Superbubble Feedback in RAMSES

Superbubble Feedback in RAMSES

By Dennis Gallant, B.Sc.

A Thesis Submitted to the School of Graduate Studies in the Partial Fulfilment of the Requirements for the Degree Master of Science

McMaster University

Master of Science (2020)

Hamilton, Ontario (Department of Physics and Astronomy)

TITLE: Superbubble Feedback in RAMSES

AUTHOR: Dennis Gallant, B.Sc. (Saint Mary's University)

SUPERVISOR: Dr. James Wadsley

NUMBER OF PAGES: ix, 75

Abstract

We present a new model for stellar feedback in the hydrodynamics code RAMSES. This model, called the superbubble model, represents many temporally and spatially-clustered supernova events combining to form a single large bubble. It avoids the overcooling problem by splitting the gas into two distinct phases with their own temperatures and densities. It features conduction and a sub-grid implementation of evaporation in order to facilitate mass-loading of the bubble's interior. We find that this model produces bubbles 40% larger, and preserves four times as much of the supernovae energy, compared to the default RAMSES case. These results are similar to those obtained in earlier work in the Gasoline code by Keller et al (2014), and demonstrate that the superbubble model is an effective way of generating the feedback necessary to simulate galaxies without relying on any free parameters or ad hoc adjustments.

Contents

List of Figures

List of Acronyms

AGN active galactic nucleus

AMR adaptive mesh refinement

ISM interstellar medium

SN supernova

SPH smoothed particle hydrodynamics

UV ultraviolet

Chapter 1

Introduction

In the Great Debate (Shapley and Curtis 1921), astronomers first first argued about the existence of galaxies beyond our own Milky Way. Since then, it has become common knowledge that our galaxy is merely one of many such objects in the observable Universe, with recent estimates putting the total number of galaxies at up to two trillion (Conselice et al. 2016). Although we can see many other galaxies of all shapes, sizes, and ages with modern telescopes, there are still many questions that remain unanswered about their history and formation. In most scientific fields, theories can be tested in a lab experimentally, but the same is often not true of astrophysics. We can't build a star in a lab, let alone an entire galaxy. Furthermore, the timescales on which galaxies evolve are so long that no scientist could ever watch it occur naturally in their lifetime. Thus, we turn to computer simulations for answers.

In a computer, we can simulate physics we believe to apply to galaxies. We can allow those physics to play out over the course of billions of simulated years, all within a few days to months in real time. The results can then be compared to what observers see through their telescopes, allowing us to see if our understanding of the physics of galaxies is correct. One aspect of galaxy formation that has long challenged theorists to model is feedback; the ways in which stars and other energy sources heat up and influence their environment.

1.1 Galaxy Formation and the role of Feedback

Galaxy formation is a complex and challenging problem. In a recent review, Naab and Ostriker (2017) describe the history of the field and the theoretical difficulties in modelling galaxies. We know from observations of our own galaxy as well as high-redshift data that most galaxies formed billions of years ago, when the Universe was young. These early structures formed from gas coalescing into gravitationally-bound structures in the presence of massive dark matter halos. Not all of the gas that formed these early galaxies collapsed into stars, however. Instead, this gas is heated, dispersed, and sometimes outright ejected from the galaxy by feedback from various energy sources within the galaxy. These energy sources heat up gas and drive outflows from the galactic disk, which has the effect of regulating the star formation rate. This explains why we still see star formation today in many galaxies, instead of all the Universe's stars forming at once at a very early epoch.

Feedback can come in many forms. We are specifically interested in feedback that can regulate star formation, whether that be through turbulence or pressure. On a galactic scale, there are processes such as accretion from the galactic halo, shearing forces, and other interactions with the extragalactic environment. Some galaxies have active galactic nuclei and jets that can drastically change the environment. On smaller scales, we have the effects of stellar feedback, which gives structure to the interstellar medium (ISM). Figure 1.1, reproduced from Grond et al. (2019), plots the luminosity per solar mass of a typical stellar population. Most of the energy comes from UV radiation released over many years. However, it is the mechanical stellar feedback that we are interested in with this work: stellar winds and supernova, capable of outputting enormous amounts of energy into the galaxy very suddenly, and in a very small region of space as well.

