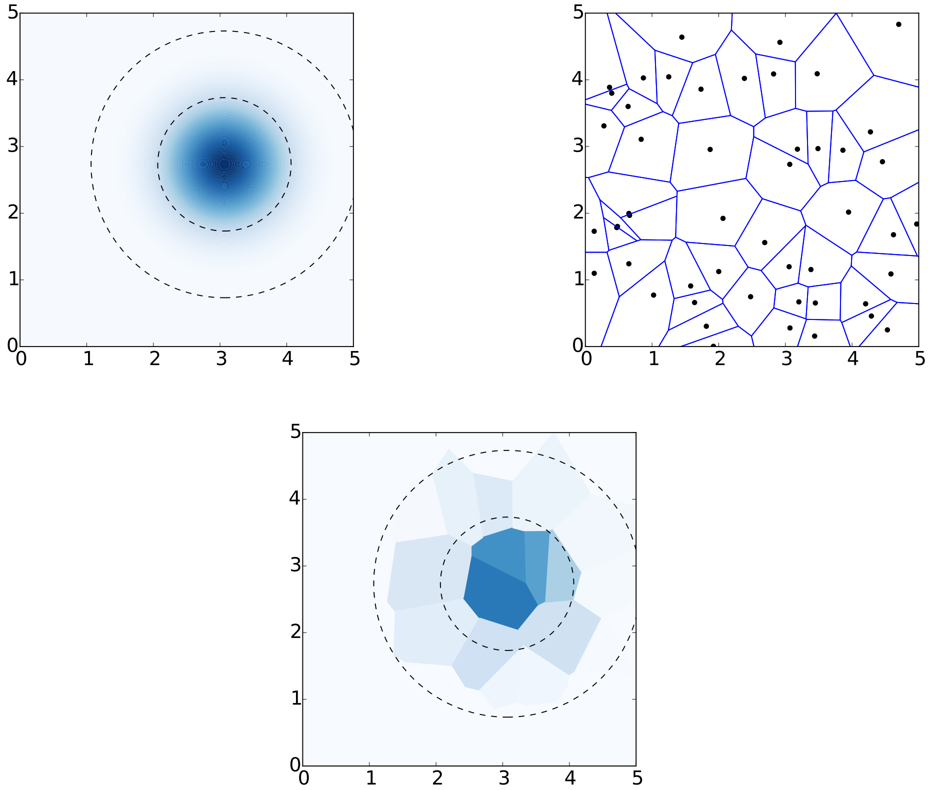

Figure 4.6: *Top left*: 2D cubic spline kernel function with $h = 1$. Dark blue corresponds to higher density, and the dashed contours are at $r = h$ and $r = 2h$. The value of the kernel function is zero outside of the larger circle. *Top right*: 2D Voronoi grid with 50 randomly selected generating sites. *Bottom*: the average density of each Voronoi cell, calculated with the analytic density mapping is plotted in colour. The same colour scheme is used as in the top left plot, and we can see that we preserve the SPH density structure. *The figure is adapted from Petkova et al. (2018).*

rigorously, I have implemented a numerical integration algorithm in 3D and applied it to the same test case as the one in Figure 4.7. The numerical integration followed Simpson's rule, and it showed good agreement with the analytic solution (see Figure 4.8). With this I have demonstrated the validity of the proposed density mapping method.

4.4.2 Timing

As a next step of the testing, I have applied the analytic density mapping to data taken from an SPH simulation of a clumpy cloud in 3D (Forgan and Bonnell (in prep.)). The simulation was produced with Phantom (Price et al. (2017)), and contains 400728 particles evolved for about 1700 years. In its initial state, the modelled cloud contained high density cores embedded into a uniform low density medium, and was later perturbed by a shock. In order to test my density mapping method, I have used the last snapshot of the simulation and have constructed a single grid cell around the position of each particle.

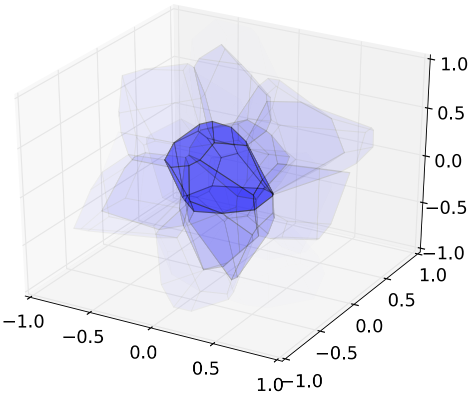

Figure 4.7: 3D cubic spline kernel function with $h = 0.5$ is mapped onto a 3D Voronoi grid with 50 randomly sampled generating sites. The average density of each cell, computed with the analytic density mapping, is used as cell transparency, with darker regions corresponding to higher column density along the line of sight. The result is similar to Figure 4.6, but performed for a 3D test. *The figure is adapted from Petkova et al. (2018).*

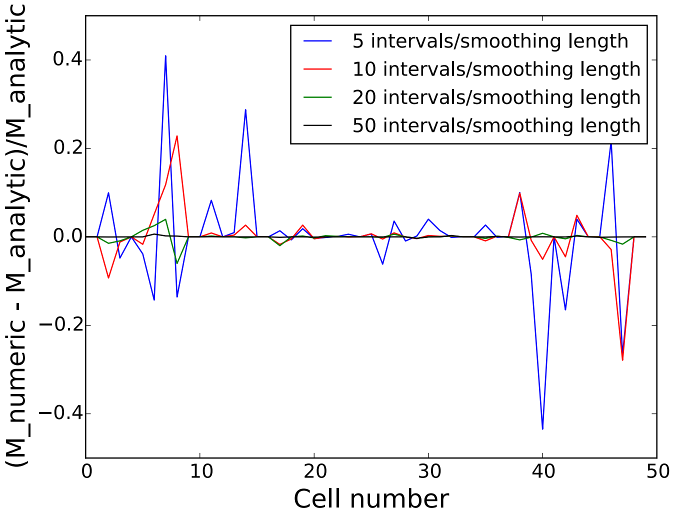

Figure 4.8: Fractional difference between the numerically calculated and the analytically calculated cell masses for each cell of the setup in Figure 4.7. The differently coloured lines correspond to different number of intervals per smoothing length that were used for the numerical integration. By increasing the number of intervals, the numerical masses converge to the analytically calculated ones, demonstrating the validity of the method. *The figure is adapted from Petkova et al. (2018).*

In order to apply the analytic density mapping to a dataset of multiple SPH particles and grid cells, I have made changes to the previously mentioned computer implementation in 3D. Instead of looping over all cells for each particle, which would be slow and inefficient, I have considered only the cells which would be reached by the particle's kernel function. This was done by creating a simple friends-of-friends type of algorithm, realised with a queue structure. The queue structure contains cell indices and it initially starts with only one element, which is the index of the cell generated around the particle's location. As we loop through the queue, we check the location of each wall of the current grid cell. If a wall is partially or fully within a radius of two smoothing lengths from the particle, we add the cell on the other side of the wall to the back of the queue, as long as it is not already in it. This ensures that all cells which receive non-zero mass contributions from a particle are covered.

I have used the SPH dataset in order to perform timing tests of the density mapping algorithm. In order to do so, I have selected subsets of the the dataset and timed the performance of the mapping algorithm for each one of them. Since the number of cells that receive non-zero mass contribution from a given particle is between about 50 and 100, for the larger subsets we can treat that number as constant and expect the computing time to depend roughly linearly on the number of particles in the subset.

If we were to compare full SPH datasets of different total particle numbers, this linear dependence might not hold. Zhu et al. (2015) have demonstrated that for the purposes of numerical convergence, an SPH dataset should have the number of neighbours of each particle scaling as the square root of the total particle number. If this relationship is followed, then the computing time would be proportional to the total number of particles to the power of 1.5 when different SPH datasets are compared.

Figure 4.9 shows the expected linear dependence of the analytic computing time as a function of particle numbers. I have repeated the calculation using the numerical integration with 10 intervals per smoothing length, and once again found a linear dependence. We can clearly see, however, that the analytic solution is a factor of about 200 faster than the numerical one, which makes the former much more favourable to use than the latter.

4.4.3 Comparison to other density mapping methods

The previous section demonstrates the validity of the analytic density mapping method, and now I will compare its results to the approximate density calculations introduced at the beginning of the chapter. As a reminder, what we have called method 1 is when a single grid cell is created around each particle's position, and the density is given by dividing the particle's mass by the cell volume. Method 2, on the other hand, computes

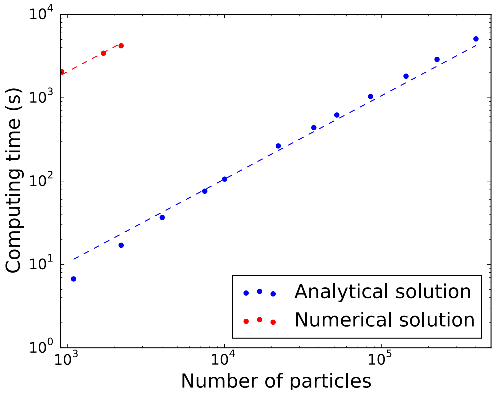

Figure 4.9: Computing time as a function of number of SPH particles for the analytically calculated density structure (blue) and the numerically calculated density structure with 10 intervals per smoothing length (red). In both cases, there is a linear dependence between computing time and the number of particles, as expected. The numerical solution is a factor of 200 times slower than the analytic one. *The figure is adapted from Petkova et al. (2018).*

the SPH density at the centroid of each cell and uses it as the average cell density. I have already stated that both method 1 and method 2 are likely to lead to errors, and this is also evident in Figure 4.10.

Figure 4.10 contains a comparison of method 1 and method 2 with the analytic density mapping. The methods were applied to four different datasets — a clumpy cloud, a uniform density box, a disk galaxy and a cloud affected by a supernova explosion. The clumpy cloud is the same dataset that was used in the previous section, and the other models were produced with SPHNG (Bate et al. (1995)). The box consists of 122333 uniformly sampled particles, with some noise present in the particle positions and densities. For this test I have used the initial setup of the model without evolving it in time. Similarly, I have also used the initial setup of the disk galaxy model (Ramón-Fox and Bonnell (2016)). The galaxy consists of 500000 gas particles, which follow a smooth density power-low without any features (McMillan and Dehnen (2007)). Finally, the post-supernova cloud represents a shock wall where many complex structures are present and it contains 208155 particles (Lucas et al. (in prep.)).

We can see that for all datasets in Figure 4.10 both method 1 and method 2 can produce significant errors from the analytic density values. Method 2 has broader range of density deviations from the analytic solution than method 1, however, the bulk of its particles are concentrated in a narrower region around 0%. In addition to its primary peak, method 2 has a secondary peak at -100% for three of the four datasets, which corresponds

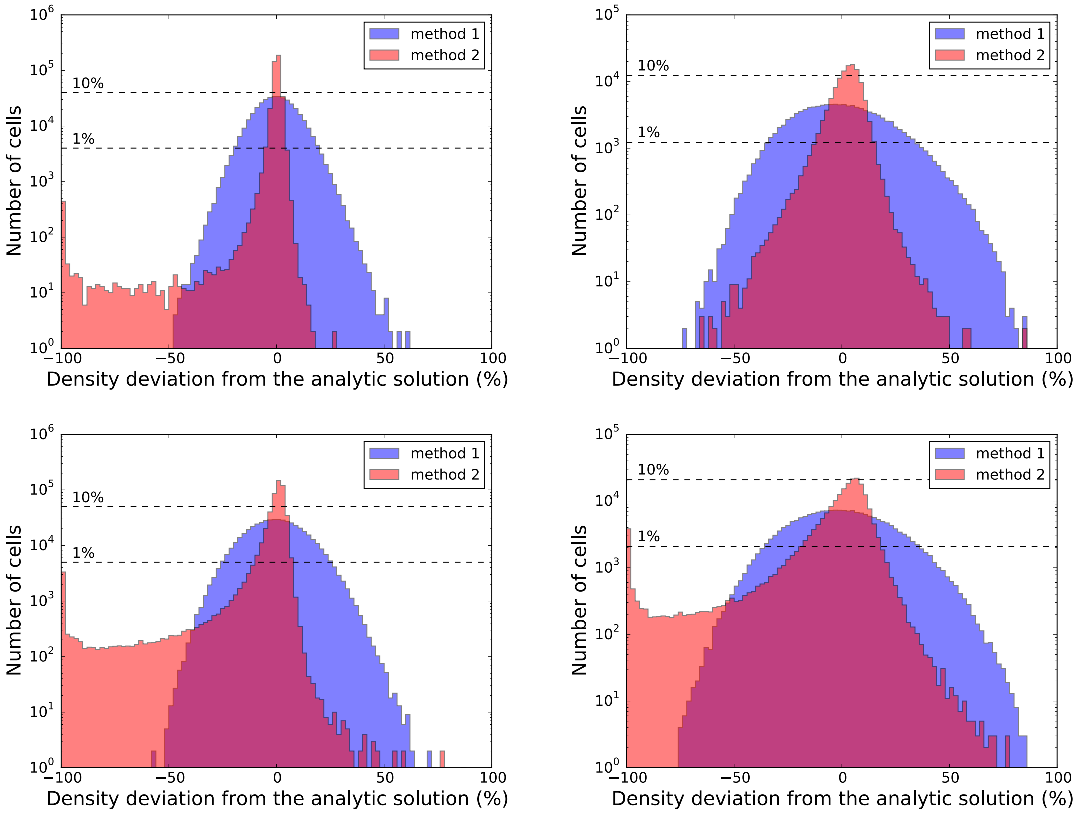

Figure 4.10: Histograms showing the accuracy of two commonly used methods for density calculation: particle mass divided by cell volume (method 1) and SPH density at the cell centroid point (method 2). Method 1 and 2 are compared to the analytically computed density for the case of a clumpy cloud (*top left*), uniform density box (*top right*), disk galaxy (*bottom left*) and supernova shock wall (*bottom right*). The dashed horizontal lines indicate the level of 1% and 10% respectively of the total number of cells. This is to note the low numbers of cells in the wings of the distributions. Both method 1 and 2 show significant deviations from the analytic solution, which can cause inaccuracies when MCRT is performed on SPH data. *The figure is adapted from Petkova et al. (2018).*

to cells with zero density in method 2. These cells are located at the outskirts of the data samples, where there is cell elongation due to the cuboid boundaries. This is why the dataset without a secondary peak is the one of the uniform density cube.

As previously discussed, method 2 does not ensure mass conservation, however the mass discrepancy between the SPH particles and the grid cells is small. For each of the datasets the total mass of all cells with densities obtained by method 2 differs from the analytic total mass by less than 5%. This leads us to the conclusion that method 2 appears preferable to method 1, especially for datasets without many elongated cells at the boundaries.

It is unclear how strongly a Monte Carlo Radiative Transfer calculation would be affected by the choice of density mapping method, just by looking at the density errors. Furthermore, even if there is a significant variation in the output of the radiative transfer calculation depending on the density mapping, it is not trivial to foresee how strongly a radiation hydrodynamics simulation would be affected. For these reasons further testing and comparison between the three types of density mapping will be presented in Chapter 5.

4.5 Tabulation

4.5.1 Creating a table of pre-computed values of I_P

In order to further boost the computing performance of the density mapping algorithm, I will pre-compute and tabulate I_P in 3D (see equation 4.90) for different values of the parameters it contains. In 3D these parameters are $r_0 \in [0, \infty)$, $R_0 \in [0, \infty)$ and $\phi \in [0, \pi/2)$. The infinite ranges for r_0 and R_0 present a challenge for the tabulation, which needs to be addressed.

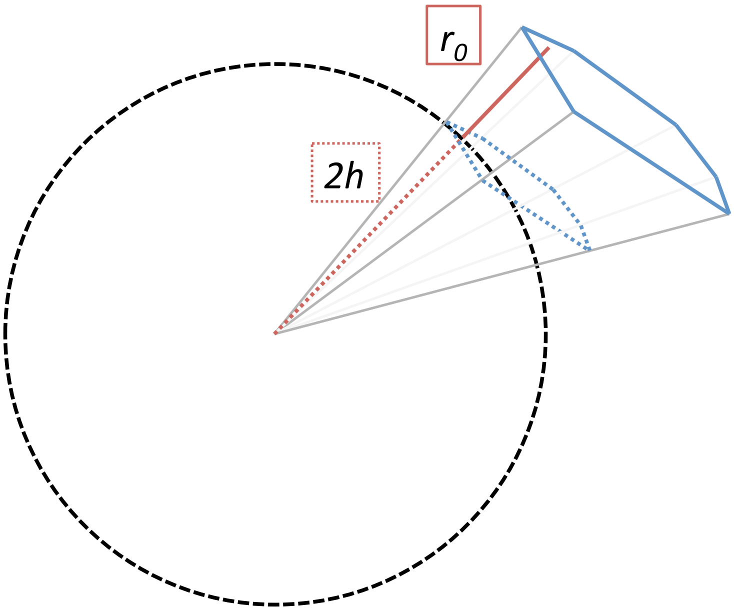

Figure 4.11: Demonstration that a cell wall with $r_0 > 2h$ contains the same fraction of an SPH particle (and therefore has the same I_P values) as the projected wall at $r_0 = 2h$, provided all of the angles of the wall pyramids are preserved.

First we can notice that for $r_0 \geq 2h$ we have a cell wall which does not intersect the non-zero region of the of the particle's kernel. This means that for $r_0 > 2h$ the integral of the volume of the wall pyramid (Figure 4.5) is the same as that for a wall at $r_0 = 2h$ with the same angular dimensions (see Figure 4.11). This allows us to only consider the range of $[0, 2h]$ as possible values of r_0.

Furthermore, instead of using R_0, we can consider the parameter $\mu_0 = \cos\theta_0 = \frac{r_0}{\sqrt{r_0^2+R_0^2}}$, which is defined in the range of $[0, 1]$. This way by selecting a value for r_0 and μ_0 we uniquely set a value for R_0.

Finally, to tabulate the values of I_P, I chose 50 equally spaced values of r_0 between 0 and $0.1h$, and further 200 equally spaced values between $0.1h$ and $2h$. Additionally, for both μ_0 and $\cos\phi$ I selected 150 equally spaced values between 0 and 0.98, and further 150 equally spaced values between 0.98 and 1.0. For each combination of these three parameters, I computed I_P.

4.5.2 Obtaining I_P from the pre-computed table

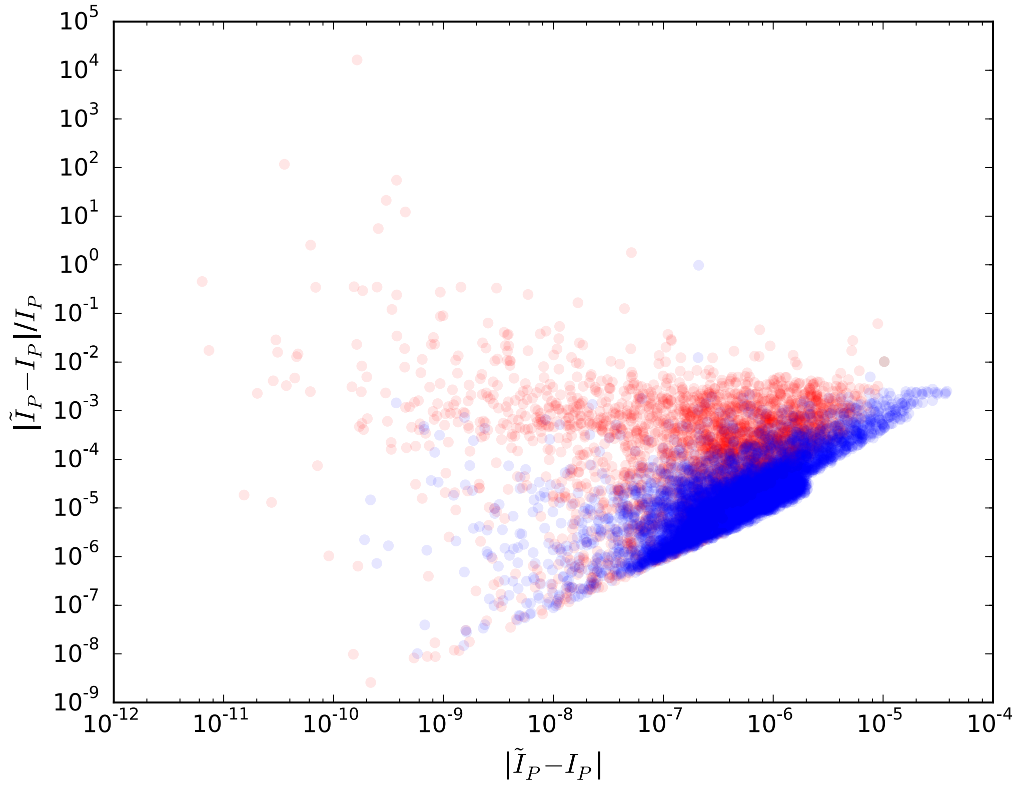

Figure 4.12: Absolute error vs relative error of the approximate value of $\tilde{I}_P$ obtained by pre-computing and tabulating I_P. Potted are values of $|\tilde{I}_P - I_P|$ and $|\tilde{I}_P - I_P|/I_P$ for 10000 randomly generated parameter triplets of r_0, R_0 and ϕ. The red data points are those for which $\tilde{I}_P$ is an overestimate of I_P and the blue ones are underestimates.

Having this table of pre-computed values allows us to quickly approximate I_P for any values of r_0, R_0 and ϕ by interpolating. Due to the two-part equal spacing between the tabulated values it is easy and fast to find the location of interest in the table of pre-computed values.

I have used 3D linear interpolation in order to obtain an approximate value of I_P (from now on referred to as $\tilde{I}_P$) and found that this method produces good agreement with the exact I_P (see Figure 4.12). By comparing 10000 different values of $\tilde{I}_P$ to their I_P analogues, we can see that for $I_P > 10^{-5}$ the relative error of the approximate value is under 1%. For $I_P < 10^{-5}$ the relative error becomes higher, however it is worth recalling that I_P represents a fraction of an SPH particle's mass. For all practical purposes a mass fraction under 10^{-5} is negligible, and therefore we should not be too concerned about discrepancies between $\tilde{I}_P$ and I_P in that range, as this will not affect the density mapping significantly.

As an extra accuracy test I have repeated the density mapping for the dataset of a clumpy cloud presented in Section 4.4.3 by using $\tilde{I}_P$. We can see in Figure 4.13 that all of the Voronoi cells with mapped densities have errors that are less than 0.7%, which is indeed very small. Note that the other approximate density mapping methods produced much larger errors for the same dataset (see Figure 4.10).

Additionally, the tabulation and interpolation of I_P has decreased the computing time for the density mapping by about 1.5 times. This is a significant improvement when large datasets are considered and when the mapping is performed multiple times, such as in the case of live radiation-hydrodynamics.

4.6 Other kernel functions

From the analytic derivations presented in section 4.2 and 4.3 we can notice that any polynomial kernel function could be integrated analytically, provided that we can find expressions for all J_m and I_n (see equation 4.28 and 4.78). Therefore in this section I will demonstrate that this is always the case and I will provide solutions to these integrals.

As a reminder, the forms of J_m and I_n, as given by equation 4.28 and 4.78, are as follows:

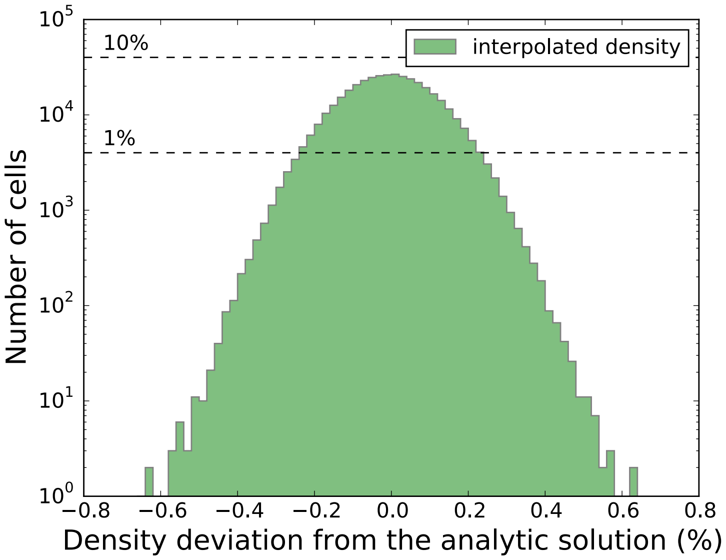

Figure 4.13: A histogram of density error (in %) for a density mapping obtained by using $\tilde{I}_P$ as opposed to I_P on an SPH dataset of a clumpy cloud (see Section 4.4.3). The horizontal dashed lines mark the levels of 1% and 10% of the total number of cells in the dataset. The minor approximation created by tabulating pre-computed values of I_P and then interpolating creates a very small error, especially compared to the other approximate methods presented in Figure 4.10.

$$J_m = \int \frac{\mathrm{d}\phi}{\cos^m \phi}, \tag{4.91}$$

$$I_n = \int \mu^n \mathrm{d}\phi = \int \left(\frac{\frac{r_0}{R_0}\cos\phi}{\sqrt{1+\frac{r_0^2}{R_0^2}\cos^2\phi}} \right)^n \mathrm{d}\phi, \tag{4.92}$$

where $n, m \in \mathbb{Z}$, $m \geq 0$, $n \leq 1$.

The expressions for J_m were already presented in section 4.2, and here I will reiterate that for the even and odd powers of m we have respectively (using Gradshteyn and Ryzhik (2007)):

$$J_{2k} = \int \frac{\mathrm{d}\phi}{\cos^{2k}\phi} = \frac{\sin\phi}{2k-1} \left(\sec^{2k-1}\phi + \sum_{p=1}^{k-1} \frac{2^p (k-1)(k-2)...(k-p)}{(2k-3)(2k-5)...(2k-2p-1)} \sec^{2k-2p-1}\phi \right) + C, \tag{4.93}$$

and

$$J_{2k+1} = \int \frac{\mathrm{d}\phi}{\cos^{2k+1}\phi} = \frac{\sin\phi}{2k}\left(\sec^{2k}\phi + \sum_{p=1}^{k-1}\frac{(2k-1)(2k-3)...(2k-2p+1)}{2^p(k-1)(k-2)...(k-p)}\sec^{2k-2p}\phi\right) + \frac{(2k-1)!!}{2^k k!}\ln\tan\left(\frac{\pi}{4}+\frac{\phi}{2}\right) + C. \quad (4.94)$$

I will now continue by finding similar expressions for I_n. Section 4.3 handles the cases of $n = 0$ and $n = 1$, hence I will focus on the negative values of n.

For the even powers ($n = -2k$, $k \in \mathbb{N}$) of μ we can simplify equation 4.92 as follows:

$$\begin{aligned} I_{-2k} &= \int \left(1 + \frac{1}{\frac{r_0^2}{R_0^2}\cos^2\phi}\right)^k \mathrm{d}\phi && (4.95) \\ &= \sum_{l=0}^{k}\int \binom{k}{l}\frac{\mathrm{d}\phi}{\frac{r_0^{2l}}{R_0^{2l}}\cos^{2l}\phi}, && (4.96) \end{aligned}$$

where $\binom{k}{l}$ are binomial coefficients. The above is simply a sum of integrals that have the same form as J_{2k}. Hence, by using the expression in equation 4.93, we can obtain all of the solutions that we need.

For the odd values of n we will use that:

$$\mathrm{d}\phi = -\frac{R_0}{r_0}\frac{\mathrm{d}\mu}{(1-\mu^2)\sqrt{1-\left(1+\frac{R_0^2}{r_0^2}\right)\mu^2}}. \quad (4.97)$$

Let $\alpha = \frac{R_0}{r_0}$ and $n = -2k-1$ ($k \in \mathbb{N}$). This changes the expression for I_n to:

$$\begin{aligned} I_{-2k-1} &= \int \frac{-\alpha\mathrm{d}\mu}{\mu^{2k+1}(1-\mu^2)\sqrt{1-(1+\alpha^2)\mu^2}} && (4.98) \\ &= \int \frac{-\alpha(1-\mu^2+\mu^2)\mathrm{d}\mu}{\mu^{2k+1}(1-\mu^2)\sqrt{1-(1+\alpha^2)\mu^2}} && (4.99) \\ &= \int \frac{-\alpha\mathrm{d}\mu}{\mu^{2k+1}\sqrt{1-(1+\alpha^2)\mu^2}} + \int \frac{-\alpha\mathrm{d}\mu}{\mu^{2k-1}(1-\mu^2)\sqrt{1-(1+\alpha^2)\mu^2}} && (4.100) \\ &= I'_{-2k-1} + I_{-2k+1}. && (4.101) \end{aligned}$$

The above expression establishes a recurrence relation between I_{-2k-1} and the previous odd power I_{-2k+1} in the series, and we are left to determine $I^{'}_{-2k-1}$ (since the first elements in the series, I_1 and I_{-1}, have already been derived in section 4.3).

Let $u = \sqrt{1-(1+\alpha^2)\mu^2}$. Then $\mathrm{d}u = -\frac{(1+\alpha^2)\mu \mathrm{d}\mu}{\sqrt{1-(1+\alpha^2)\mu^2}}$, and $\mu^2 = \frac{1-u^2}{1+\alpha^2}$. The expression for $I^{'}_{-2k-1}$ can be written as:

$$I^{'}_{-2k-1} = \alpha(1+\alpha^2)^k \int \frac{\mathrm{d}u}{(1-u^2)^{k+1}}. \tag{4.102}$$

From Gradshteyn and Ryzhik (2007) we have that:

$$\int \frac{\mathrm{d}u}{(1-u^2)^{k+1}} = \frac{u}{2k+1} \sum_{l=1}^{k} \frac{(2k+1)(2k-1)...(2k-2l+3)}{2^l k(k-1)...(k-l+1)(1-u^2)^{k-l+1}} + \frac{(2k-1)!!}{2^{k+1}k!}(\log(1+u) - \log(1-u)) + C. \tag{4.103}$$

With this I demonstrate how to construct the solution for the general case of a polynomial kernel function. The specific solutions for some of the commonly used kernel functions are included in Appendix A.

4.7 Broader applicability of the method

After demonstrating that the method presented in this chapter can be applied to any polynomial kernel function, I would like to point out a few other key expansions to what was already shown.

First of all, the analytic density mapping method allows us to have a Voronoi cell distribution which does not need to match the SPH particle positions or numbers. This was already demonstrated in Figure 4.6 and 4.7, where we had a single SPH particle and multiple Voronoi cells. As I have already discussed in Chapter 3 there are circumstances in which we would like to be able to construct the Voronoi grid independently from the particle positions, in order to avoid poor resolution effects (see also Koepferl et al. (2016)).

Additionally, the focus of this chapter has been the mapping of SPH particles onto a Voronoi grid, however while performing the integrations we did not use any of the properties of a Voronoi tessellation. The density mapping can be therefore used with any other grid structure that has polyhedral cells, such as a Cartesian or AMR grid, making the approach a lot more broadly applicable.

Finally, while what I need for my work is the density structure that is described by the SPH particles, all other SPH properties use smoothing kernels in a way that is mathematically equivalent to the density (see Chapter 2). This means that the method presented here can be applied to the mapping of any other SPH quantity onto a range of grid structures. This can be used for accurate imaging of SPH data (see Figure 4.14) and enabling direct comparison between SPH, grid and moving mesh codes.

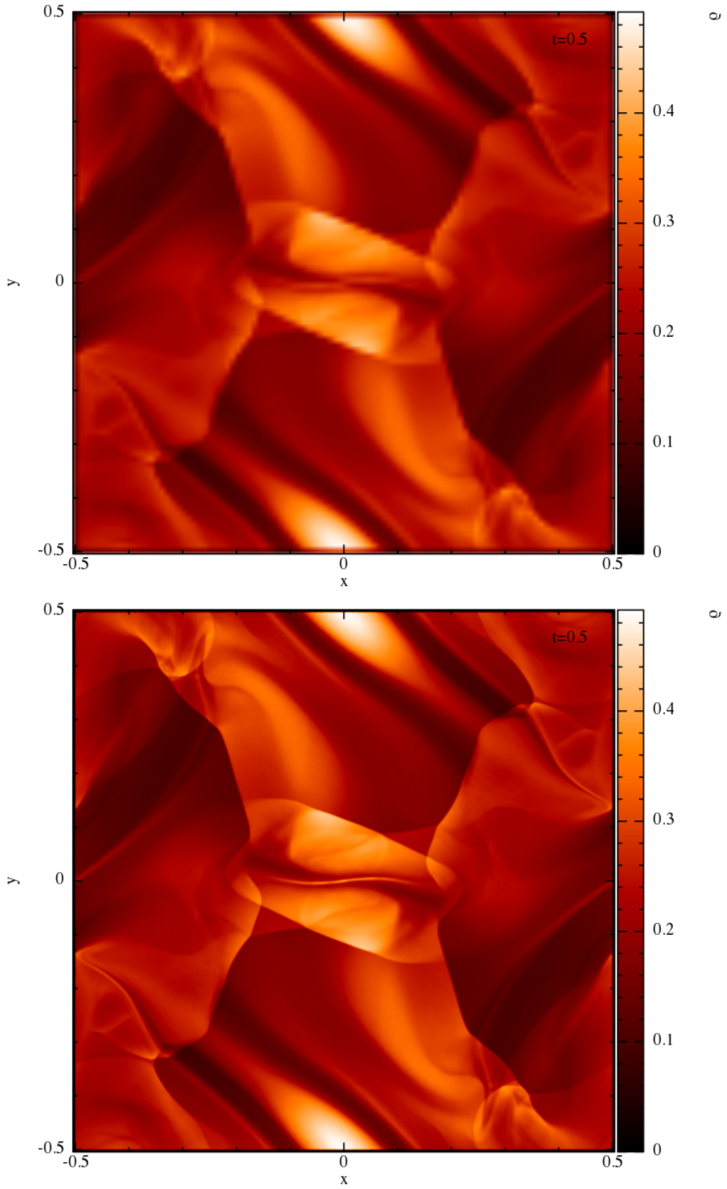

Figure 4.14: Examples of exact SPH imaging performed by SPLASH (Price (2007)), using the mapping method of Petkova et al. (2018). The two images have a different number of pixels but the same total pixel count. *Image source: Price et al. (2017).*

Chapter 5

Radiation Hydrodynamics

In this chapter I will use the work developed in the previous chapters in order to reach the goal of my thesis, namely, to achieve live radiation hydrodynamics using SPH and MCRT. I will present the publicly available codes that I have used for modelling the hydrodynamics (Phantom, Price et al. (2017)) and the radiative transfer (CMacIonize, Vandenbroucke and Wood (2018)), as well as the modifications which I have implemented to allow me to run them together.

Furthermore, I will demonstrate the accurate execution of this live radiation hydrodynamics scheme by applying it to the well studied test problem of D-type expansion of an H II region. And finally, I will apply the scheme to a more realistic star formation setup, in which high mass young stars disperse most of the gas of the cloud they are embedded in via ionising radiation (Dale et al. (2012)).

5.1 Introduction to the codes

5.1.1 Phantom

The SPH code that I used for this work is called Phantom and it was created with a focus on stellar, planetary and galactic astrophysics (Price et al. (2017)). The code is well suited for the study of star formation as it is 3-dimensional and it includes the treatment of self-gravity, sink particles and gas-dust mixtures. Additionally, it can be run as both ideal and non-ideal MHD.

Phantom has been used widely for variety of problems and contains a broad range of functionality, most of which is beyond the scope of this thesis (for a detailed review see Price et al. (2017)). The features which I have used include self-gravity, the creation of sink particles and artificial conductivity. The latter is switched on by default and is used for the correct treatment of shocks and hydrodynamical instabilities.

PHANTOM is written in Fortran 90 and is parallelised using OpenMP and MPI. The code is highly modular, allowing for quick and easy modifications.

5.1.2 CMacIonize

The MCRT code that I used for the live radiation hydrodynamics is called CMACIONIZE (Vandenbroucke and Wood (2018)). The code models photoionisation on a variety of 3D grid structures, including Cartesian, AMR and Voronoi grid, and as such was well suited for this project. CMACIONIZE contains a live radiation hydrodynamics scheme in itself, allowing the user to choose between a fixed grid and a moving mesh grid hydrodynamics. This extra functionality has been demonstrated to work correctly (Vandenbroucke and Wood (2018)), through comparison with the D-type expansion benchmark, which will be introduced later in this chapter.

CMACIONIZE models the photoionisation of hydrogen and helium self-consistently in the energy range between 13.6 and 54.4 eV. The ionisation states of a number of metal ions are modelled approximately, as they contribute to the cooling of the gas. Presently there is no treatment of lower energy photons, dust or radiation pressure (for more information see Vandenbroucke and Wood (2018)).

CMACIONIZE is written in C++ and can be run in parallel using a mixture of OpenMP and MPI. Additionally, the code has a library interface that allows it to be called directly from C, C++ or Fortran programs.

5.2 Radiation hydrodynamics structure and modifications

5.2.1 Code overview

In order to achieve live radiation hydrodynamics I have established connection between the two codes by using PHANTOM's option for performing live analysis and CMACIONIZE's library functionality. A schematic of the codes' communication points is shown in Figure

5.1.

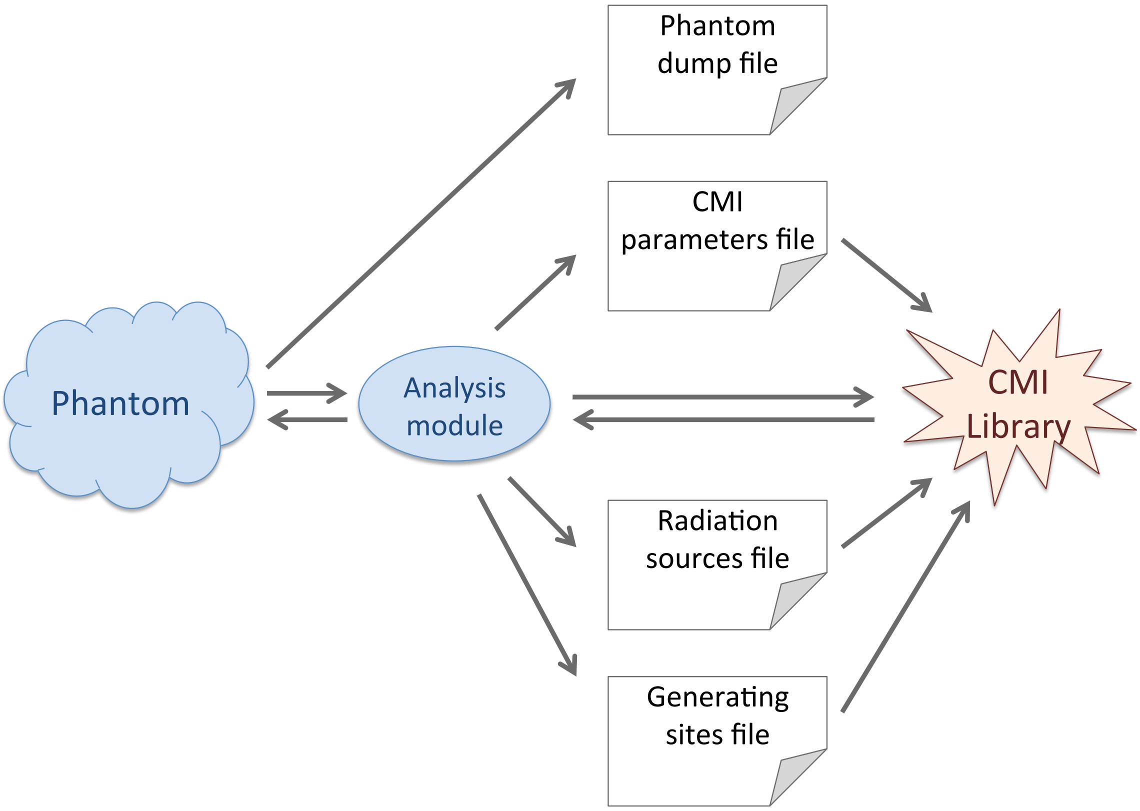

Figure 5.1: A schematic of the relationship between different code parts involved in the live radiation hydrodynamics scheme combining Phantom and CMacIonize (shortened to CMI in the schematic).

Alongside its hydrodynamics calculations, Phantom can execute an analysis module, which is typically used to post-process the SPH outputs (dump files). There is a "live analysis" option, which ensures that the analysis module is executed during Phantom's runtime. When the "live analysis" option is chosen, all of Phantom's data arrays can be modified by the analysis module each time a dump file is created. I have included the call to the CMacIonize library as part of the analysis module, as this was the least intrusive point of contact for the overall flow of Phantom. In order to establish contact with the library, the analysis module passes along information about the SPH particle positions, masses and smoothing lengths, and receives the neutral hydrogen fraction of each particle. The module then uses the neutral hydrogen fractions in order to modify the internal energies of the SPH particles in Phantom. Additionally, the analysis module generates a CMacIonize parameters file necessary for the execution of the MCRT, together with a file containing the positions and luminosities of the ionising sources and a file containing the positions of a set of Voronoi cell generating sites.

This particular way of linking the two codes has some clear advantages and disadvantages. Unfortunately the use of radiative feedback is currently tied to how often PHANTOM outputs dump files, which can result in an unnecessarily high output frequency when we want to perform CMACIONIZE more often. On the other hand, the simplicity of this setup reduces the probability of mistakes and inaccuracies, and in principle it can allow for the radiative transfer to be performed at each hydrodynamics time step.

5.2.2 Code modifications

Both PHANTOM and CMACIONISE are modular, allowing me to modify only small sections of them in order to achieve the goal of a radiation hydrodynamics scheme. The majority of the modifications were in CMACIONIZE and were related to ensuring that the density mapping from SPH particles onto a Voronoi grid was performed correctly. A detailed description of the changes that I made to the two codes is listed below.