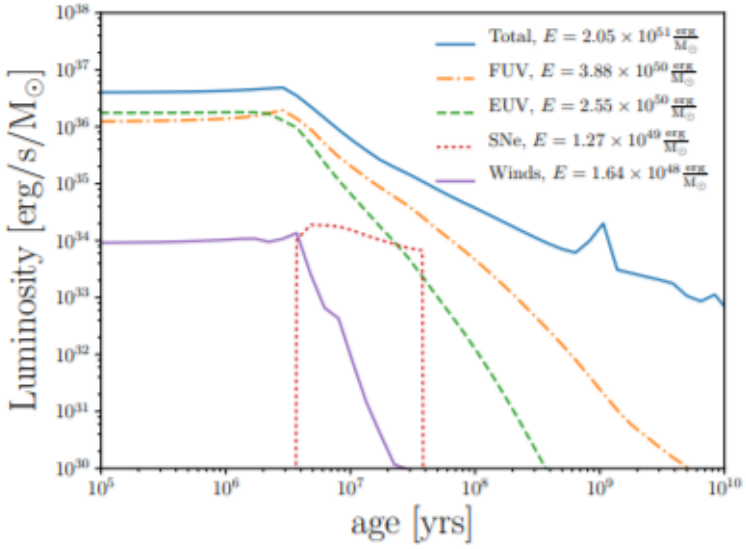

Figure 1.1: Luminosity per solar mass as a function of time, from Grond et al (2019). This feedback comes from a stellar population that follows the initial mass function from Chabrier (2003).

This rapid and powerful energy burst makes supernovae a crucial part of understanding galaxy evolution, but also makes them very difficult to model properly.

Any sensible feedback model must be able to regulate star formation and influence the structure of the ISM. In the absence of any opposing process, all of the ISM gas would rapidly collapse into star-forming clumps, turning all of the gas in the galaxy into stars orders of magnitude faster than the Gyr+ timescales inferred for typical spirals (Kennicutt 1998). Since our own galaxy is billions of years old and still has a healthy mix of stars and gas, we know that this can not be the truth. While accretion of new gas from outside the galaxy is important, it alone can not sustain star formation at observed rates without feedback (Ho et al. 2019). On the other hand, too much feedback would blow all of the gas out of the galaxy, leaving it barren and starved of star-forming potential. Due to the wide range of scales on which feedback occurs and the

limits of simulation resolution, we often have to make simplifying assumptions in order to save computing time. Astronomers have long struggled to come up with affordable stellar feedback models that result in the correct amount of star formation while also accurately and honestly representing physics as we understand it. Many models that accomplish both of these goals do so at the cost of including free parameters that remove predictive power by fine-tuning answers to achieve the desired result.

1.2 History of SN Feedback in simulations

An early example of a stellar feedback model can be found in Katz (1992). This model simply dumped the energy from feedback into the galaxy in the form of thermal energy. This barely caused an effect in the galaxy at large as the energy rapidly cooled by radiating away. This presents the first major obstacle for feedback models, namely the tendency for losing large amount of energy via cooling. Later authors such as Dalla Vecchia and Schaye (2012) point out that real stellar feedback can produce outflows of very high temperature ($> 10^6$ K) which would avoid this problem since cooling is very weak at that temperature. However, most simple feedback models have no way of naturally achieving and sustaining such a high temperature.

Other early attempts tried inputting the feedback energy as purely kinetic (Navarro and White, 1993), which avoids the cooling problem and therefore has a much stronger effect on the galaxy. However, this is not physically motivated, but rather a way of coping with the problem by working around it. It also tends to produce fluid velocities that are arbitrary and resolution-dependant. Durier and Dalla Vecchia (2012) later showed that there is physical promise to this method if the shock waves of feedback bubbles could be properly resolved, but

that is a challenge even with modern resolutions on galactic scales.