Phantom

There are three ways in which I have made changes to PHANTOM. I have modified the Makefile in order to set some of the simulation parameters and to include the compiler flags that allow for the CMACIONIZE library to be called from PHANTOM. I have also written setup files in order to create the desired initial conditions for various simulation runs. Finally, I have written analysis files which realised the connection with CMACIONIZE. While the exact details of the first two modifications have limited contribution towards the clarity of this thesis, I will overview the structure of the analysis file in more depth.

A typical analysis file might create some additional files to enable the use of CMACIONIZE. As previously mentioned, these are a CMACIONIZE parameters file, a radiation sources file and a Voronoi generating sites file. Some of the simulations require the CMACIONIZE parameters file to be created only once. This file needs to be overwritten by the PHANTOM analysis only if the number of Vornoi generating sites has changed from one dump to the next or if the simulation boundaries have changed. The Voronoi cell generating sites file is typically overwritten at each dump, since we would like to follow the motion of the particles. And finally, the radiation sources file is only relevant, and hence only created, for the simulations in which we have multiple ionising sources (such as sink particles) and/or when the sources can change their positions.

After creating the additional files, a typical analysis file passes particle positions, smoothing lengths and masses to the CMacIonize library and it receives the neutral fraction of hydrogen for each particle. Using this information, new internal energies for the SPH particles are computed and assigned. The exact way of computing the energies differs between simulations and it will be additionally specified later in the chapter.

CMacIonize

The necessary changes to CMacIonize were more substantial and were primarily related to the way that the density mapping from the SPH particles to the Voronoi grid (or any other grid) was handled. The default density mapping in CMacIonize is done by computing the SPH density at the centroid of each cell, and assigning it as the average cell density. This approach is the one labeled as method 2 in Chapter 4. In order to compute the SPH density the method first finds all of the SPH particles that overlap with a cell's centroid by using a neighbours finding algorithm and an internal octree that CMacIonize constructs.

As discussed in Chapter 4, method 2 deviates from the true average cell density and does not conserve the total particle mass through the mapping. Due to these considerations, I have implemented the analytic density mapping presented in Chapter 4 and Petkova et al. (2018) into CMacIonize. The analytic density mapping finds the total mass contained in each grid cell, and then divides it by the cell volume in order to obtain an accurate average density. The cell mass is calculated by integrating the kernel function of each nearby SPH particle over the volume of the cell. The integrals are broken down into summation of the expression for I_P (see equation 4.90), evaluated for the position of each cell vertex. The computing time of the density mapping was improved by tabulating a set of I_P values, as described in Section 4.5.

In order to incorporate the analytic density mapping into CMacIonize, I have first written functions which pre-compute and tabulate I_P for a range of parameter values, and then interpolate I_P for any given set of parameters. An additional function sums up these interpolated values for I_P and obtains the accurate cell density. Since the density mapping method from Chapter 4 requires knowledge of all the particles that overlap with any part of the cell, I have adapted the neighbours finding algorithm to account for that. I have approximated the cell to a sphere with a centre at the centroid of the cell and with radius equal to the distance from the centroid to the furthest away vertex. I have then modified the neighbours finding algorithm to search for particles that intersect this sphere. The above modification ensures that all of the particles that overlap with parts of the cell are accounted for, and even through some unnecessary particles are detected by the search, their inclusion does not affect the outcome of the density calculation.

A further implemented change in CMacIonize handles the reverse mapping from the grid cells to the SPH particles. The default reverse mapping in CMacIonize was looping over all cells and for each cell it was assigning the cell's fraction of neutral hydrogen to the closest SPH particle. This type of reverse mapping does not take into account the SPH particle distribution, and in some cases it can leave some SPH particles without a neutral fraction value. Instead of using it, I have implemented a reverse mapping which does the opposite of the analytic density mapping of Chapter 4. Within the new reverse mapping method each SPH particle receives ionised material from the cells it has contributed mass towards. The fractional contribution of ionised material from a cell to a particle is the same as the fraction of mass that the particle has contributed towards the cell.

Finally, I have also made some minor modifications in the way that the input and the output of CMacIonize were handled. All of the code changes described in this section can be found on GitHub[1].

5.3 StarBench test

In order to confirm that the radiation hydrodynamics scheme is working correctly I will apply it to the well studied test problem of the D-type expansion of an H II region. I will then compare the outcome of my simulations to the results of other numerical schemes, which were presented in the StarBench paper (Bisbas et al. (2015)).

5.3.1 D-type expansion of an H II region

Bisbas et al. (2015) modelled and measured the radius of the ionisation front during the D-type expansion of an H II region and compared the numerical results to theoretical

---

[1] https://github.com/mapetkova/CMacIonize

predictions. Here I will start by overviewing the theoretical framework that they have used.

We have already discussed in Chapter 1 that the D-type expansion of an H II region begins when the ionised material has reached the Strömgren radius:

$$R_{St} = \left(\frac{3\dot{Q} m_p^2}{4\pi \alpha_B \rho_o^2} \right)^{1/3} . \tag{5.1}$$

The above equation is for pure atomic hydrogen gas, where $\dot{Q}$ is the number of ionising photons emitted per unit time, m_p is the proton mass, ρ_o is the density of the medium and α_B is the recombination coefficient.

Raga et al. (2012a) have derived that the radius of the ionisation front as a function of time, $R_{Sp}(t)$, evolves following the equation:

$$\frac{1}{c_i} \frac{\mathrm{d}R_{Sp}(t)}{\mathrm{d}t} = \left(\frac{R_{St}}{R_{Sp}(t)} \right)^{3/4} - \frac{\mu_i T_o}{\mu_o T_i} \left(\frac{R_{St}}{R_{Sp}(t)} \right)^{-3/4} , \tag{5.2}$$

where c_i is the sound speed in the ionised medium, μ_o and μ_i are the mean molecular weights of the neutral and the ionised gas, and T_o and T_i are the neutral and ionised temperatures respectively. They have obtained the above equation by considering the pressure balance between the ionised material and the shock, as well as that of the shock and the neutral medium.

We can argue that typically $\frac{\mu_i T_o}{\mu_o T_i} << 1$ since $T_i >> T_o$. Therefore, in early times $\frac{\mu_i T_o}{\mu_o T_i} \left(\frac{R_{St}}{R_{Sp}(t)} \right)^{-3/4}$ can be neglected and then equation 5.2 leads to the Spitzer solution for the radius of the ionisation front (Spitzer (1978)):

$$R_{Sp}(t) = R_{St} \left(1 + \frac{7}{4} \frac{c_i t}{R_{St}} \right)^{4/7} . \tag{5.3}$$

Raga et al. (2012b) improve on the work of Raga et al. (2012a) by including the inertia of the expanding shock as it propagates through the neutral medium. Their consideration leads to an expansion equation for the position of the shock, $R_{HI}(t)$, given by:

$$\frac{1}{c_i}\frac{\mathrm{d}R_{HI}(t)}{\mathrm{d}t} = \sqrt{\frac{4}{3}\left(\frac{R_{St}}{R_{HI}(t)}\right)^{3/2} - \frac{\mu_i T_o}{2\mu_o T_i}}. \tag{5.4}$$

Similarly, in early times $\frac{\mu_i T_o}{2\mu_o T_i}$ can be neglected, which is equivalent to neglecting the thermal pressure of the neutral gas. Equation 5.4 then leads to the Hosokawa-Inutsuka solution (Hosokawa and Inutsuka (2006)):

$$R_{HI}(t) = R_{St}\left(1 + \frac{7}{4}\sqrt{\frac{4}{3}}\frac{c_i t}{R_{St}}\right)^{4/7}. \tag{5.5}$$

In late times the neglected terms become significant, and eventually the expansion stagnates at $t = t_{STAG}$, characterised by $\dot{R}_{Sp}(t_{STAG}) = 0$ and $\dot{R}_{HI}(t_{STAG}) = 0$ for the two cases.

Using equations 5.2 and 5.4, we get the following stagnation radii, respectively:

$$R_{STAG,Sp} = R_{St}\left(\frac{c_i}{c_o}\right)^{4/3} \tag{5.6}$$

and

$$R_{STAG,HI} = R_{St}\left(\frac{8}{3}\right)^{2/3}\left(\frac{c_i}{c_o}\right)^{4/3}, \tag{5.7}$$

where c_o is the sound speed in the neutral gas.

5.3.2 Simulation setup

Following the StarBench paper (Bisbas et al. (2015)), I have used two different setups in order to test the simulated H II region expansion. Both of them model the early phase of the expansion and differ only in the size of the simulated region. For these tests I have arranged the SPH particles in a tightly pack hexagonal grid, filling up a cube with side R_{cl}, and total mass M_{cl}. The total number of particles, N, was chosen so that each particle would have a mass of $m_{part} \approx 10^{-3}$ M$_\odot$, and the initial Strömgren radius would contain roughly 10^4 particles (Bisbas et al. (2015)). The source of ionising radiation was positioned in the centre of the cube.

The parameters used in the two setups are presented in Table 5.1. Additionally, for the

Parameter	Units	Large test	Small tests
$\dot{Q}$	s^{-1}	10^{49}	10^{49}
μ_o		1	1
μ_i		0.5	0.5
ρ_o	g cm^{-3}	5.21×10^{-21}	5.21×10^{-21}
T_i	K	10^4	10^4
c_i	km s^{-1}	12.85	12.85
T_o	K	10^2	10^2
c_o	km s^{-1}	0.91	0.91
M_{cl}	$M_\odot$	1231	317.5
R_{cl}	pc	2.52	1.6
N		1,222,650	324,825
m_{part}	$M_\odot$	1.01×10^{-3}	0.98×10^{-3}

Table 5.1: Simulation parameters for the two models of early D-type expansion of an H II region.

radiative transfer we assumed that the photoionisation cross-section was $\sigma = 6.3 \times 10^{-18}$ cm^2, and the recombination coefficient was $\alpha_B = 2.7 \times 10^{-13}$ cm^3 s^{-1}, while disregarding the diffuse field.

CMacIonize was performed with 10 iterations of 10^6 photon packets, in order to compute accurate ionic fractions. In the beginning of the first iteration, the radiative transfer assumed that all cells were fully ionised, then it propagated the photon packets and it updated the ionic fractions at the end of the iteration, based on the photon packet trajectories. The same process was repeated during the following iterations with the new ionic fractions. The use of 10 iterations was demonstrated to ensure good convergence for a dataset of about 2.6×10^5 cells by Vandenbroucke and Wood (2018). To ensure that 10 iterations were sufficient for my own simulations, I have performed convergence tests, which are shown in Section 5.3.4.

For the hydrodynamics we used a fixed time step of about 0.85×10^{-3} Myr for all SPH particles and did not include gravity or magnetic fields. The equation of state was polytropic with $\gamma = 1.00011$ in order to mimic two-temperature isothermal gas, even though the exact value of γ does not affect the H II region expansion significantly according to the models included in Bisbas et al. (2015). The initial temperature of all gas particles was $T_o = 100$ K. After each time CMacIonize was called, the particles with ionic fraction higher than 0.5 had their internal energies set to correspond to a temperature of $T_i = 10^4$K and a mean molecular weight of $\mu_i = 0.5$.

In the larger simulation the density of the SPH particles was mapped onto a Voronoi grid, using the analytic method described in Chapter 4 and Petkova et al. (2018). For computational efficiency the values of the integral of the cubic spline kernel function were pre-computed and used for interpolation. The smaller simulation was performed multiple times, with three different types of density mapping (the one from Petkova et al. (2018), and method 1 and 2 from Chapter 4) in order to test how the density mapping affected the expansion rate.

5.3.3 Results

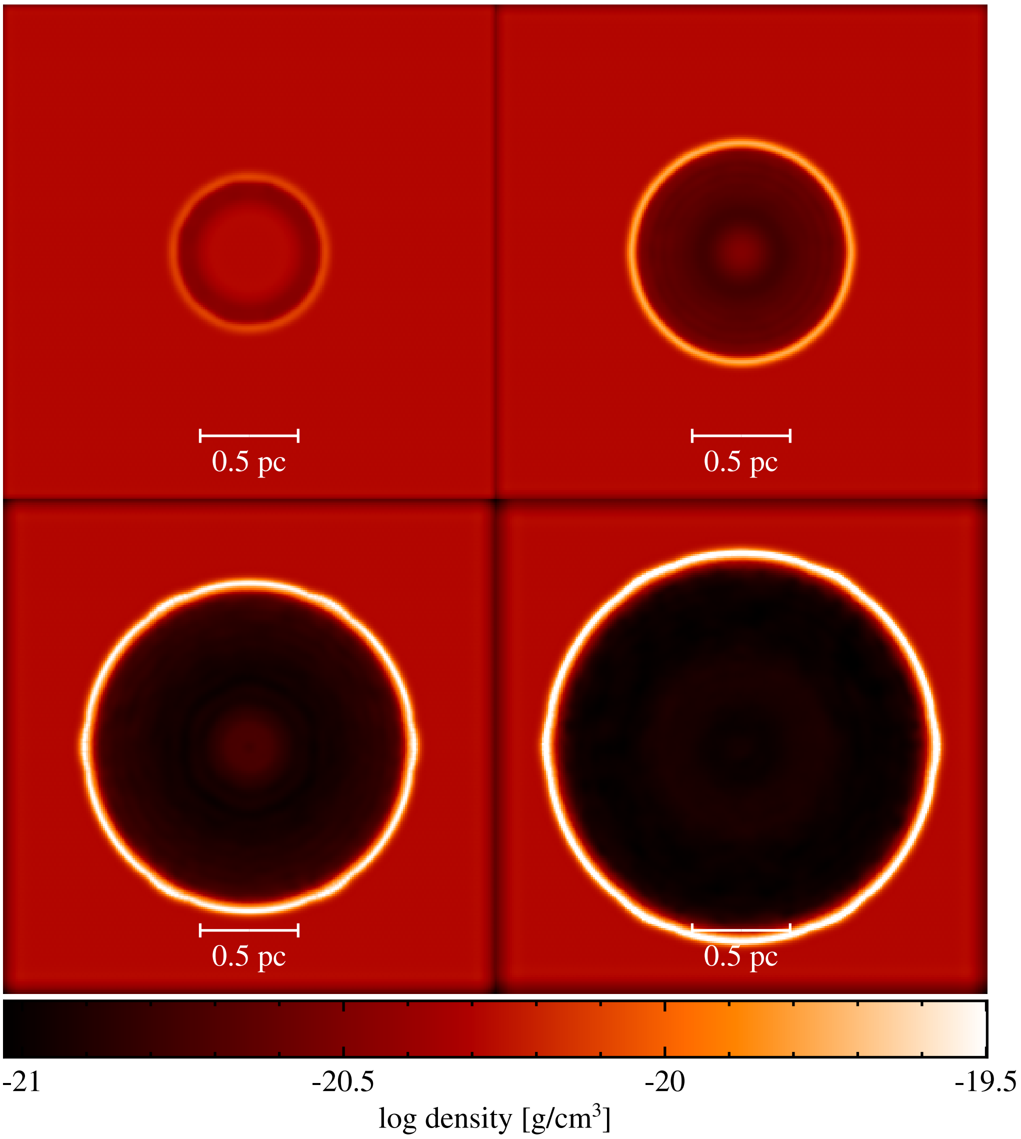

Figure 5.2: Density slices at $z = 0$ of the early D-type expansion of an H II region (large dataset), computed with PHANTOM + CMACIONIZE radiation hydrodynamics scheme. The snapshots are at $t = 0.005$ Myr *(top, left)*, $t = 0.02$ Myr *(top, right)*, $t = 0.05$ Myr *(bottom, left)*, $t = 0.07$ Myr *(bottom, right)*. This figure was created using SPLASH (Price (2007)).

Figure 5.2 shows density plots of the larger simulation at different times. We can see that qualitatively the D-type expansion is working correctly, as an initially spherical region is expanding rapidly and sweeping up a shell of shocked gas. In order to quantify the rate of expansion, I have determined the position of the ionisation front by selecting particles with ionic fractions between 0.45 and 0.55 and averaging their radial distance from the source. This procedure was repeated for each SPH snapshot and the time evolution of the ionisation front is presented in Figure 5.3, together with the analytic curves of the Spitzer and Hosokawa-Initsuka solutions (see equations 5.3 and 5.5), and the average empirically derived expansion curve from the StarBench paper (Bisbas et al. (2015)). Additionally, Figure 5.3 contains the relative error of the ionisation front radius compared to the theoretical solution.

A similar procedure as above was repeated for the smaller dataset (see Figure 5.4). This time the position of the ionisation front was calculated by averaging the radial distances from the light source of the particles with ionic fraction between 0.3 and 0.7. This modification was made in order to improve the uncertainties in determining the ionisation front (see the small spike at about 0.03 Myr in Figure 5.3).

5.3.4 Discussion

Larger setup

While qualitatively the modelling of the D-type expansion has been successful, it seems like the Phantom + CMacIonize live radiation hydrodynamics scheme produces a slower expansion rate than expected, as shown in Figure 5.3. The empiric StarBench solution is positioned between the Spitzer and the Hosokawa-Inutsuka solution, while the method of this chapter starts off under the Spitzer curve. A very similar behaviour, however, can also be seen in SEREN (Hubber et al. (2011)), which is the only SPH code used in the StarBench study. At very early times SEREN produces an ionisation front radius smaller than the Spitzer solution, which is then surpassed after about 0.04–0.05 Myr (Bisbas et al. (2015)). The same can be seen in Figure 5.3, which suggests that this behaviour can be related to the nature of SPH itself.

Indeed, Bisbas et al. (2015) reported that SEREN produced a thicker shell of shocked gas relative to the other codes, which was the reason for the slower initial expansion. We

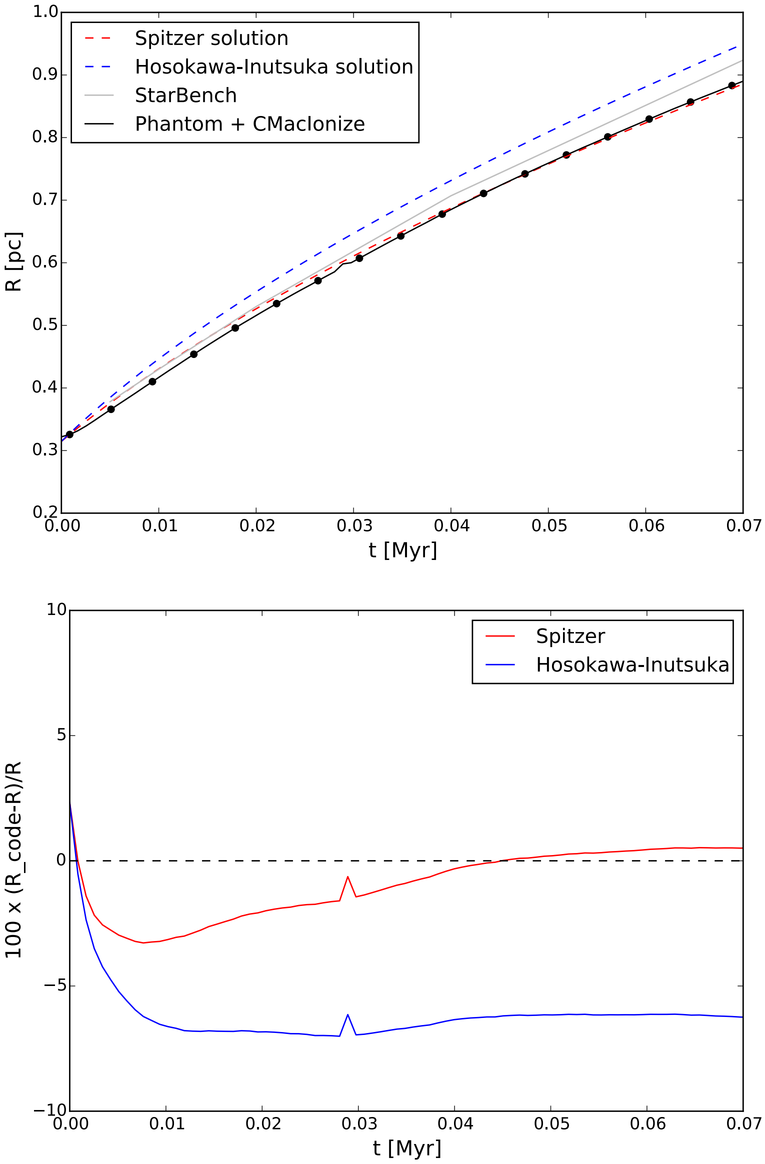

Figure 5.3: The early D-type expansion of an H II region (large dataset). *Top:* ionisation front radius as a function of time. The Spitzer and Hosokawa-Inutsuka solutions are shown in dashed lines, the averaged 3D StarBench solution is in solid grey and the Phantom + CMacIonize live radiation hydrodynamics scheme is shown in solid black line and black circles. *Bottom:* the relative error of the Phantom + CMacIonize solution compared to the Spitzer (red) and Hosokawa-Inutsuka (blue) solutions.

can see in Figure 5.5 that our model gives rise to a shock which is broad enough that it contains the analytically predicted positions of the ionisation front. The higher density inside the shock results in higher recombination rates, and hence this explains the initial smaller size of the H II region.