Various other workarounds for the overcooling issue have been attempted. Many have suggested shutting off cooling on recently-heated gas (Thacker and Couchman 2000, Sommer-Larson et al 2003, Stinson et al 2006). This would essentially equate to pretending the gas in our simulation is as hot as we know it should be. This can also be accomplished by separating the gas itself into two phases, one for newly introduced material and one for the cold, dense ISM (Springel and Hernquist 2003, Marri and White 2003). Even still, in a summary of the field, Scannapieco et al. (2012) concluded that no current techniques for stellar feedback could adequately reproduce observed galaxies. While some methods could regulate the stellar mass of the galaxy effectively, they often did so at the expense of the shape and angular momentum, evacuating the galactic disc of gas entirely. All of these techniques were also highly resolution dependant, with little numerical convergence. Finally, even the models that did produce good galaxies relied on assumptions or free parameters that strayed from first-principles physics. A similar conclusion was reached by Rosdahl et al. (2017) who test several different feedback methods in the RAMSES code and find that none of them can adequately regulate galaxies on their own. This implies either that our feedback models are lacking, or there are other processes at play that we are not accounting for. With the superbubble model, we hope to show that feedback alone can be the answer.

1.3 Superbubbles

In real galaxies, stellar feedback doesn't come from singular, isolated supernova events, nor does it come from a steady output of stellar winds spread evenly throughout the galaxy. Rather, stellar feedback from young stars is highly

correlated in both time and space, as young stars output almost all of the luminosity. Young stars exist in small clusters and usually live for less than 10 Myr, meaning that feedback occurs in short bursts of enormous strength as many young stars burn out at nearly the same time. The combined energies from their feedback form bubbles of hot, very sparse material in the interstellar medium, which are called superbubbles. The first observation of a superbubble may have been made by Menon (1962), in a radio survey of the Rosette Nebula. Menon noted a pronounced dip in the density of ionized gas in the centre of the nebula. Figure 1.2, originally from that paper, shows a contour plot of 10cm observations of the nebula. Later authors such as Mathews (1966) suggested that this could be the result of a high-temperature stellar wind blowing out the ISM. In 1975, Castor et al. proposed a mathematical model for such events, creating the model that would come to be known as the superbubble.

Superbubbles consist of huge regions of very hot gas bounded by a cooler, dense shell as the expanding bubble collides with the ISM. This structure is illustrated in Figure 1.3. These bubbles can exceed a kiloparsec in diameter, sweeping out huge portions of the ISM and maintaining high temperatures for millions of years. They offer a natural process through which feedback can heat the ISM to a point where it won't cool or form stars, therefore regulating the star formation rate. Despite this, they are not usually powerful enough to accelerate the gas beyond the galaxy's escape velocity, which prevents them from outright removing the gas in the galactic disk. Instead, superbubbles observed in nature tend to create chimney-like structures in the ISM, venting gas into the halo (Norman, 1989). The extent of a superbubble will depend on the initial mass function of the cluster powering it. This function could determine the energy injection rate, the total amount of mass ejected, and the duration of the

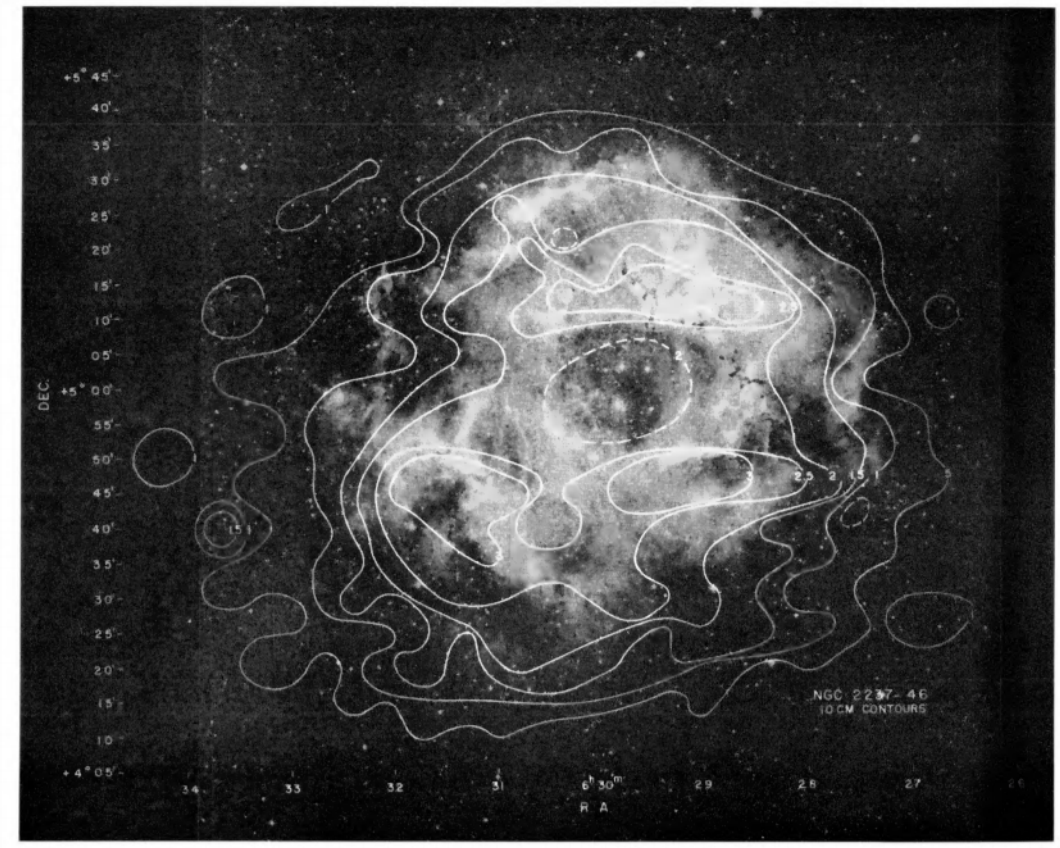

Figure 1.2: Contour plot of 10cm observations of the Rosette Nebula, originally from Menon (1962). A large cavity is seen in the centre. This hole would later be understood to be a superbubble.

energy injection as well as how constant it is (Leitherer 1999).

While early observations of superbubbles were made with radio telescopes, today they are best detected through their X-ray emission. Papers such as Cash et al. (1980) detect a superbubble in Cygnus through the X-rays emitted by their very hot gas. Emission lines from excited hydrogen near the shell/bubble interface can also be useful. Gerald et al. (2002) map out correlations between X-ray emission and Hα lines in NGC 3079 to trace the structure of a superbubble there. With modern tools like the HST and the Chandra X-ray observatory, detecting superbubbles in nature is easier than ever. This makes the ability to accurately model them in a way that generates observational predictions very valuable.

Superbubbles seem to be the obvious missing piece in the question of feedback. Why then has it taken so long for simulators studying galaxy evolution to start including them in their models? The main reasons have to do with the physics involved. Thermal conduction plays a crucial role in building superbubbles, transferring mass inward across the shell and towards the interior of the bubble. This has the result of heating up much more mass over time compared to the non-conductive case at the same resolution. Lower-resolution siumlations without conduction can still heat more mass than higher-resolution runs with it, however. The effects of conduction have been common knowledge among the stellar feedback community for decades (see the discussion of Weaver 1977 below). Galaxy cluster simulations must also consider conduction, as the gas in clusters easily reaches the temperatures where conduction is important. On the level of individual galaxies, this is not the case unless superbubbles are considered. This makes it easier to ignore for galaxy simulators. Even once

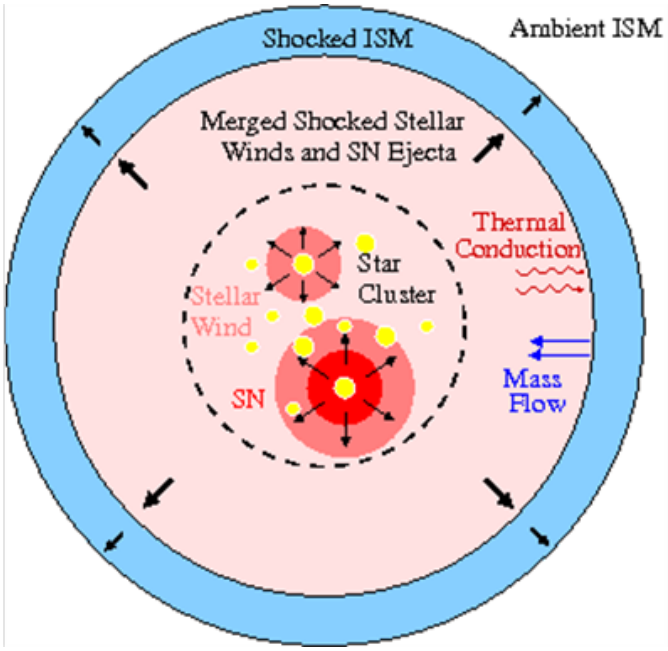

Figure 1.3: Illustration of superbubble structure. A small inner region produces a constant outflow of energy, which drives a shock into the ISM. The interior of the bubble is a large, hot, very sparse region. Conduction draws mass across the shell and into the bubble. For our purposes, the inner region is small enough that it can be treated as a point source.

the importance of conduction was understood, simulators were still hesitant to include it due to its computational intensiveness. Even today, after several years of attempts to simulate superbubbles, there are still lingering questions as to why they don't exactly match observations (El Badry et al. 2019), implying that there is still more physics missing from our modern models, and that our inability to fully resolve these bubbles will remain a challenge going forward.