Looking closely at Figure 5.2, we can notice that the ionisation-driven shock is not exactly circular. There are small artefacts produced by the initial hexagonal arrangement

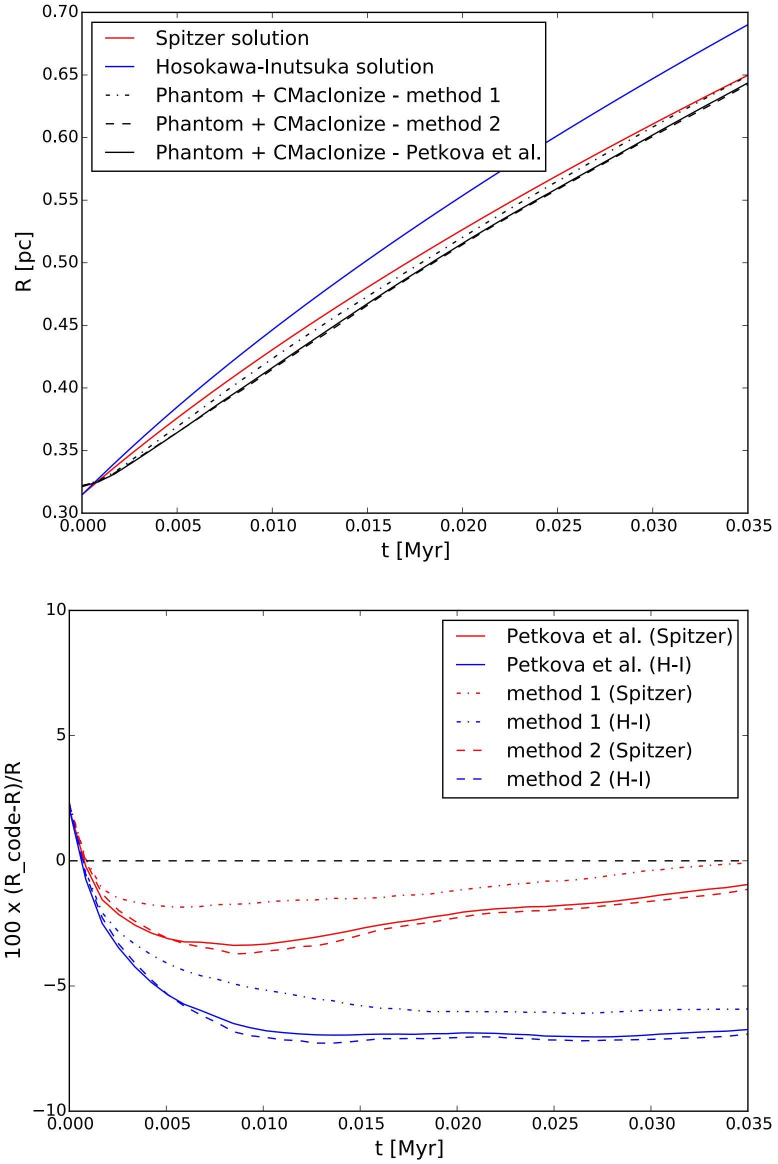

Figure 5.4: The early D-type expansion of an H II region (small dataset). *Top:* ionisation front radius as a function of time. The Spitzer and Hosokawa-Inutsuka solutions are shown in solid red and blue lines, and the Phantom + CMacIonize live radiation hydrodynamics scheme is shown in three black lines of different styles. The difference between the black lines comes from the different types of density mapping from the SPH particles onto a Voronoi grid (see Chapter 4 for more details). *Bottom:* the relative error of the Phantom + CMacIonize solution compared to the Spitzer (red) and Hosokawa-Inutsuka (blue) solutions for the three different types of density mapping.

of the SPH particles. These small errors can be avoided by using a 'glass' dataset, as was done by Bisbas et al. (2015). A glass is a crystalline arrangement, into which the SPH particles tend to relax, as it minimises their energy (see Price (2012) for more details). Even with this source of errors in mind, however, we see good agreement between the simulation performed with Phantom + CMacIonize, and the StarBench test, which indicates that

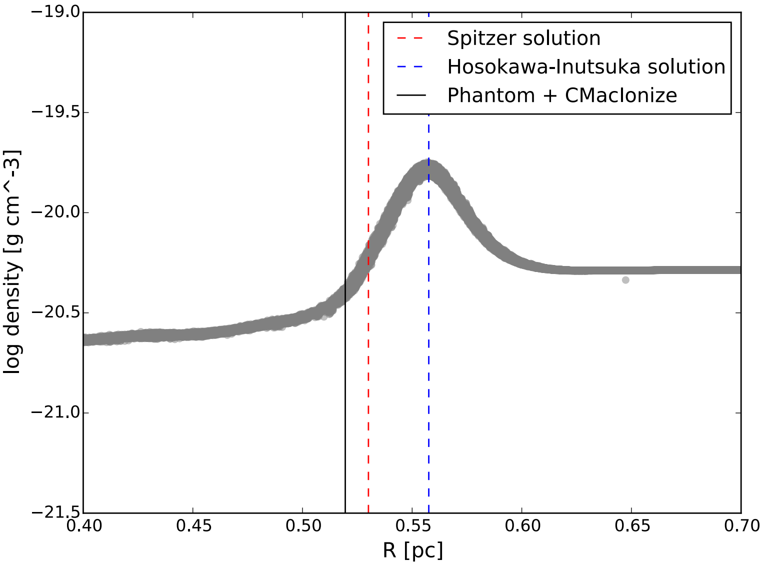

Figure 5.5: Density as a function of radial distance from the source of the D-type expansion model of an H II region, produced with Phantom + CMacIonize. The snapshot was taken at about 0.02 Myr from the beginning of the simulation. The vertical lines show the positions of the Spitzer and the Hosokawa-Inutsuka solutions, as well as the position of the ionisation front as calculated with Phantom + CMacIonize.

the live radiation hydrodynamics scheme of this thesis is performing correctly.

Smaller setup

In Figure 5.4 we can see how the analytic density mapping of Petkova et al. (2018) compares to method 1 (i.e. particle mass divided by cell volume) and method 2 (i.e. SPH density at the centroid of a cell) when they are applied to a hydrodynamics problem. We can see that the expansion rate obtained with method 2 is similar to the expansion produced with the exact density mapping of Petkova et al. (2018), and it seems to be underestimating the size of the H II region by a small amount. Method 1, on the other hand, results in a greater deviation from the expansion curve of the simulation done with the exact density mapping. These results are not surprising, since in Chapter 4 we saw that method 1 produces larger density errors than method 2.

From Figure 5.4 we can conclude that at least in simple geometries method 2 is a good approximation for the cell density, when dynamical problems are studied. We need further tests in order to find out if that remains the case in complex, non-spherically symmetric systems.

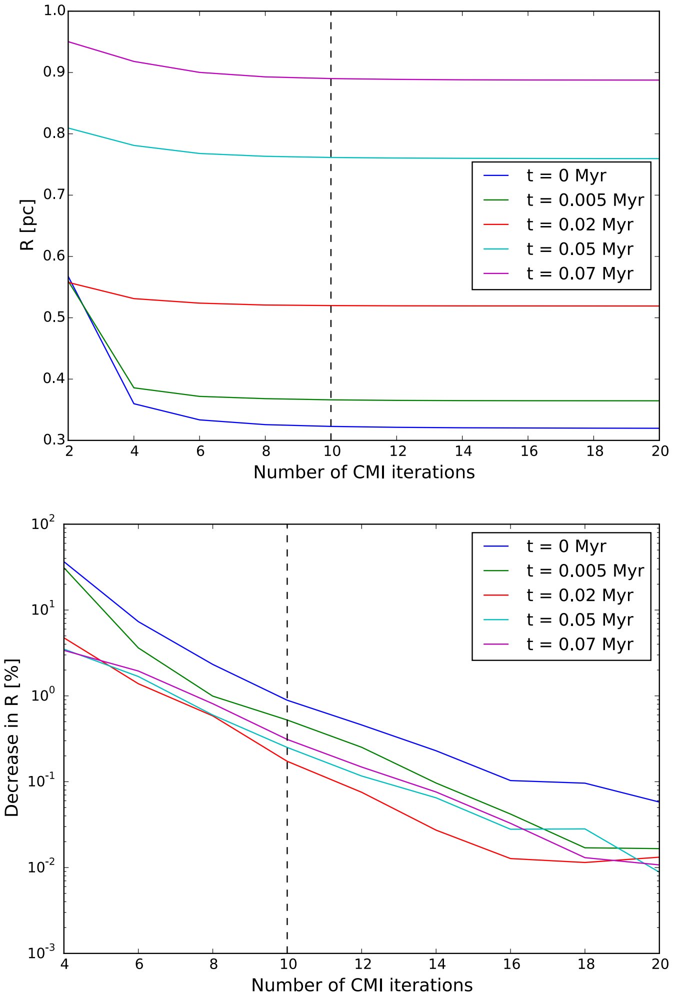

Figure 5.6: Convergence tests of CMacIonize at different simulation times for the large simulation setup. *Top:* ionisation front radius measured at each iteration. *Bottom:* The incremental decrease in ionisation front radius at each iteration. The vertical dashed line marks the number of iterations used for this benchmark simulation.

Convergence

To further confirm the validity of the above results, I have included radiative transfer convergence tests performed as post-processing of the larger dataset. Since both datasets have the same particle mass resolution, examining one of them is sufficient. After the larger simulation was completed, I have selected 5 different outputs (corresponding to 0 Myr, 0.005 Myr, 0.02 Myr, 0.05 Myr and 0.07 Myr) and for each of them applied CMacIonize with different numbers of iterations, ranging from 2 to 20. After each CMacIonize run, I calculated the ionisation front radius by averaging the radial distances of the particles with ionic fraction between 0.3 and 0.7. The results of these tests are shown in Figure 5.6.

Figure 5.6 demonstrates that all of the CMacIonize runs initially overestimate the

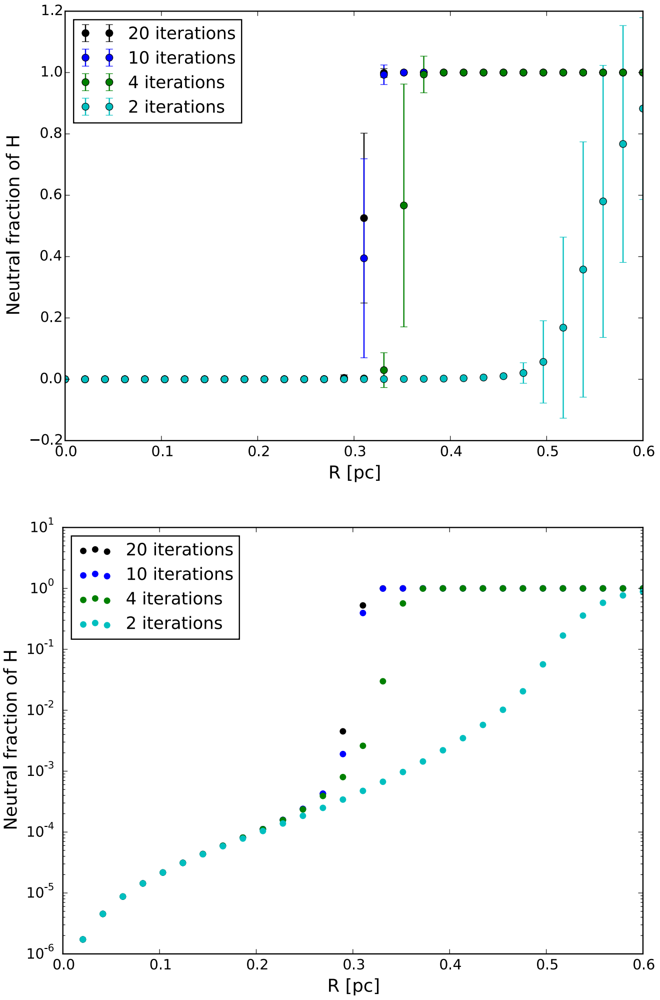

Figure 5.7: Radial profile of the neutral gas fraction of the large simulation setup at 0 Myr. The colours represent different numbers of iterations of CMacIonize. The gas particles were divided into radial bins for which the average neutral fraction was calculated. The top plot contains error bars corresponding to the scatter in each bin. The bottom plot contains the same data on a log scale without the error bars, for clarity.

value for the radius of the H II region, and this estimate gradually decreases later on. At earlier simulation times the system takes more iterations to converge to a value for the radius of the H II region, than at later times. At 10 iterations the decrease in the size of the H II region is at most 1% for all tested simulation times, which demonstrates sufficient convergence.

Finally, Figure 5.7 shows the neutral gas fraction as a function of radial distance for a small number of iterations at 0 Myr. This particular snapshot was selected since it was the slowest one to converge. As expected, we can see that the central region of the simulation has a neutral fraction of about 0, which transitions to 1 further away from the source. This transition region is much larger and more gradual at 2 iterations, and it becomes steeper as the number of iterations increases. The expectation is to have nearly a step function at the ionisation front. Figure 5.7 demonstrates that the difference between 10 and 20 iterations in that regard is minimal, which further comes to show a good convergence.

5.4 Feedback and star formation

5.4.1 Initial conditions

After demonstrating the accurate performance of the PHANTOM + CMACIONIZE live radiation hydrodynamics scheme with a test problem, I will now apply it to a more realistic star formation scenario. For that I will use some of the data presented in Dale et al. (2012) as initial conditions for my simulation.

Dale et al. (2012) studied cloud dispersal due to ionising stellar radiation. In their paper they included star forming clouds spanning a range of initial masses, radii and velocity dispersions. The clouds were evolved with an SPH code (Bate et al. (1995)) until a sufficient number of sources of ionising radiation were formed, and afterwards photoionisation was included following a ray-tracing algorithm and a Strömgren volume technique, as described in Dale et al. (2007) and Dale and Bonnell (2011).

The equation of state that was used was a piecewise barotropic, with $P = k\rho^\gamma$ and

$$\gamma = \begin{cases} 0.75, & \rho \leq \rho_1, \\ 1.0, & \rho_1 \leq \rho \leq \rho_2, \\ 1.4, & \rho_2 \leq \rho \leq \rho_3, \\ 1.0, & \rho \geq \rho_3. \end{cases} \tag{5.8}$$

In the above $\rho_1 = 5.5 \times 10^{-19}$ g cm^{-3}, $\rho_2 = 5.5 \times 10^{-15}$ g cm^{-3} and $\rho_3 = 2.0 \times 10^{-13}$ g cm^{-3}.

This equation of state is meant to mimic the effect of line cooling in the lowest density range, followed by the isothermal phase of dust cooling at ~ 10 K, as the density increases in the early stages of cloud collapse. The third density range represents the optically thick phase reached later on in the process of star formation. The final isothermal region is added for the correct creation of sink particles, and it roughly corresponds to the star reaching a hydrostatic state. The critical density criterion for sink creation is within he second density range, however if some of the other sink creation criteria are not locally met, a clump of gas can continue to increase its density without becoming a sink. This process can move the clump into the third density range for γ, where the energy and temperature increase with density. Without the final isothermal region the temperature of the clump could keep increasing to unphysical values and completely prevent the sink particle formation.

For my work, I have used the dataset from Dale et al. (2012) corresponding to the cloud which suffered the greatest disruption from feedback. The dataset is labeled as 'Run I' and has a total mass of 10^4 $\mathrm{M}_\odot$, an initial radius of 10 pc, and initial velocity dispersion of 2.1 km s^{-1}. The dataset contains 10^6 SPH particles, which is a suitable resolution for the sinks to represent individual stars.

By including ionising radiation in Run I, Dale et al. (2012) found that the gas became ionised in a roughly bipolar region around the central stellar cluster of the cloud. The mass fraction of the ionised gas almost immediately reached 0.03 and gradually increased to 0.1 by the end of the simulation, 2.2 Myr after the ionisation was switched on. Even with this modest ionisation fraction, about 58% of gas/stars were estimated to be unbound, and a vast cavity was carved out around the ionising sources. Several pillars were positioned at the outskirts of the cavity, pointed towards the most massive stellar cluster, and containing newly formed stars at their tips.

5.4.2 Simulation setup

I have used the dataset from Run I evolved to the point where there were already three sources of ionising radiation (i.e. three sinks of masses ≥ 20 $\mathrm{M}_\odot$) as the initial conditions for my radiation hydrodynamics scheme. In this thesis, I present a first comparison with a simplified version of the simulation due to time constraints. There were three ways in which I have performed simplifications:

1. I only used a subset of Run I, encompassing the densest part of the cloud (see Figure 5.8). This region contains all (44) of the sinks and 322,532 SPH gas particles, and has mass of 3526.56 $\mathrm{M}_\odot$, which is approximately 35% of the total cloud mass.

2. I adopted the initial internal energies of the particles from Run I, however, I did

not continue using the piecewise barotropic equation of state, as this would have required further modifications of PHANTOM. I have instead used an adiabatic equation of state with $\gamma \approx 1$ for the radiation hydrodynamics.

3. I did not use the analytic density mapping of SPH particles on the Voronoi grid. The cell density was assumed to be the SPH density at the centroid of the cell, and this type of density mapping was labeled as 'method 2' in Chapter 4.

The ionising feedback of stars with masses under 20 $\mathrm{M}_\odot$ was neglected, just as in Dale et al. (2012), and the photon fluxes, $\dot{Q}$, from the most massive stars were calculated using the formula:

$$\log(\dot{Q}) = 48.1 + 0.02(M_\star - 20\mathrm{M}_\odot). \tag{5.9}$$

CMACIONIZE was performed once every 1.5–7.5 $\times 10^{-3}$ Myr, and it used 10 iterations of 10^6 photon packets each time. The neutral fraction of each particle was used to linearly interpolate between 7.5 K and 10,000 K and the corresponding new internal energy was assigned to the particle only if it exceeded the particle's previous internal energy. This ensures that neutral particles can become ionised, however, an ionised particle cannot return to a neutral state.

New sink particles were created by PHANTOM when the density was above 2.34×10^{-16} g cm^{-3} and the energy and angular momentum criteria were met. This was set to match the sink formation process of Dale et al. (2012).

5.4.3 Results

From the moment when the ionising radiation was switched on, small pockets of hot gas (≈ 0.02 pc) appeared around the sources and disrupted the accretion disks that had been formed around the most massive stars. As the ionising sources exited the densest parts of the cloud, the regions of ionised gas increased in volume, and in number of SPH particles (see Figure 5.9).

The full ionisation history of the simulation is shown in Figure 5.10. In the figure we can see a gradual removal of gas (in red), as it becomes accreted by stars (in green). The mass fraction of ionised particles, as computed by CMACIONIZE at each SPH snapshot,

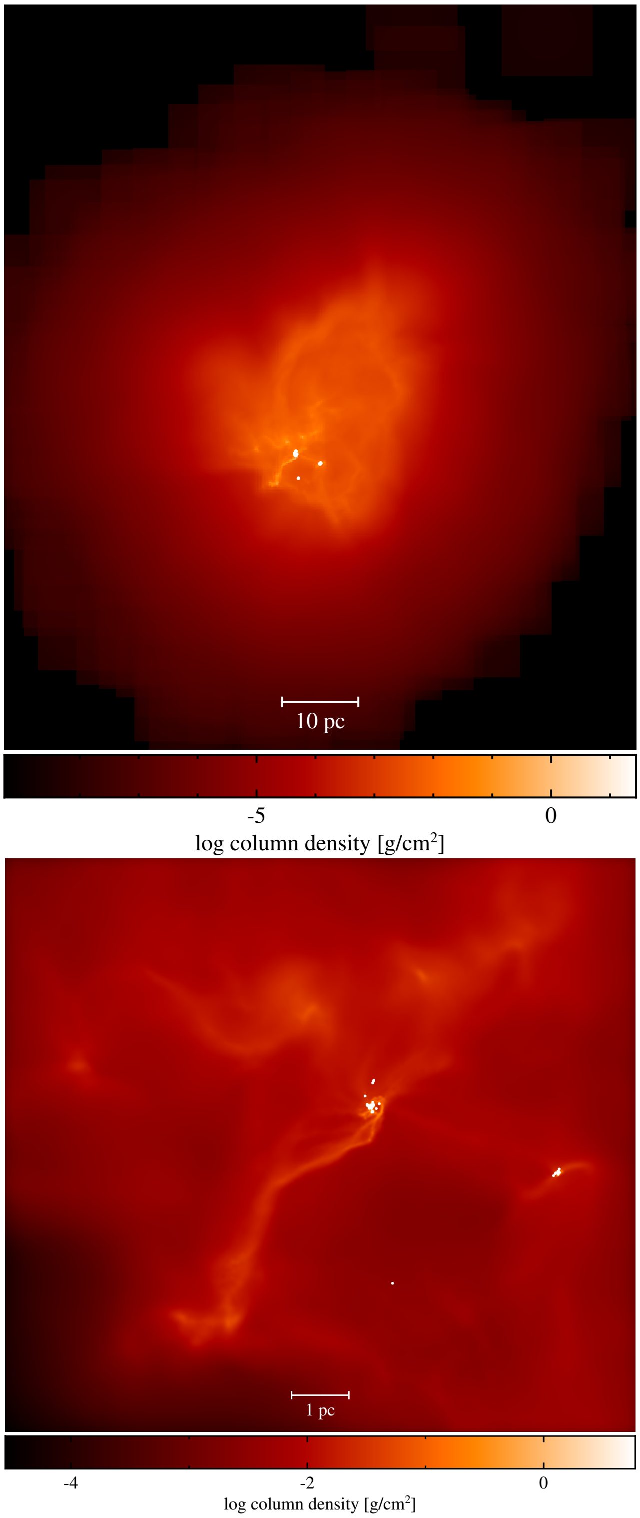

Figure 5.8: Column density plots of the full simulation volume of Run I from Dale et al. (2012) (*top*) and the subset used for modelling ionising feedback with Phantom + CMacIonize (*bottom*). The sink particles are plotted in white and scale lengths of 10 pc and 1 pc respectively are included in the plots. This figure was created using SPLASH (Price (2007)).

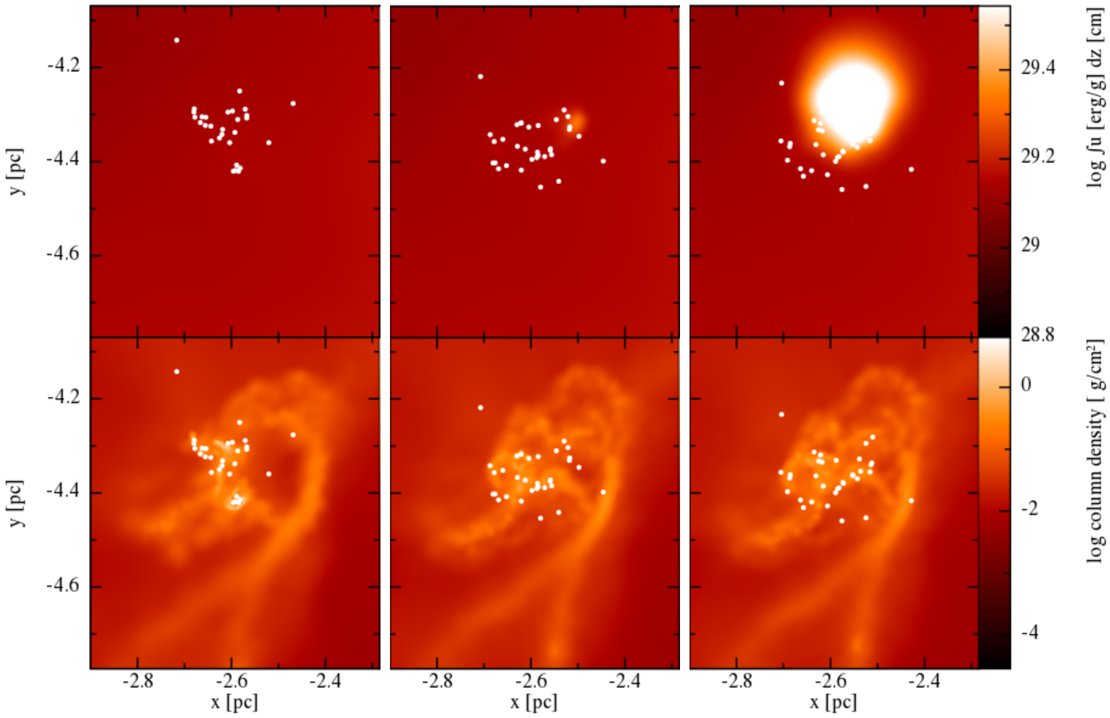

Figure 5.9: Snapshots of the stellar cluster formed in the central region of the simulation at 0 Myr (*left*), 0.045 Myr (*middle*) and 0.053 Myr (*right*) after the ionisation was switched on. The top panels show the specific internal energy integrated along the z-axis, and the bottom panels show column density, also obtained by integrating along the z-xis. The white markers indicate the positions of the sink particles. The figure demonstrates the rapid increase in the size of the ionised region between two consecutive SPH snapshots, likely due to geometric factors. This figure was created using SPLASH (Price (2007)).

is shown in blue. We can see that this curve fluctuates rapidly likely due to 3D geometric factors, however, this behaviour is not reflected within the SPH code, since the particles are not allowed to recombine. The inability of the SPH particles to recombine and return to a neutral state after being ionised means that SPH experiences monotonically increasing amounts of ionised gas. An upper estimate of the amount of ionised gas in the SPH simulation is given by the dashed curve, labelled as modified gas. The modified gas is a counter of each instance when the calculated internal energy due to ionisation was greater than the internal energy of the particle, and the particle's energy was modified as a result. This is an upper limit, and not an accurate measure of the number of ionised particles, as a single particle can have its internal energy increased multiple times.

The star forming cloud was evolved for about 0.4 Myr, out of the intended 2.2 Myr, due to the time constraints of the project. Within the code's runtime, the ionisation was not able to change the cloud morphology, since the ionised gas was being dragged along

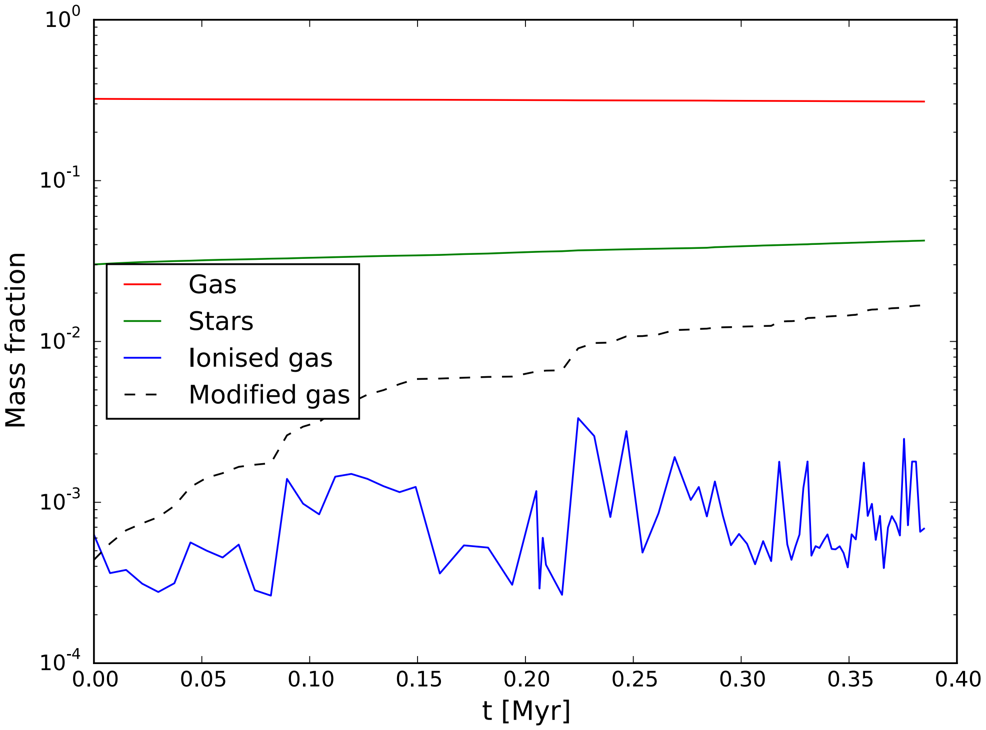

Figure 5.10: Time evolution of the gas depletion (red), star formation efficiency (green), and the ionised gas fraction (blue). The dashed line shows the number of instances of the internal energies of particles being increased due to ionisation, expressed as a gas mass. All quantities are normalised to the mass of the original cloud, i.e. 10^4 $M_\odot$.

by the gravitational collapse[2].

The volume occupied by the ionised gas is roughly bipolar in shape, with the dense, disk-like central region remaining mostly neutral. The accretion on the high mass stars has been steady despite the ionisation, as evidenced by the steady increase of the luminosity for most of the simulation. The jump in luminosity at about 0.325 Myr is due to a fourth sink becoming massive enough to be included as an ionising source. The luminosity increases from about 1.5 $\times 10^{49}$ s^{-1} to 2.5 $\times 10^{49}$ s^{-1} over the course of the simulation (see Figure 5.11). The highest luminosity value of our simulation already exceeds the total ionising luminosity reported by Dale et al. (2012) at 2.2 Myr, which indicates that our high mass stars are accreting more rapidly than in the Dale et al. (2012) simulation.

Most of the new sink particles have been forming in the vicinity of the ionised regions, although it is unclear if we are witnessing triggered star formation without additional data and more in-depth analysis. While the total sink mass increases very gradually, we can see in Figure 5.11 that the creation of sinks sometimes happens in bursts. Additionally,

---

[2] This is illustrated more clearly in the videos provided on the supplementary CD.

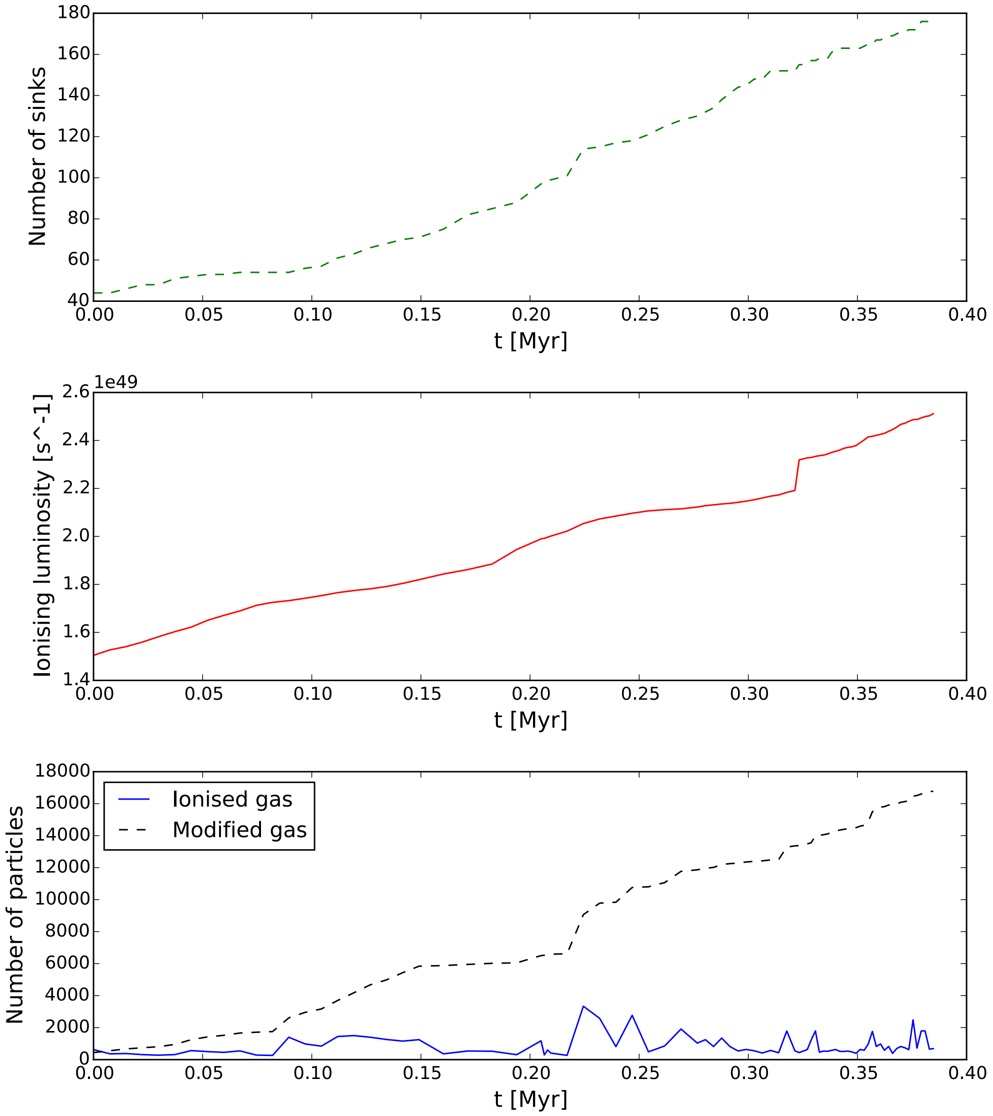

Figure 5.11: Time evolution of the number of sinks (green, dashed), total ionising luminosity (red), and the ionised gas fraction (blue). The dashed black line shows the fraction of the particles whose internal energies were increased due to ionisation.

the newly formed sink particles preferentially occupy lower mass ranges. In Figure 5.12 we can see that the shape of the IMF changes between the start point and the end point of the simulation due to the creation of many low mass sinks.

5.4.4 Discussion

The surprising outcome of this simulation is the fact that it did not manage to significantly alter the cloud morphology. Dale et al. (2012) reported that their method, when applied to this particular setup, resulted in carved out cavities and pillars containing newborn stars, as observed around real-life H II regions. Meanwhile, the ionised gas in the above

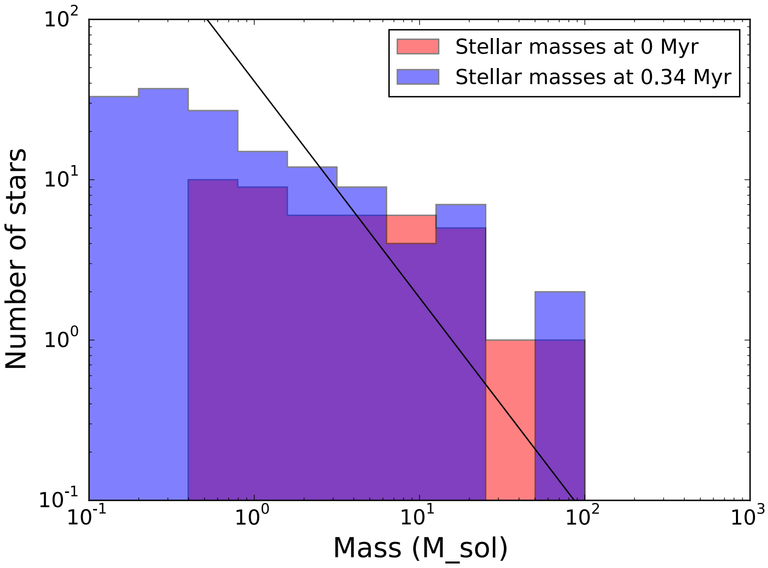

Figure 5.12: Stellar mass functions at 0.0 Myr (red) and 0.34 Myr (blue). The Salpeter mass function (Salpeter (1955)) is plotted in black line.

simulation did not show any signs of expansion, which was unexpected and could be due to several factors.

First, it could be an inaccuracy in the radiation hydrodynamics implementation. The StarBench test assured us that the most basic version of the scheme functioned correctly. There were, however, a few key differences between the benchmark and the star formation simulation. In addition to the basic functionality of the former, the latter included self-gravity, sink particles and multiple radiation sources. While the individual performance of these features has been studied outside the scope of this radiation hydrodynamics scheme by other authors (see Price et al. (2017) and Vandenbroucke and Wood (2018)), it might still be beneficial to construct simple tests demonstrating that they function correctly when combined within the scheme.

Another, more likely, group of reasons for the discrepancy with Dale et al. (2012) has to do with the way that the radiation hydrodynamics scheme was set up to perform the star formation simulation. It is possible that the 10 iterations of 10^6 photon packets were not sufficient to allow for the convergence of CMacIonize. This could be at least a partial explanation of the fluctuations of ionised particle numbers shown in Figure 5.10, although these can also be due to geometric effects. If the ionic fractions are not well converged, this would affect mostly the ionisation front particles, which are more numerous in an irregularly-shaped H II region then in a spherical one, making the inaccuracy more pronounced in this simulation. What we learned about numerical convergence of radiative transfer from Figure 5.7, however, is that the earlier iterations tend to overestimate the size of the ionised region, which is inconsistent with the fewer ionised particles that we observed in comparison to Dale et al. (2012).

Another setup issue could be arising from the simplistic choice of generating sites for the Voronoi grid. As discussed in Chapter 3, taking the SPH particle positions as cell generating sites can create poor resolution at regions of high density gradient. Such numerical issues can lead to shadowing effects (Koepferl et al. (2016)), and artificial restrictions to the size of the HII region, such as the one we believe to be observing here. Unfortunately, including the cell insertion algorithm from Chapter 3 into the radiation hydrodynamics scheme was outside the timeframe of this PhD project and it will be attempted in a further study.

Finally, it is also possible that the discrepancy is at least partially due to the different hydrodynamics and radiation treatment between this work and Dale et al. (2012). As previously discussed, we have used an isothermal equation of state, whereas Dale et al. (2012) used a complex barotropic one, and this difference might be affecting the hydrodynamics. On the other hand, the lower ionised mass fraction in our simulation compared to Dale et al. (2012) might not be purely caused by shadowing, but it could also be a result of the different modelling of ionising radiation. If these factors are indeed causing significant differences to occur, the results of this simulation could be of value to the study of cloud dispersal due to stellar feedback.

Another important point to address is that there were multiple adjustments to the simulation during its runtime. Above a certain time step size, the execution of PHANTOM would be automatically aborted due to energy conservation errors (see Section 2.2.11 of Price et al. (2017) for more details). Because of that the time step (also affecting the frequency of execution of CMACIONIZE) had to be decreased multiple times during the simulation. Additionally, at certain occasions the Voronoi grid construction algorithm had difficulties handling the huge range of particle/cell separations. This led to the forced removal of grid generating sites, where the local density would exceed a threshold value of 6.77×10^{-13} g cm^{-3}. This modification was establish from about 0.2 Myr onward and it caused the omission of approximately 100 grid cells at each execution of CMACIONIZE.

There are many aspects of this simulation that can be improved upon. Both of the runtime adjustments, which were mentioned should be implemented from the beginning of the simulation for consistency. Furthermore, the discrepancy with the outcome of Dale et al. (2012) should be thoroughly investigated, and different choices of Voronoi cell generating sites should be explored. Overall, however, this early stage of live radiation hydrodynamics application has shown promising first results.

5.5 Computing time

The computing time is one of the main points which need to be addressed when assessing whether the use of the radiation hydrodynamics scheme is feasible. As discussed in the very beginning of the thesis, it is the computing cost which has been preventing researchers from combining SPH and MCRT so far. In the light of that consideration, I will now examine how the different parts of the radiation hydrodynamics compare in terms of CPU time.

In the case of the large D-type expansion test, the simulation is clearly dominated by the execution of CMacIonize (see Figure 5.13). The most expensive step within CMacIonize is the density mapping, which is performed twice — once when transitioning from the SPH particles to the grid, and then when going back from the grid to the particles. Much less time, in comparison, is required for the construction of the Voronoi grid and for the execution of the Monte Carlo photon packet propagation. Even though the grid construction is considered to be an expensive step, the simple geometry of the mostly evenly spaced particles kept it at a low cost. As the ionised gas expanded and created a more prominent shock, the computing time for all of the radiative transfer parts increased. Meanwhile, the simple nature of the hydrodynamics — not considering gravity or other costly effects — kept the computing time of Phantom at a low number throughout.

Since the exact density mapping of Petkova et al. (2018) requires a substantial amount of computing time, it is of interest to look at how fast the radiation hydrodynamics performs when we use approximations for the cell density. The computing cost between each two consecutive SPH snapshots of the small D-type expansion test is shown in Figure 5.14. We can see that using either method 1 or method 2 reduces the computing time of CMacIonize substantially, however the reduced computing time is still about 3 times

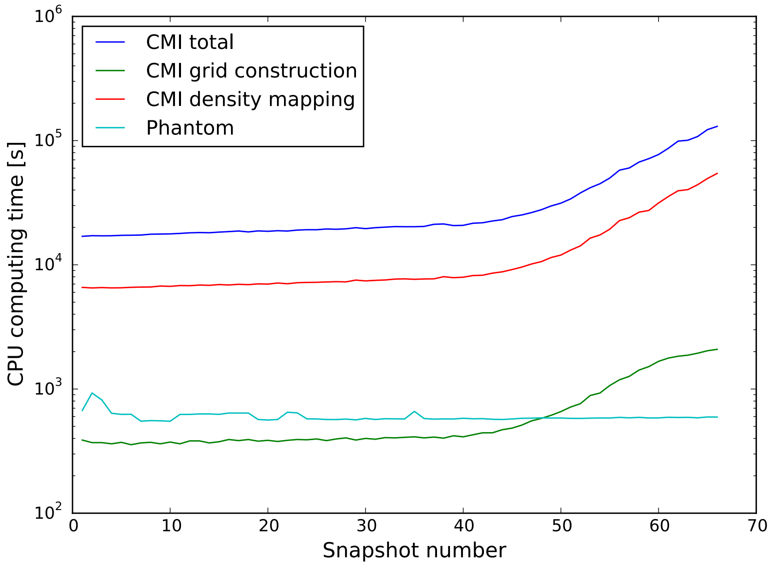

Figure 5.13: CPU time between two consecutive outputs (snapshots) of the large D-type expansion test. The total CPU time is split up between CMacIonize and Phantom. The CMacIonize grid construction and density mapping contributions are also displayed separately. Note that the CPU time of a single density mapping procedure is displayed, and at each CMacIonize run the density mapping is performed twice.

longer than the CPU time required by Phantom.