Weaver et al. (1977) takes the basic model outlined by Castor et al. (1975) and expands it with a more detailed solution. Their paper calculates analytically the results of a strong stellar wind (such as that caused by a large number of closely clustered young stars) impacting the ISM. They include thermal conduction between the hot interior and the cooler, dense shell, but neglect cooling in their model, instead modifying their answer after the fact to account for radiative losses. They arrive at a self-similar solution, indicating that a continuous injection of energy from a small region of space can support an expanding spherical bubble that sustains itself indefinitely as long as the power source survives. Their solution shows that the bubble's radius grows as a power law in time. Finally, their solution predicts that 65% of the energy from the supernovae is retained in the bubble, of which three-tenths is kinetic and seven-tenths is thermal. The remaining 35% of the energy is lost due to radiative cooling as the bubble's expanding shock heats the surrounding ISM, which then cools to form the thin shell. This model assumes that the medium the bubble is expanding into is uniform. Kompaneets (1960) performs a similar calculation for an inhomogeneous medium and find that the bubble takes on an hourglass-like shape as it expands faster in less dense directions (such as out of the disk). Later, Vishniac (1983) analyses the stability of spherical blast waves and finds them overstable against small gravitational or dynamic perturbations. This means

that even small perturbations will grow to dramatically impact the system as a whole, giving it a clumpier, fragmented form.

Keller et al. (2014) successfully simulated superbubbles using conduction and a two-phase model of the ISM to avoid overcooling and physically regulate the hot mass without use of any free parameters. They apply this model to both a Milky Way analog and a dwarf galaxy and show that the superbubbles regulate star formation and produce significant outflows. In Keller et al. (2015), the model is applied to a cosmological galaxy, showing that outflows driven by superbubbles can produce the features we observe both at high redshift and the present day, without the need for any additional feedback mechanisms. In a follow-up paper, Keller et al. (2016) caution that for galaxies with virial masses $10^{12} M_{\odot}$ and above, supernova feedback alone is insufficient. Galaxies such as these are likely regulated at least in part by AGN feedback.

Recent works, such as El-Badry et al. (2019) have taken Weaver's classic bubble solutions and compared them to simulations at very high (sub-parsec) resolutions. El-Badry includes the cooling that Weaver neglected, which occurs in a tiny region near the shell as cold mass moves inward through evaporation. They then update Weaver's analytical solutions to include this cooling, arriving at new equations that describe the temporal evolution of superbubbles. Figure 1.4, originally from their paper, demonstrates this additional cooling. The temperature gradient at the shell/ISM boundary is not actually perfectly sharp, and there will exist some gas in the middle between the hot interior and the cold shell. It is here that there exists the potential for additional radiative cooling. It is unclear how applicable these findings are to other simulations, for a couple of reasons. First, the resolution of their fiducial runs is 0.024 pc, an incredibly