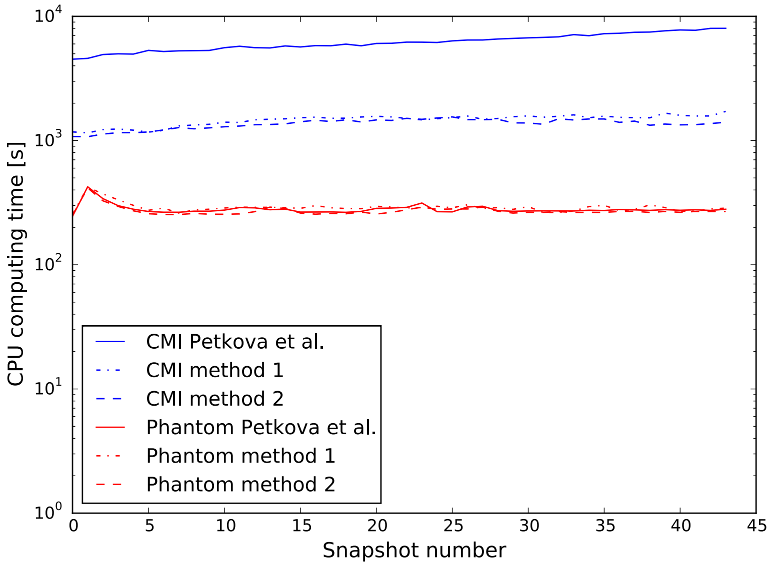

Figure 5.14: CPU time between two consecutive outputs (snapshots) of the small D-type expansion test. The total CPU time is split up between CMacIonize and Phantom. The different line styles signify the three different types of density mapping.

The star formation simulation, on the other hand, exhibits a very different computing time breakdown (see Figure 5.15). The inclusion of gravity substantially increased the time expenditure of Phantom, as compared to CMacIonize. The computational cost of the hydrodynamics oscillates, depending on the variable star formation rate and the sink particle creation. At later times, PHANTOM clearly dominates the CPU time breakdown. Meanwhile, CMACIONIZE is primarily spending time on the Voronoi grid construction (about 97%), as this appears to be the computational bottleneck of the radiative feedback algorithm. Figure 5.15 shows a simulation which used the density mapping of method 2, however the total CMACIONIZE computing time remains within the same order of magnitude even if the exact density mapping is adopted.

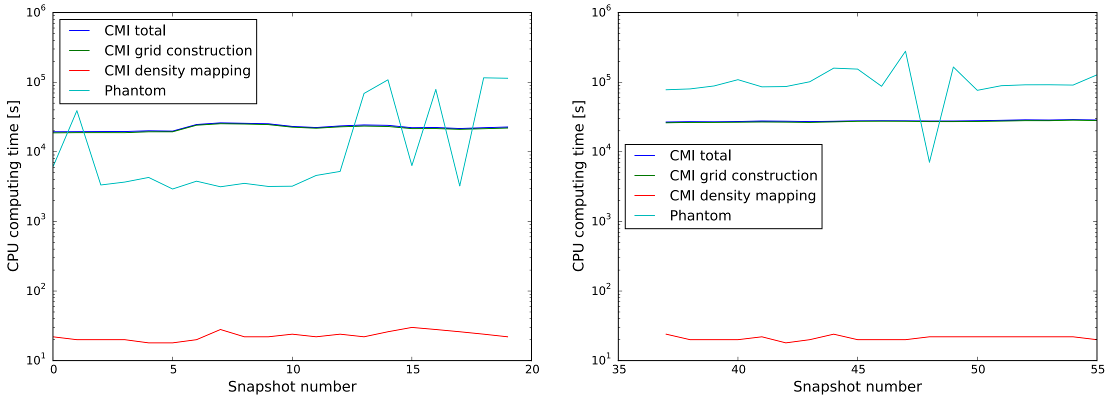

Figure 5.15: CPU time between two consecutive outputs (snapshots) of the star formation with feedback simulation. The early time outputs (*left*) are 0.0075 Myr apart, and the late time ones (*right*) are separated by half of that time. The total CPU time is split up between CMACIONIZE and PHANTOM. The CMACIONIZE grid construction and density mapping contributions are also displayed separately. Note that the CPU time of a single density mapping procedure is displayed, and at each CMACIONIZE run the density mapping is performed twice. Additionally, the blue and green lines are almost indistinguishable since the CMACIONIZE computing time is dominated by the grid construction.

Finally, all of the above computing cost observations let us conclude that the relative performance of the SPH and MCRT codes is highly problem specific. For the scenarios of great research interest, such as star cluster formation, the ionising feedback does not add an impossible computational overhead, and therefore the radiation hydrodynamics scheme proposed in this thesis can be considered as successful and feasible.

Chapter 6

Conclusion and Future Work

Conclusion

In this thesis I have presented the successful combination of a Smoothed Particle Hydrodynamics code (Phantom, Price et al. (2017)) with a Monte Carlo Radiative Transfer method (CMacIonize, Vandenbroucke and Wood (2018)), as a live radiation hydrodynamics scheme. In order to run the two codes simultaneously, I have used a Voronoi tessellation as a linking step. At regular time intervals the SPH code would call the MCRT code for computing the ionisation of the simulated cloud. Whenever this happens, the SPH densities would be mapped onto a Voronoi grid, which is used for propagating photon packets by the MCRT. After the ionic fraction is computed for each Voronoi cell, it is then mapped back onto the SPH particles and used for modifying their internal energies.

As a first step towards achieving the radiation hydrodynamics scheme, I have discussed the construction of the Voronoi grid, and in particular the choice of a set of generating sites as seeds for the cell locations. I have demonstrated that using the SPH particle positions as generating sites does not yield satisfactory results when there are large density gradients present in the SPH dataset. I have also proposed an algorithm for improving the grid resolution at regions of high density gradients by strategically adding a small number of additional cells. This algorithm does not require a lot of computing time to perform (it depends roughly linearly on the number of SPH particles), however time constraints with the PhD project has prevented me from incorporating it into the live radiation hydrodynamics scheme.

In the following chapter I have addressed the next preliminary step for the successful combination of SPH and MCRT, which was the density mapping from SPH onto the Voronoi grid. I have developed a novel method for performing this mapping, which calculates the exact average grid cell density and ensures mass conservation (Petkova et al. (2018)). The method is based on the analytic integration of the SPH kernel function in order to determine what fraction of the mass of each SPH particle contributes to a given cell. I have derived the integral of a set of selected kernel functions — the first three B-splines (Schoenberg (1946), Monaghan and Lattanzio (1985)) and the first three of the Wendland series (Wendland (1995)) — over the area of a random polygon in 2D and a random polyhedron in 3D. I have then produced a computer implementation of this density mapping method, in which I pre-compute the values of the kernel integrals in order to speed up the calculations. The computer implementation was thoroughly tested against the results of a numerical integration scheme in order to ensure its correct operation. I compared the proposed method to commonly used approximate ways of mapping SPH densities onto a grid, and found that the approximate methods can underestimate or overestimate the cell densities by up to 50%, depending on the SPH dataset.

Finally, I combined the SPH code Phantom with the photoionisation MCRT code CMacIonize into a live radiation hydrodynamics (RHD) scheme and tested its performance. I have applied the RHD scheme to the well studied problem of D-type expansion of an H II region and found excellent agreement with other research (Bisbas et al. (2015)). I have further repeated the same test with three different types of density mappings — the one described in Petkova et al. (2018), one in which the cell density is approximated by a particle's mass divided by the volume of the cell (method 1), and one where the average cell density is assumed to be the SPH density at the centroid of the cell (method 2). I found that unlike method 1, method 2 followed closely the D-type expansion rate that was produced by the exact density mapping of Petkova et al. (2018). I then proceeded to run a star formation simulation in which high mass stars in a cluster provided ionising feedback and were expected to disrupt the structure of the cloud they were embedded in. For this I have used the initial conditions of 'Run I' from Dale et al. (2012) and the density mapping of method 2, as it required less computing time than the exact density calculation of Petkova et al. (2018). The simulation was evolved for 0.4 Myr out of the intended 2.2 Myr (used by Dale et al. (2012)) due to time constraints with the project.

Within that timeframe, the high mass stars ionised a large volume of gas in a roughly bipolar distribution, in agreement with Dale et al. (2012). There was rapid star formation in the disk-like structure between the ionised regions, likely enabled by the pressure of the ionised gas. The newborn stars were preferentially of low mass and had a different mass distribution than the one reported by Dale et al. (2012) at 2.2 Myr. Unlike the work of Dale et al. (2012), the feedback in my simulation did not cause morphological changes of the star-forming cloud, which is likely due to the choice of Voronoi grid generating sites, and discrepancies in the hydrodynamics and/or the radiative transfer methods that were adopted.

Overall we have strong evidence that the basic live radiation hydrodynamics scheme introduced in this thesis is functioning correctly, although its use in more complex simulations require further studies. The addition of ionising feedback to the hydrodynamics creates a reasonable computing overhead when we model realistic astrophysical problems involving gravity. The radiation hydrodynamics scheme is therefore a promising new tool, which can be applied to a variety of models where ionisation is a dominant process.

Future work

The end point of this thesis offers many directions in which this work can be continued, and here I will list a small number of them. The first future work project is to complete the simulation of star formation with ionising feedback. So far it has only been evolved for 0.4 Myr, however continuing it until it reaches 2.2 Myr would allow for full comparison with the results of Dale et al. (2012). Furthermore, I plan to alter some of the parameters, such as the equation of state and the time step between consecutive ionisation calculations, and to study how robust the outcomes of the simulation are under these changes. After examining the performance of the radiation hydrodynamics scheme thoroughly, I intend to apply it to further problems of astronomical interest where ionisation is a dominant process, for example the the dispersal of molecular clouds that are internally or externally illuminated by ionising sources or accretion onto massive stars.

Another future work plan is to continue the development of the radiation hydrodynamics code by establishing a procedure for selecting a set of Voronoi grid generating sites to represent a particular SPH dataset. The algorithm presented in Chapter 3 is a great starting point, however it requires live radiation hydrodynamics tests in order to establish how useful it is.

Finally, I plan to incorporate the density mapping method introduced in Chapter 4 and Petkova et al. (2018) into the visualisation tool SPLASH (Price (2007)). There are two ways in which the mapping method can improve the functionality of SPLASH. The first one is by including an exact SPH mapping option when rendering images, and it will ensure that even low resolution images will have accurate pixel counts. The other way is by improving the SPH-to-grid option, which enables the interpolation of SPH parameter values onto a Cartesian grid, and can be used for comparing SPH to grid-based simulations.

Appendix A

Integrals of Other Kernel Functions

By following the derivation presented in Chapter 4 we can obtain the integral of any polynomial kernel function inside the volume (in 3D) or area (in 2D) of any polyhedron or polygon respectively. In this appendix I have included the solutions for six different commonly used kernels from two different families.

Kernel functions and notation

The kernels that I have selected are the first three B-splines (Schoenberg (1946), Monaghan and Lattanzio (1985)) and the first three of the Wendland series (Wendland (1995)). I have adopted the notation used in Dehnen and Aly (2012) in order to present the kernels in a shorter and more consistent way in Table A.1.

The kernel functions have non-zero values up to a radius of H, which corresponds to the unitless parameter $q = 1$. In order to construct the functions, the expressions in the second column of Table A.1 have to be multiplied by A/H^2 in 2D and A/H^3 in 3D. Finally, we also use the notation $(\cdot)_+ \equiv max\{0, \cdot\}$.

The cubic spline function is the one that was considered in detail in Chapter 4. To link between the current notation and the one in the chapter we can write $H = 2h$ and $q = r/H$.

Kernel name	Kernel function	A in 2D	A in 3D
Cubic spline	$b_4 = (1-q)_+^3 - 4(\frac{1}{2}-q)_+^3$	$\frac{80}{7\pi}$	$\frac{16}{\pi}$
Quartic spline	$b_5 = (1-q)_+^4 - 5(\frac{3}{5}-q)_+^4 + 10(\frac{1}{5}-q)_+^4$	$\frac{5^6 3}{2398\pi}$	$\frac{5^6}{512\pi}$
Quintic spline	$b_6 = (1-q)_+^5 - 6(\frac{2}{3}-q)_+^5 + 15(\frac{1}{3}-q)_+^5$	$\frac{3^7 7}{478\pi}$	$\frac{3^7}{40\pi}$
Wendland C^2	$\psi_{3,1} = (1-q)_+^4(1+4q)$	$\frac{7}{\pi}$	$\frac{21}{2\pi}$
Wendland C^4	$\psi_{4,2} = (1-q)_+^6(1+6q+\frac{35}{3}q^2)$	$\frac{9}{\pi}$	$\frac{495}{32\pi}$
Wendland C^6	$\psi_{5,3} = (1-q)_+^8(1+8q+25q^2+32q^3)$	$\frac{78}{7\pi}$	$\frac{1365}{64\pi}$

Table A.1: A set of commonly used kernel functions and their normalisation constants in 2D and 3D. The first three are B-splines and the other three are Wendland kernels, where the notation $(\cdot)_+ \equiv max\{0, \cdot\}$ (see Dehnen and Aly (2012)).

I_P in 2D

Under the new notation, the general form of I_P can be written as:

$$I_P = A\sum_{i=0}^{n} a_i J_i q_0^i, \tag{A.1}$$

where $q_0 = r_0/H$.

Therefore, in order to construct any solution I_P we need to know the non-zero a_i coefficients, which I have listed in Table A.2 and A.3. The general form of the corresponding J_i expressions has been described in Section 4.6.

I_P in 3D

In 3D the form of I_P can be written as:

$$I_P = A\sum_{i=-n}^{1} a_i I_i q_0^{1-i}, \tag{A.2}$$

where $q_0 = r_0/H$.

Once again, in order to construct any solution I_P we need to know the non-zero a_i coefficients, which are listed in Table A.5 and A.7. The I_i expressions have been presented

Kernel name	Range of q				Non-zero	a_i-s		
Cubic spline	$q \leq \frac{1}{2}$		$a_2 = \frac{1}{4}$		$a_4 = -\frac{3}{4}$	$a_5 = \frac{3}{5}$		
	$\frac{1}{2} < q \leq 1$	$a_0 = -\frac{1}{160}$	$a_2 = \frac{1}{2}$	$a_3 = -1$	$a_4 = \frac{3}{4}$	$a_5 = -\frac{1}{5}$		
	$q > 1$	$a_0 = \frac{7}{160}$						
Quartic spline	$q \leq \frac{1}{5}$		$a_2 = \frac{23}{125}$		$a_4 = -\frac{3}{5}$		$a_6 = 1$	
	$\frac{1}{5} < q \leq \frac{3}{5}$	$a_0 = \frac{1}{5^6 3}$	$a_2 = \frac{22}{125}$	$a_3 = \frac{8}{75}$	$a_4 = -\frac{6}{5}$	$a_5 = \frac{8}{5}$	$a_6 = -\frac{2}{3}$	
	$\frac{3}{5} < q \leq 1$	$a_0 = -\frac{727}{5^6 6}$	$a_2 = \frac{1}{2}$	$a_3 = -\frac{4}{3}$	$a_4 = \frac{3}{2}$	$a_5 = -\frac{4}{5}$	$a_6 = \frac{1}{6}$	
	$q > 1$	$a_0 = \frac{1199}{5^6 3}$						
Quintic spline	$q \leq \frac{1}{3}$		$a_2 = \frac{11}{81}$		$a_4 = -\frac{5}{9}$		$a_6 = \frac{5}{3}$	$a_7 = -\frac{10}{7}$
	$\frac{1}{3} < q \leq \frac{2}{3}$	$a_0 = \frac{5}{3^7 14}$	$a_2 = \frac{17}{162}$	$a_3 = \frac{25}{81}$	$a_4 = -\frac{35}{18}$	$a_5 = \frac{10}{3}$	$a_6 = -\frac{5}{2}$	$a_7 = \frac{5}{7}$
	$\frac{2}{3} < q \leq 1$	$a_0 = -\frac{251}{3^7 14}$	$a_2 = \frac{1}{2}$	$a_3 = -\frac{5}{3}$	$a_4 = \frac{5}{2}$	$a_5 = -2$	$a_6 = \frac{5}{6}$	$a_7 = -\frac{1}{7}$
	$q > 1$	$a_0 = \frac{239}{3^7 7}$						

Table A.2: A set of I_P coefficients for a series of different 2D spline kernel functions.

Kernel name	Range of q			Non-zero	a_i-s			
Wendland C^2	$q \leq 1$		$a_2 = \frac{1}{2}$		$a_4 = -\frac{5}{2}$	$a_5 = 4$	$a_6 = -\frac{5}{2}$	$a_7 = \frac{4}{7}$
	$q > 1$	$a_0 = \frac{1}{14}$						
Wendland C^4	$q \leq 1$		$a_2 = \frac{1}{2}$		$a_4 = -\frac{7}{3}$		$a_6 = \frac{35}{3}$	$a_7 = -\frac{64}{3}$
			$a_8 = \frac{35}{2}$	$a_9 = -\frac{64}{9}$	$a_{10} = -\frac{7}{6}$			
	$q > 1$	$a_0 = \frac{1}{18}$						
Wendland C^6	$q \leq 1$		$a_2 = \frac{1}{2}$		$a_4 = -\frac{11}{4}$		$a_6 = 11$	
			$a_8 = -\frac{231}{4}$	$a_9 = \frac{352}{3}$	$a_{10} = -\frac{231}{2}$	$a_{11} = 64$	$a_{12} = -\frac{77}{4}$	$a_{13} = \frac{32}{13}$
	$q > 1$	$a_0 = \frac{7}{156}$						

Table A.3: A set of I_P coefficients for a series of different 2D Wendland kernel functions (Wendland (1995)).

in Section 4.6.

Additionally, it is worth pointing out that $a_0 \in \{B_1, B_2, ...\}$, where:

$$B_j = \sum_{k=0}^{m} b_k q_0^k. \tag{A.3}$$

The non-zero b_k coefficients of each B_j have been listed in Table A.4 and A.6.

Kernel name	Range of q	B_j			Non-zero	b_i-s				
Cubic spline	$q \leq \frac{1}{2}$	B_1			$b_3 = -\frac{1}{12}$		$b_5 = \frac{3}{20}$	$b_6 = -\frac{1}{10}$		
		B_2		$b_1 = -\frac{1}{160}$	$b_3 = -\frac{1}{12}$		$b_5 = \frac{3}{20}$	$b_6 = -\frac{1}{10}$		
		B_3		$b_1 = -\frac{7}{160}$	$b_3 = -\frac{1}{12}$		$b_5 = \frac{3}{20}$	$b_6 = -\frac{1}{10}$		
	$\frac{1}{2} < q \leq 1$	B_2	$b_0 = -\frac{1}{960}$		$b_3 = -\frac{1}{6}$	$b_4 = \frac{1}{4}$	$b_5 = -\frac{3}{20}$	$b_6 = \frac{1}{30}$		
		B_3	$b_0 = -\frac{1}{960}$	$b_1 = \frac{1}{20}$	$b_3 = -\frac{1}{6}$	$b_4 = \frac{1}{4}$	$b_5 = -\frac{3}{20}$	$b_6 = \frac{1}{30}$		
	$q > 1$	B_3	$b_0 = \frac{1}{64}$							
Quartic spline	$q \leq \frac{1}{5}$	B_1			$b_3 = -\frac{23}{375}$		$b_5 = \frac{3}{25}$		$b_7 = -\frac{1}{7}$	
		B_2		$b_1 = \frac{1}{5^6 3}$	$b_3 = -\frac{23}{375}$		$b_5 = \frac{3}{25}$		$b_7 = -\frac{1}{7}$	
		B_3		$b_1 = -\frac{727}{5^6 6}$	$b_3 = -\frac{23}{375}$		$b_5 = \frac{3}{25}$		$b_7 = -\frac{1}{7}$	
		B_4		$b_1 = \frac{1199}{5^6 3}$	$b_3 = -\frac{23}{375}$		$b_5 = \frac{3}{25}$		$b_7 = -\frac{1}{7}$	
	$\frac{1}{5} < q \leq \frac{3}{5}$	B_2	$b_0 = \frac{2}{5^7 21}$		$b_3 = -\frac{22}{375}$	$b_4 = -\frac{2}{75}$	$b_5 = \frac{6}{25}$	$b_6 = -\frac{4}{15}$	$b_7 = \frac{2}{21}$	
		B_3	$b_0 = \frac{2}{5^7 21}$	$b_1 = -\frac{243}{5^6 2}$	$b_3 = -\frac{22}{375}$	$b_4 = -\frac{2}{75}$	$b_5 = \frac{6}{25}$	$b_6 = -\frac{4}{15}$	$b_7 = \frac{2}{21}$	
		B_4	$b_0 = \frac{2}{5^7 21}$	$b_1 = \frac{1198}{5^6 3}$	$b_3 = -\frac{22}{375}$	$b_4 = -\frac{2}{75}$	$b_5 = \frac{6}{25}$	$b_6 = -\frac{4}{15}$	$b_7 = \frac{2}{21}$	
	$\frac{3}{5} < q \leq 1$	B_3	$b_0 = -\frac{437}{5^6 21}$		$b_3 = -\frac{1}{6}$	$b_4 = \frac{1}{3}$	$b_5 = -\frac{3}{10}$	$b_6 = \frac{2}{15}$	$b_7 = -\frac{1}{42}$	
		B_4	$b_0 = -\frac{437}{5^6 21}$	$b_1 = \frac{1}{30}$	$b_3 = -\frac{1}{6}$	$b_4 = \frac{1}{3}$	$b_5 = -\frac{3}{10}$	$b_6 = \frac{2}{15}$	$b_7 = -\frac{1}{42}$	
	$q > 1$	B_4	$b_0 = \frac{128}{5^6}$							
Quintic spline	$q \leq \frac{1}{3}$	B_1			$b_3 = -\frac{11}{243}$		$b_5 = \frac{1}{9}$		$b_7 = -\frac{5}{21}$	$b_8 = \frac{5}{28}$
		B_2		$b_1 = \frac{5}{3^7 14}$	$b_3 = -\frac{11}{243}$		$b_5 = \frac{1}{9}$		$b_7 = -\frac{5}{21}$	$b_8 = \frac{5}{28}$
		B_3		$b_1 = -\frac{251}{3^7 14}$	$b_3 = -\frac{11}{243}$		$b_5 = \frac{1}{9}$		$b_7 = -\frac{5}{21}$	$b_8 = \frac{5}{28}$
		B_4		$b_1 = \frac{239}{3^7 7}$	$b_3 = -\frac{11}{243}$		$b_5 = \frac{1}{9}$		$b_7 = -\frac{5}{21}$	$b_8 = \frac{5}{28}$
	$\frac{1}{3} < q \leq \frac{2}{3}$	B_2	$b_0 = \frac{5}{2^3 3^8 7}$		$b_3 = -\frac{17}{486}$	$b_4 = -\frac{25}{324}$	$b_5 = \frac{7}{18}$	$b_6 = -\frac{5}{9}$	$b_7 = \frac{5}{14}$	$b_8 = -\frac{5}{56}$
		B_3	$b_0 = \frac{5}{2^3 3^8 7}$	$b_1 = -\frac{128}{3^7 7}$	$b_3 = -\frac{17}{486}$	$b_4 = -\frac{25}{324}$	$b_5 = \frac{7}{18}$	$b_6 = -\frac{5}{9}$	$b_7 = \frac{5}{14}$	$b_8 = -\frac{5}{56}$
		B_4	$b_0 = \frac{5}{2^3 3^8 7}$	$b_1 = -\frac{473}{3^7 14}$	$b_3 = -\frac{17}{486}$	$b_4 = -\frac{25}{324}$	$b_5 = \frac{7}{18}$	$b_6 = -\frac{5}{9}$	$b_7 = \frac{5}{14}$	$b_8 = -\frac{5}{56}$
	$\frac{2}{3} < q \leq 1$	B_3	$b_0 = -\frac{169}{2^3 3^7 7}$		$b_3 = -\frac{1}{6}$	$b_4 = \frac{5}{12}$	$b_5 = -\frac{1}{2}$	$b_6 = \frac{1}{3}$	$b_7 = -\frac{5}{42}$	$b_8 = \frac{1}{56}$
		B_4	$b_0 = -\frac{169}{2^3 3^7 7}$	$b_1 = \frac{1}{42}$	$b_3 = -\frac{1}{6}$	$b_4 = \frac{5}{12}$	$b_5 = -\frac{1}{2}$	$b_6 = \frac{1}{3}$	$b_7 = -\frac{5}{42}$	$b_8 = \frac{1}{56}$
	$q > 1$	B_4	$b_0 = \frac{10}{3^7}$							

Table A.4: A set of B_j coefficients used in calculating I_P for a set of 3D spline kernel functions.

Kernel name	Range of q			Non-zero	a_i-s				
Cubic spline	$q \leq \frac{1}{2}$		$a_0 = B_1$	$a_{-2} = \frac{1}{12}$		$a_{-4} = -\frac{3}{20}$	$a_{-5} = \frac{1}{10}$		
	$\frac{1}{2} < q \leq 1$	$a_1 = \frac{1}{960}$	$a_0 = B_2$	$a_{-2} = \frac{1}{6}$	$a_{-3} = -\frac{1}{4}$	$a_{-4} = \frac{3}{20}$	$a_{-5} = -\frac{1}{30}$		
	$q > 1$	$a_1 = -\frac{1}{64}$	$a_0 = B_3$						
Quartic spline	$q \leq \frac{1}{5}$		$a_0 = B_1$	$a_{-2} = \frac{23}{375}$		$a_{-4} = -\frac{3}{25}$		$a_{-6} = \frac{1}{7}$	
	$\frac{1}{5} < q \leq \frac{3}{5}$	$a_1 = -\frac{2}{5^7 21}$	$a_0 = B_2$	$a_{-2} = \frac{22}{375}$	$a_{-3} = \frac{2}{75}$	$a_{-4} = -\frac{6}{25}$	$a_{-5} = \frac{4}{15}$	$a_{-6} = -\frac{2}{21}$	
	$\frac{3}{5} < q \leq 1$	$a_1 = \frac{437}{5^6 21}$	$a_0 = B_3$	$a_{-2} = \frac{1}{6}$	$a_{-3} = -\frac{1}{3}$	$a_{-4} = \frac{3}{10}$	$a_{-5} = -\frac{2}{15}$	$a_{-6} = \frac{1}{42}$	
	$q > 1$	$a_1 = -\frac{128}{5^6}$	$a_0 = B_4$						
Quintic spline	$q \leq \frac{1}{3}$		$a_0 = B_1$	$a_{-2} = \frac{11}{243}$		$a_{-4} = -\frac{1}{9}$		$a_{-6} = \frac{5}{21}$	$a_{-7} = -\frac{5}{28}$
	$\frac{1}{3} < q \leq \frac{2}{3}$	$a_1 = -\frac{5}{2^3 3^8 7}$	$a_0 = B_2$	$a_{-2} = \frac{17}{486}$	$a_{-3} = \frac{25}{324}$	$a_{-4} = -\frac{7}{18}$	$a_{-5} = \frac{5}{9}$	$a_{-6} = -\frac{5}{14}$	$a_{-7} = \frac{5}{56}$
	$\frac{2}{3} < q \leq 1$	$a_1 = \frac{169}{2^3 3^7 7}$	$a_0 = B_3$	$a_{-2} = \frac{1}{6}$	$a_{-3} = -\frac{5}{12}$	$a_{-4} = \frac{1}{2}$	$a_{-5} = -\frac{1}{3}$	$a_{-6} = \frac{5}{42}$	$a_{-7} = -\frac{1}{56}$
	$q > 1$	$a_1 = -\frac{10}{3^7}$	$a_0 = B_4$						

Table A.5: A set of I_P coefficients for a series of different 3D spline kernel functions.

Kernel name	Range of q	B_j			Non-zero	b_i-s				
Wendland C^2	$q \leq 1$	B_1			$b_3 = -\frac{1}{6}$		$b_5 = \frac{1}{2}$	$b_6 = -\frac{2}{3}$	$b_7 = \frac{5}{14}$	$b_8 = -\frac{1}{14}$
		B_2		$b_1 = \frac{1}{14}$	$b_3 = -\frac{1}{6}$		$b_5 = \frac{1}{2}$	$b_6 = -\frac{2}{3}$	$b_7 = \frac{5}{14}$	$b_8 = -\frac{1}{14}$
	$q > 1$	B_2	$b_0 = \frac{1}{42}$							
Wendland C^4	$q \leq 1$	B_1			$b_3 = -\frac{1}{6}$		$b_5 = \frac{7}{15}$		$b_7 = -\frac{5}{3}$	$b_8 = \frac{8}{3}$
			$b_9 = -\frac{35}{18}$	$b_{10} = \frac{32}{45}$	$b_{11} = -\frac{7}{66}$					
		B_2		$b_1 = \frac{1}{18}$	$b_3 = -\frac{1}{6}$		$b_5 = \frac{7}{15}$		$b_7 = -\frac{5}{3}$	$b_8 = \frac{8}{3}$
			$b_9 = -\frac{35}{18}$	$b_{10} = \frac{32}{45}$	$b_{11} = -\frac{7}{66}$					
	$q > 1$	B_2	$b_0 = \frac{8}{495}$							
Wendland C^6	$q \leq 1$	B_1			$b_3 = -\frac{1}{6}$		$b_5 = \frac{11}{20}$		$b_7 = -\frac{11}{7}$	
			$b_9 = \frac{77}{12}$	$b_{10} = -\frac{176}{15}$	$b_{11} = \frac{21}{2}$	$b_{12} = -\frac{16}{3}$	$b_{13} = \frac{77}{52}$	$b_{14} = -\frac{16}{91}$		
		B_2		$b_1 = \frac{7}{156}$	$b_3 = -\frac{1}{6}$		$b_5 = \frac{11}{20}$		$b_7 = -\frac{11}{7}$	
			$b_9 = \frac{77}{12}$	$b_{10} = -\frac{176}{15}$	$b_{11} = \frac{21}{2}$	$b_{12} = -\frac{16}{3}$	$b_{13} = \frac{77}{52}$	$b_{14} = -\frac{16}{91}$		
	$q > 1$	B_2	$b_0 = \frac{16}{1365}$							

Table A.6: A set of B_j coefficients used in calculating I_P for a set of 3D Wendland kernel functions.

Kernel name	Range of q	Non-zero a_i-s							
Wendland C^2	$q \leq 1$		$a_0 = B_1$	$a_{-2} = \frac{1}{6}$		$a_{-4} = -\frac{1}{2}$	$a_{-5} = \frac{2}{3}$	$a_{-6} = -\frac{5}{14}$	$a_{-7} = \frac{1}{14}$
	$q > 1$	$a_1 = -\frac{1}{42}$	$a_0 = B_2$						
Wendland C^4	$q \leq 1$		$a_0 = B_1$	$a_{-2} = \frac{1}{6}$		$a_{-4} = -\frac{7}{15}$		$a_{-6} = \frac{5}{3}$	$a_{-7} = -\frac{8}{3}$
		$a_{-8} = \frac{35}{18}$	$a_{-9} = -\frac{32}{45}$	$a_{-10} = \frac{7}{66}$					
	$q > 1$	$a_1 = -\frac{8}{495}$	$a_0 = B_2$						
Wendland C^6	$q \leq 1$		$a_0 = B_1$	$a_{-2} = \frac{1}{6}$		$a_{-4} = -\frac{11}{20}$		$a_{-6} = \frac{11}{7}$	
		$a_{-8} = -\frac{77}{12}$	$a_{-9} = \frac{176}{15}$	$a_{-10} = -\frac{21}{2}$	$a_{-11} = \frac{16}{3}$	$a_{-12} = -\frac{77}{52}$	$a_{-13} = \frac{16}{91}$		
	$q > 1$	$a_1 = -\frac{16}{1365}$	$a_0 = B_2$						

Table A.7: A set of I_P coefficients for a series of different 3D Wendland kernel functions.

Bibliography

Agertz, O., Moore, B., Stadel, J., Potter, D., Miniati, F., Read, J., Mayer, L., Gawryszczak, A., Kravtsov, A., Nordlund, Å., Pearce, F., Quilis, V., Rudd, D., Springel, V., Stone, J., Tasker, E., Teyssier, R., Wadsley, J., Walder, R.. Fundamental differences between SPH and grid methods. MNRAS 2007;380:963–978. doi:`10.1111/j.1365-2966.2007.12183.x`.

Appel, A.W.. An Efficient Program for Many-Body Simulation. SIAM Journal on Scientific and Statistical Computing 1985;6:85–103.

Aurenhammer, F.. Power diagrams: properties, algorithms and applications. SIAM J Comput 1987;16(1):78–96. doi:`10.1137/0216006`.

Aurenhammer, F., Edelsbrunner, H.. An optimal algorithm for constructing the weighted voronoi diagram in the plane. Pattern Recognition 1984;17(2):251 – 257. URL: `http://www.sciencedirect.com/science/article/pii/0031320384900645`. doi:`https://doi.org/10.1016/0031-3203(84)90064-5`.

Bastian, N., Covey, K.R., Meyer, M.R.. A Universal Stellar Initial Mass Function? A Critical Look at Variations. Annual Review of Astronomy and Astrophysics 2010;48:339–389. doi:`10.1146/annurev-astro-082708-101642`. `arXiv:1001.2965`.

Bate, M.R., Bonnell, I.A., Price, N.M.. Modelling accretion in protobinary systems. MNRAS 1995;277:362–376. doi:`10.1093/mnras/277.2.362`. `arXiv:astro-ph/9510149`.

Beerer, I.M., Koenig, X.P., Hora, J.L., Gutermuth, R.A., Bontemps, S., Megeath, S.T., Schneider, N., Motte, F., Carey, S., Simon, R., Keto, E., Smith, H.A., Allen, L.E., Fazio, G.G., Kraemer, K.E., Price, S., Mizuno, D., Adams, J., Hernndez, J., Lucas, P.W.. A spitzer view of star formation in the cygnus x north complex. The Astrophysical Journal 2010;720(1):679. URL: `http://stacks.iop.org/0004-637X/720/i=1/a=679`.

Binney, J., Tremaine, S.. Galactic dynamics, 1987.

Bisbas, T.G., Haworth, T.J., Williams, R.J.R., Mackey, J., Tremblin, P., Raga, A.C., Arthur, S.J., Baczynski, C., Dale, J.E., Frostholm, T., Geen, S., Haugbølle, T., Hubber, D., Iliev, I.T., Kuiper, R., Rosdahl, J., Sullivan, D., Walch, S., Wünsch, R.. STARBENCH: the D-type expansion of an H II region. MNRAS 2015;453:1324–1343. doi:`10.1093/mnras/stv1659`. `arXiv:1507.05621`.

Bisbas, T.G., Wünsch, R., Whitworth, A.P., Hubber, D.A., Walch, S.. Radiation-driven Implosion and Triggered Star Formation. ApJ 2011;736:142. doi:`10.1088/0004-637X/736/2/142`. `arXiv:1105.3727`.

Bonnell, I.A., Bate, M.R.. Accretion in stellar clusters and the collisional formation of massive stars. MNRAS 2002;336:659–669. doi:`10.1046/j.1365-8711.2002.05794.x`.

Bonnell, I.A., Smith, R.J., Clark, P.C., Bate, M.R.. The efficiency of star formation in clustered and distributed regions. MNRAS 2011;410:2339–2346. doi:`10.1111/j.1365-2966.2010.17603.x`. `arXiv:1009.1152`.

Bowyer, A.. Computing dirichlet tessellations*. The Computer Journal 1981;24(2):162. URL: `+http://dx.doi.org/10.1093/comjnl/24.2.162`. doi:`10.1093/comjnl/24.2.162`.

Brinch, C., Hogerheijde, M.R.. LIME - a flexible, non-LTE line excitation and radiation transfer method for millimeter and far-infrared wavelengths. A&A 2010;523:A25. doi:`10.1051/0004-6361/201015333`. `arXiv:1008.1492`.

Brookshaw, L.. Solving the Heat Diffusion Equation in SPH (Invited paper). Mem Soc Astron Ital 1994;65:1033.

Camps, P., Baes, M., Saftly, W.. Using 3D Voronoi grids in radiative transfer simulations. A&A 2013;560:A35. doi:`10.1051/0004-6361/201322281`. `arXiv:1310.1854`.

Cappellari, M., McDermid, R.M., Alatalo, K., Blitz, L., Bois, M., Bournaud, F., Bureau, M., Crocker, A.F., Davies, R.L., Davis, T.A., de Zeeuw, P.T., Duc, P.A., Emsellem, E., Khochfar, S., Krajnović, D., Kuntschner, H., Lablanche, P.Y., Morganti, R., Naab, T., Oosterloo, T., Sarzi, M., Scott, N., Serra, P., Weijmans, A.M., Young, L.M.. Systematic variation of the stellar initial mass function in early-type galaxies. Nature 2012;484:485–488. doi:10.1038/nature10972. arXiv:1202.3308.

Choudhuri, A.R.. The physics of fluids and plasmas : an introduction for astrophysicists /, 1998.

Choudhury, R., Mookerjea, B., Bhatt, H.C.. Triggered Star Formation and Young Stellar Population in Bright-rimmed Cloud SFO 38. ApJ 2010;717:1067–1083. doi:10.1088/0004-637X/717/2/1067. arXiv:1005.1841.

Clark, P.C., Glover, S.C.O., Klessen, R.S.. TreeCol: a novel approach to estimating column densities in astrophysical simulations. MNRAS 2012;420:745–756. doi:10.1111/j.1365-2966.2011.20087.x. arXiv:1109.3861.

Clarke, C., Carswell, B.. Principles of Astrophysical Fluid Dynamics, 2007.

Da Rio, N., Robberto, M., Hillenbrand, L.A., Henning, T., Stassun, K.G.. The Initial Mass Function of the Orion Nebula Cluster across the H-burning Limit. ApJ 2012;748:14. doi:10.1088/0004-637X/748/1/14. arXiv:1112.2711.

Dale, J.E., Bonnell, I.. Ionizing feedback from massive stars in massive clusters: fake bubbles and untriggered star formation. MNRAS 2011;414:321–328. doi:10.1111/j.1365-2966.2011.18392.x. arXiv:1103.1532.

Dale, J.E., Ercolano, B., Bonnell, I.A.. Ionizing feedback from massive stars in massive clusters - II. Disruption of bound clusters by photoionization. MNRAS 2012;424:377–392. doi:10.1111/j.1365-2966.2012.21205.x. arXiv:1205.0360.

Dale, J.E., Ercolano, B., Clarke, C.J.. A new algorithm for modelling photoionizing radiation in smoothed particle hydrodynamics. MNRAS 2007;382:1759–1767. doi:10.1111/j.1365-2966.2007.12486.x. arXiv:0705.3396.

Dale, J.E., Wünsch, R., Smith, R.J., Whitworth, A., Palouš, J.. The fragmentation of expanding shells - III. Oligarchic accretion and the mass spectrum of fragments. MNRAS 2011;411:2230–2240. doi:10.1111/j.1365-2966.2010.17844.x. arXiv:1010.2131.

Dehnen, W., Aly, H.. Improving convergence in smoothed particle hydrodynamics simulations without pairing instability. MNRAS 2012;425:1068–1082. doi:10.1111/j.1365-2966.2012.21439.x. arXiv:1204.2471.

Dirichlet, G.L.. Über die Reduktion der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. Journal für die reine und angewandte Mathematik 1850;40:209–227.

Duchêne, G., Kraus, A.. Stellar Multiplicity. Annual Review of Astronomy and Astrophysics 2013;51:269–310. doi:`10.1146/annurev-astro-081710-102602`. `arXiv:1303.3028`.

Duffell, P.C., MacFadyen, A.I.. TESS: A Relativistic Hydrodynamics Code on a Moving Voronoi Mesh. ApJS 2011;197:15. doi:`10.1088/0067-0049/197/2/15`. `arXiv:1104.3562`.

Elmegreen, B.G., Lada, C.J.. Sequential formation of subgroups in OB associations. ApJ 1977;214:725–741. doi:`10.1086/155302`.

Field, G.B., Goldsmith, D.W., Habing, H.J.. Cosmic-Ray Heating of the Interstellar Gas. ApJ 1969;155:L149. doi:`10.1086/180324`.

Forgan, D., Bonnell, I.A.. Clumpy shocks as the driver of velocity dispersion in molecular clouds: the effects of self-gravity and magnetic fields in prep.;.

Forgan, D., Rice, K.. Native synthetic imaging of smoothed particle hydrodynamics density fields using gridless Monte Carlo radiative transfer. MNRAS 2010;406:2549–2558. doi:`10.1111/j.1365-2966.2010.16842.x`. `arXiv:1004.2846`.

Foster, P.N., Chevalier, R.A.. Gravitational Collapse of an Isothermal Sphere. ApJ 1993;416:303. doi:`10.1086/173236`.

Fryxell, B., Olson, K., Ricker, P., Timmes, F.X., Zingale, M., Lamb, D.Q., MacNeice, P., Rosner, R., Truran, J.W., Tufo, H.. Flash: An adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes. The Astrophysical Journal Supplement Series 2000;131(1):273. URL: `http://stacks.iop.org/0067-0049/131/i=1/a=273`.

Gingold, R.A., Monaghan, J.J.. Smoothed particle hydrodynamics - Theory and application to non-spherical stars. MNRAS 1977;181:375–389. doi:`10.1093/mnras/181.3.375`.

Gradshteyn, I.S., Ryzhik, I.M.. Table of integrals, series, and products. Seventh ed. Elsevier/Academic Press, Amsterdam, 2007. Translated from the Russian, Translation edited and with a preface by Alan Jeffrey and Daniel Zwillinger.

Harries, T.J.. Radiation-hydrodynamical simulations of massive star formation using Monte Carlo radiative transfer - I. Algorithms and numerical methods. MNRAS 2015;448:3156–3166. doi:`10.1093/mnras/stv158`. `arXiv:1501.05754`.

Harries, T.J., Monnier, J.D., Symington, N.H., Kurosawa, R.. Three-dimensional dust radiative-transfer models: the Pinwheel Nebula of WR 104. MNRAS 2004;350:565–574. doi:`10.1111/j.1365-2966.2004.07668.x`. `arXiv:astro-ph/0401574`.

Heyer, M.H., Brunt, C.M.. The Universality of Turbulence in Galactic Molecular Clouds. ApJ 2004;615:L45–L48. doi:`10.1086/425978`.

Hillenbrand, L.A.. On the Stellar Population and Star-Forming History of the Orion Nebula Cluster. AJ 1997;113:1733–1768. doi:`10.1086/118389`.

Hollenbach, D.J., Yorke, H.W., Johnstone, D.. Disk Dispersal around Young Stars. Protostars and Planets IV 2000;:401.

Hopkins, P.F.. A general class of Lagrangian smoothed particle hydrodynamics methods and implications for fluid mixing problems. MNRAS 2013;428:2840–2856. doi:`10.1093/mnras/sts210`. `arXiv:1206.5006`.

Hosokawa, T., Inutsuka, S.i.. Dynamical Expansion of Ionization and Dissociation Front around a Massive Star. II. On the Generality of Triggered Star Formation. ApJ 2006;646:240–257. doi:`10.1086/504789`. `arXiv:astro-ph/0511165`.

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

Jeans, J.. Astronomy and Cosmogony. 2nd ed. Cambridge Library Collection - Astronomy. Cambridge University Press, 2009. doi:`10.1017/CBO9780511694363`.

Jeans, J.H.. The Stability of a Spherical Nebula. Philosophical Transactions of the Royal Society of London Series A 1902;199:1–53. doi:`10.1098/rsta.1902.0012`.

Joe, B.. Construction of three-dimensional delaunay triangulations using local transformations. Computer Aided Geometric Design 1991;8(2):123 – 142. URL: `http://www.sciencedirect.com/science/article/pii/016783969190038D`. doi:`https://doi.org/10.1016/0167-8396(91)90038-D`.

Kahn, F.D.. The acceleration of interstellar clouds. BAN 1954;12:187.

Kennicutt, R.C., Evans, N.J.. Star Formation in the Milky Way and Nearby Galaxies. ARAA 2012;50:531–608. doi:`10.1146/annurev-astro-081811-125610`. `arXiv:1204.3552`.

Kennicutt Jr., R.C.. The Global Schmidt Law in Star-forming Galaxies. ApJ 1998;498:541–552. doi:`10.1086/305588`. `arXiv:astro-ph/9712213`.

Kessel-Deynet, O., Burkert, A.. Radiation-driven implosion of molecular cloud cores. MNRAS 2003;338:545–554. doi:`10.1046/j.1365-8711.2003.05737.x`.

Koepferl, C.M., Robitaille, T.P., Dale, J.E., Biscani, F.. Insights from Synthetic Star-forming Regions: I. Reliable Mock Observations from SPH Simulations. ApJ Supplement - awaiting publishing 2016;`arXiv:1603.02270`.

Krumholz, M.R., Klein, R.I., McKee, C.F.. Radiation-hydrodynamic Simulations of the Formation of Orion-like Star Clusters. I. Implications for the Origin of the Initial Mass Function. ApJ 2011;740:74. doi:`10.1088/0004-637X/740/2/74`. `arXiv:1104.2038`.

Krumholz, M.R., Klein, R.I., McKee, C.F., Offner, S.S.R., Cunningham, A.J.. The Formation of Massive Star Systems by Accretion. Science 2009;323:754. doi:`10.1126/science.1165857`. `arXiv:0901.3157`.

Kuiper, R., Klahr, H., Dullemond, C., Kley, W., Henning, T.. Fast and accurate frequency-dependent radiation transport for hydrodynamics simulations in massive star formation. A&A 2010;511:A81. doi:`10.1051/0004-6361/200912355`. `arXiv:1001.3301`.

Kuiper, R., Klessen, R.S.. The reliability of approximate radiation transport methods for irradiated disk studies. A&A 2013;555:A7. doi:`10.1051/0004-6361/201321404`. `arXiv:1305.2197`.

Kurosawa, R., Hillier, D.J.. Tree-structured grid model of line and polarization variability from massive binaries. A&A 2001;379:336–346. doi:`10.1051/0004-6361:20011233`. `arXiv:astro-ph/0109355`.

Larson, R.B.. The Evolution of Protostars – Theory. FCP 1973;1:1–70.

Larson, R.B.. Turbulence and star formation in molecular clouds. MNRAS 1981;194:809–826. doi:`10.1093/mnras/194.4.809`.

Larson, R.B.. Cloud fragmentation and stellar masses. MNRAS 1985;214:379–398. doi:`10.1093/mnras/214.3.379`.

Larson, R.B.. The physics of star formation. Reports on Progress in Physics 2003;66:1651–1697. doi:`10.1088/0034-4885/66/10/R03`. `arXiv:astro-ph/0306595`.

Lawson, C.L.. Transforming triangulations. Discrete Mathematics 1972;3(4):365 – 372. URL: `http://www.sciencedirect.com/science/article/pii/0012365X72900933`. doi:`https://doi.org/10.1016/0012-365X(72)90093-3`.

Levermore, C.D., Pomraning, G.C.. A flux-limited diffusion theory. ApJ 1981;248:321–334. doi:`10.1086/159157`.

Lloyd, S.. Least squares quantization in pcm's 1982;28:129–136.

Lomax, O., Whitworth, A.P.. SPAMCART: a code for smoothed particle Monte Carlo radiative transfer. MNRAS 2016;461:3542–3551. doi:`10.1093/mnras/stw1568`. `arXiv:1603.01223`.

Low, C., Lynden-Bell, D.. The minimum Jeans mass or when fragmentation must stop. MNRAS 1976;176:367–390. doi:`10.1093/mnras/176.2.367`.

Lucas, W.E., Bonnell, I.A., Dale, J.E.. How does early stellar feedback alter the impact of later supernovae? in prep.;.

Lucy, L.B.. A numerical approach to the testing of the fission hypothesis. AJ 1977;82:1013–1024. doi:`10.1086/112164`.

MacLachlan, J.M., Bonnell, I.A., Wood, K., Dale, J.E.. Photoionising feedback and the star formation rates in galaxies. A&A 2015;573:A112. doi:`10.1051/0004-6361/201322250`. `arXiv:1412.6918`.

Matzner, C.D.. On the Role of Massive Stars in the Support and Destruction of Giant Molecular Clouds. ApJ 2002;566:302–314. doi:10.1086/338030.

McKee, C.F., Ostriker, J.P.. A theory of the interstellar medium: three components regulated by supernova explosions in an inhomogeneous substrate. ApJ 1977;218:148–169. doi:10.1086/155667.

McKee, C.F., van Buren, D., Lazareff, B.. Photoionized stellar wind bubbles in a cloudy medium. ApJ 1984;278:L115–L118. doi:10.1086/184237.

McMillan, P.J., Dehnen, W.. Initial conditions for disc galaxies. MNRAS 2007;378:541–550. doi:10.1111/j.1365-2966.2007.11753.x. arXiv:astro-ph/0703534.

Mellema, G., Arthur, S.J., Henney, W.J., Iliev, I.T., Shapiro, P.R.. Dynamical H II Region Evolution in Turbulent Molecular Clouds. ApJ 2006;647:397–403. doi:10.1086/505294. arXiv:astro-ph/0512554.

Min, M., Dullemond, C.P., Dominik, C., de Koter, A., Hovenier, J.W.. Radiative transfer in very optically thick circumstellar disks. A&A 2009;497:155–166. doi:10.1051/0004-6361/200811470. arXiv:0902.3092.

Mirtich, B.. Fast and accurate computation of polyhedral mass properties. J Graph Tools 1996;1(2):31–50. URL: http://dx.doi.org/10.1080/10867651.1996.10487458. doi:10.1080/10867651.1996.10487458.

Momose, R., Okumura, S.K., Koda, J., Sawada, T.. Star Formation Efficiency in the Barred Spiral Galaxy NGC 4303. ApJ 2010;721:383–394. doi:10.1088/0004-637X/721/1/383. arXiv:1011.4508.

Monaghan, J.J.. Smoothed particle hydrodynamics. ARAA 1992;30:543–574. doi:10.1146/annurev.aa.30.090192.002551.

Monaghan, J.J., Lattanzio, J.C.. A refined particle method for astrophysical problems. A&A 1985;149:135–143.

Mouschovias, T.C.. Magnetic Braking, Ambipolar Diffusion, Cloud Cores, and Star Formation: Natural Length Scales and Protostellar Masses. ApJ 1991;373:169. doi:10.1086/170035.

Narita, S., Hayashi, C., Miyama, S.M.. Characteristics of Collapse of Rotating Isothermal Clouds. Progress of Theoretical Physics 1984;72:1118–1136. doi:`10.1143/PTP.72.1118`.

Norman, M.L., Wilson, J.R., Barton, R.T.. A new calculation on rotating protostar collapse. ApJ 1980;239:968–981. doi:`10.1086/158185`.

O'dell, C.R., Wen, Z., Hu, X.. Discovery of New Objects in the Orion Nebula on HST Images: Shocks, Compact Sources, and Protoplanetary Disks. ApJ 1993;410:696. doi:`10.1086/172786`.

Osterbrock, D.E., Ferland, G.J.. Astrophysics of gaseous nebulae and active galactic nuclei, 2006.

Overzier, R.A., Heckman, T.M., Wang, J., Armus, L., Buat, V., Howell, J., Meurer, G., Seibert, M., Siana, B., Basu-Zych, A., Charlot, S., Gonçalves, T.S., Martin, D.C., Neill, J.D., Rich, R.M., Salim, S., Schiminovich, D.. Dust Attenuation in UV-selected Starbursts at High Redshift and Their Local Counterparts: Implications for the Cosmic Star Formation Rate Density. ApJ 2011;726:L7. doi:`10.1088/2041-8205/726/1/L7`. `arXiv:1011.6098`.

Pawlik, A.H., Schaye, J.. TRAPHIC - radiative transfer for smoothed particle hydrodynamics simulations. MNRAS 2008;389:651–677. doi:`10.1111/j.1365-2966.2008.13601.x`. `arXiv:0802.1715`.

Petkova, M.A., Laibe, G., Bonnell, I.A.. Fast and accurate Voronoi density gridding from Lagrangian hydrodynamics data. J Comp Phys 2018;353:300–315. doi:`10.1016/j.jcp.2017.10.024`. `arXiv:astro-ph/1710.07108`.

Preibisch, T., Balega, Y., Hofmann, K.H., Weigelt, G., Zinnecker, H.. Multiplicity of the massive stars in the Orion Nebula cluster. New Astronomy 1999;4:531–542. doi:`10.1016/S1384-1076(99)00042-1`.

Price, D.J.. splash: An Interactive Visualisation Tool for Smoothed Particle Hydrodynamics Simulations. PASA 2007;24:159–173. doi:`10.1071/AS07022`. `arXiv:0709.0832`.

Price, D.J.. Modelling discontinuities and Kelvin Helmholtz instabilities in SPH. Journal of Computational Physics 2008;227:10040–10057. doi:`10.1016/j.jcp.2008.08.011`. `arXiv:0709.2772`.

Price, D.J.. Smoothed particle hydrodynamics and magnetohydrodynamics. Journal of Computational Physics 2012;231:759–794. doi:`10.1016/j.jcp.2010.12.011`. `arXiv:1012.1885`.

Price, D.J., Monaghan, J.J.. An energy-conserving formalism for adaptive gravitational force softening in smoothed particle hydrodynamics and N-body codes. MNRAS 2007;374:1347–1358. doi:`10.1111/j.1365-2966.2006.11241.x`.

Price, D.J., Wurster, J., Nixon, C., Tricco, T.S., Toupin, S., Pettitt, A., Chan, C., Laibe, G., Glover, S., Dobbs, C., Nealon, R., Liptai, D., Worpel, H., Bonnerot, C., Dipierro, G., Ragusa, E., Federrath, C., Iaconi, R., Reichardt, T., Forgan, D., Hutchison, M., Constantino, T., Ayliffe, B., Mentiplay, D., Hirsh, K., Lodato, G.. Phantom: A smoothed particle hydrodynamics and magnetohydrodynamics code for astrophysics. PASA - in press 2017;`arXiv:1702.03930`.

Raga, A.C., Cantó, J., Rodríguez, L.F.. Analytic and numerical models for the expansion of a compact H II region. MNRAS 2012a;419:L39–L43. doi:`10.1111/j.1745-3933.2011.01173.x`.

Raga, A.C., Cantó, J., Rodríguez, L.F.. The universal time-evolution of an expanding HII region. RMxAA 2012b;48:149–157.

Ramón-Fox, F.G., Bonnell, I.A.. Galactic Scale Flows and the Triggering of Star Formation in Spiral Galaxies. In: The Interplay between Local and Global Processes in Galaxies,. 2016. p. 35.

Rees, M.J.. Opacity-limited hierarchical fragmentation and the masses of protostars. MNRAS 1976;176:483–486. doi:`10.1093/mnras/176.3.483`.

Ritzerveld, J., Icke, V.. Transport on adaptive random lattices. Phys Rev E 2006;74(2):026704. doi:`10.1103/PhysRevE.74.026704`. `arXiv:cond-mat/0605139`.

Robitaille, T.P.. HYPERION: an open-source parallelized three-dimensional dust continuum radiative transfer code. A&A 2011;536:A79. doi:`10.1051/0004-6361/201117150`. `arXiv:1112.1071`.

Rybicki, G.B., Lightman, A.P.. Radiative processes in astrophysics, 1979.

Rycroft, C.H.. Voro++: A three-dimensional voronoi cell library in c++. Chaos 2009;19:041111. doi:10.1063/1.3215722.

Salpeter, E.E.. The Luminosity Function and Stellar Evolution. ApJ 1955;121:161. doi:10.1086/145971.

Schmidt, M.. The Rate of Star Formation. ApJ 1959;129:243. doi:10.1086/146614.

Schoenberg, I.J.. Contributions to the problem of approximation of equidistant data by analytic functions. Part A. On the problem of smoothing or graduation. A first class of analytic approximation formulae. Quart Appl Math 1946;4:45–99. doi:https://doi.org/10.1090/qam/15914.

Spitzer, L.. Physical processes in the interstellar medium, 1978. doi:10.1002/9783527617722.

Springel, V.. E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. MNRAS 2010;401:791–851. doi:10.1111/j.1365-2966.2009.15715.x. arXiv:0901.4107.

Springel, V., Hernquist, L.. Cosmological smoothed particle hydrodynamics simulations: the entropy equation. MNRAS 2002;333:649–664. doi:10.1046/j.1365-8711.2002.05445.x.

Springel, V., Yoshida, N., White, S.D.M.. GADGET: a code for collisionless and gasdynamical cosmological simulations. New Astronomy 2001;6:79–117. doi:10.1016/S1384-1076(01)00042-2.

Steinacker, J., Bacmann, A., Henning, T.. Application of adaptive multi-frequency grids to three-dimensional astrophysical radiative transfer. Journal of Quantitative Spectroscopy & Radiative Transfer 2002;75:765–786. doi:10.1016/S0022-4073(02)00042-0.

Strömgren, B.. The Physical State of Interstellar Hydrogen. ApJ 1939;89:526. doi:10.1086/144074.

Struve, O.. A New Slit Spectrograph for Diffuse Galactic Nebulae. ApJ 1937;86:613. doi:10.1086/143889.

Struve, O., Elvey, C.T.. Observations Made with the Nebular Spectrograph of the Mcdonald Observatory. ApJ 1939;89:119. doi:10.1086/144024.

Struve, O., van Biesbroeck, G., Elvey, C.T.. The 150-Foot Nebular Spectrograph of the McDonald Observatory. ApJ 1938;87:559. doi:`10.1086/143946`.

Sugitani, K., Tamura, M., Nakajima, Y., Nagashima, C., Nagayama, T., Nakaya, H., Pickles, A.J., Nagata, T., Sato, S., Fukuda, N., Ogura, K.. Near-Infrared Study of M16: Star Formation in the Elephant Trunks. ApJ 2002;565:L25–L28. doi:`10.1086/339196`.

Teyssier, R.. Cosmological hydrodynamics with adaptive mesh refinement. A new high resolution code called RAMSES. A&A 2002;385:337–364. doi:`10.1051/0004-6361:20011817`. `arXiv:astro-ph/0111367`.

Thompson, M.A., Urquhart, J.S., Moore, T.J.T., Morgan, L.K.. The statistics of triggered star formation: an overdensity of massive young stellar objects around Spitzer bubbles. MNRAS 2012;421:408–418. doi:`10.1111/j.1365-2966.2011.20315.x`. `arXiv:1111.0972`.

Tohline, J.E.. Hydrodynamic collapse. FCP 1982;8:1–81.

van Leer, B.. Towards the ultimate conservative difference scheme. V - A second-order sequel to Godunov's method. Journal of Computational Physics 1979;32:101–136. doi:`10.1016/0021-9991(79)90145-1`.

Vandenbroucke, B., De Rijcke, S.. The moving mesh code SHADOWFAX. Astronomy and Computing 2016;16:109–130. doi:`10.1016/j.ascom.2016.05.001`. `arXiv:1605.03576`.

Vandenbroucke, B., Wood, K.. The Monte Carlo photoionization and moving-mesh radiation hydrodynamics code CMACIONIZE. Astronomy and Computing 2018;23:40–59. doi:`10.1016/j.ascom.2018.02.005`.

Voronoi, G.. Nouvelles applications des param?etres continus a la theorie des formes quadratiques, deuxieme ?emoire, recherches sur les parallelloedres primitifs. Journal für die reine und angewandte Mathematik 1908;134:198–287.

Walborn, N.R., Smith, N., Howarth, I.D., Vieira Kober, G., Gull, T.R., Morse, J.A.. Interstellar Absorption-Line Evidence for High-Velocity Expanding Structures in the Carina Nebula Foreground. Publications of the Astronomical Society of the Pacific 2007;119:156–169. doi:`10.1086/511756`.

Walch, S., Whitworth, A.P., Bisbas, T.G., Wünsch, R., Hubber, D.A.. Clumps and triggered star formation in ionized molecular clouds. MNRAS 2013;435:917–927. doi:10.1093/mnras/stt1115. arXiv:1306.4317.

Walch, S.K., Whitworth, A.P., Bisbas, T., Wünsch, R., Hubber, D.. Dispersal of molecular clouds by ionizing radiation. MNRAS 2012;427:625–636. doi:10.1111/j.1365-2966.2012.21767.x. arXiv:1206.6492.

Watson, D.F.. Computing the n-dimensional delaunay tessellation with application to voronoi polytopes*. The Computer Journal 1981;24(2):167. URL: +http://dx.doi.org/10.1093/comjnl/24.2.167. doi:10.1093/comjnl/24.2.167.

Wendland, H.. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Advances in Computational Mathematics 1995;4(1):389–396. URL: https://doi.org/10.1007/BF02123482. doi:10.1007/BF02123482.

Whitworth, A.. The erosion and dispersal of massive molecular clouds by young stars. MNRAS 1979;186:59–67. doi:10.1093/mnras/186.1.59.

Whitworth, A.P., Bhattal, A.S., Chapman, S.J., Disney, M.J., Turner, J.A.. Fragmentation of shocked interstellar gas layers. A&A 1994a;290:421–427.

Whitworth, A.P., Bhattal, A.S., Chapman, S.J., Disney, M.J., Turner, J.A.. The Preferential Formation of High-Mass Stars in Shocked Interstellar Gas Layers. MNRAS 1994b;268:291. doi:10.1093/mnras/268.1.291.

Wolfire, M.G., Cassinelli, J.P.. Conditions for the Formation of Massive Stars. ApJ 1987;319:850. doi:10.1086/165503.

Wünsch, R., Dale, J.E., Palouš, J., Whitworth, A.P.. The fragmentation of expanding shells - II. Thickness matters. MNRAS 2010;407:1963–1971. doi:10.1111/j.1365-2966.2010.17045.x. arXiv:1005.4399.

Yorke, H.W., Krügel, E.. The dynamical evolution of massive protostellar clouds. A&A 1977;54:183–194.

Yorke, H.W., Sonnhalter, C.. On the Formation of Massive Stars. ApJ 2002;569:846–862. doi:10.1086/339264.

Zhu, Q., Hernquist, L., Li, Y.. Numerical convergence in smoothed particle hydrodynamics. The Astrophysical Journal 2015;800(1):6. URL: `http://stacks.iop.org/0004-637X/800/i=1/a=6`. doi:`10.1088/0004-637X/800/1/6`.

Zinnecker, H., Yorke, H.W.. Toward Understanding Massive Star Formation. Annual Review of Astronomy and Astrophysics 2007;45:481–563. doi:`10.1146/annurev.astro.44.051905.092549`. `arXiv:0707.1279`.