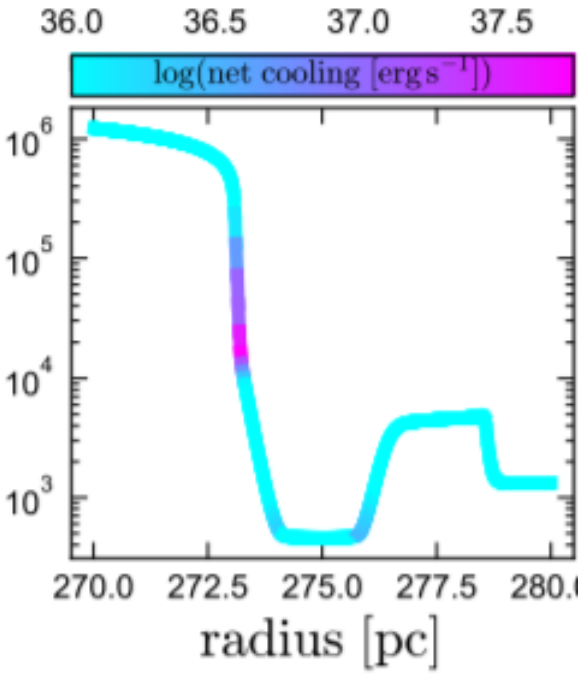

Figure 1.4: Temperature profile of a superbubble shock with extremely high resolution, from El-Badry et al. (2019). Highlighted is the region where rapid cooling occurs. As cold gas from the shell is heated by the interior, there will exist some amount of gas at a middle temperature which is prone to cooling. This results in additional energy loss compared to the Weaver (1977) solution, as Weaver assumed that the gas does not spend enough time in this temperature range to cool significantly.

low number compared to the current state-of-the-art galaxy simulations with resolutions of tens of parsecs. Second, in order to achieve this resolution, their simulations were performed in 1-d, resulting in perfect spherical symmetry. This prevents the formation of clumps or any physical mixing of material near the bubble boundary. They approximate this mixing by including artificial diffusion. This makes it unclear exactly how reliable or predictive their results might be.

1.4 Thesis goals and outline

Keller et al. (2014) presented a superbubble model for galaxy feedback in the particle-based code Gasoline. Much of the work in this thesis will be an extension of that work, attempting to apply it to a grid-based code, RAMSES. To simulate a superbubble successfully, Keller had to make several major additions to the standard hydrodynamic code:

- **Conduction:** Thermal Conduction is necessary for mass loading of the superbubble. It is the key piece of additional physics not found in most galaxy simulations.
- **Multiphase Model:** In order to avoid the common overcooling problem, the fluid is split into two phases with distinct temperatures and densities, in a pressure equillibrium with one another. This allows us to model a sub-resolution region of space where hot and cold gas interact without averaging out their temperatures, avoiding the rapid cooling temperature range seen in older simulations.
- **Evaporation:** The conductive mass flux of cold ISM material into the hot interior takes place on length scales far below our resolution limits. It must be modeled at a sub-grid level, where the conditions near the edge of the shell are used to convert some of the cold phase into hot in our multiphase model.

There are also specific challenges to implementing these changes in a grid code, discussed in Section 2.1. By successfully implementing a superbubble in RAMSES, a very different code from Gasoline, we aim to demonstrate the diverse applicability of this model.

Chapter 2 of this thesis will introduce RAMSES, the code used for this project. We will discuss the differences between grid and particle codes, the history of RAMSES and its capabilities, and the useful additions to it that already existed prior to this project. We will discuss the technical details behind how and why one would use RAMSES, and then explain the ways we worked with and modified the code to prepare for the superbubble package. Chapter 3 will focus on the new physics added to RAMSES by us to create the superbubble, including the two-phase model and evaporation. Chapter 4 will discuss the results. Finally, Chapter 5 will present our conclusions and future applications of this work.

Chapter 2

Method: Ramses code

RAMSES is a widely used and publicly available astrophysical hydrodynamics grid code introduced in Teyssier (2002). It was designed for the study of structure formation on large scales, and is also well-suited to galaxy formation and evolution. The main reason we chose RAMSES was that others had already incorporated thermal conduction into the code. RAMSES is a grid-based code that includes Adaptive Mesh Refinement (AMR), allowing for the resolution of individual cells to be recursively refined. This makes it ideal for studying systems with a wide range of important length scales. RAMSES uses a second-order Godunov method to solve its hydrodynamics. This means that it follows the method of Godunov (1959), who described the first exact Riemann solver for the Euler equations of hydrodynamics (A Riemann problem is an initial-value problem dealing with a conservation equation with a discontinuity). The Godunov scheme describes the boundaries between each cell in a numerical grid as individual Riemann problems, and solves them using a conservative finite-volume method. In addition to the hydrodynamics solver, RAMSES also includes an N-body solver for tracking dark matter particles and stars. Finally, it features cosmological capabilities, but these will not be used for this work.

In this chapter, we start by discussing the differences between grid and particle codes. We will then discuss the capabilities of RAMSES as it is found in its publicly available form, as well as how to use it. Next the additions made to the code by Dubois and Commerçon (2015) to enable conduction will be discussed. Finally, we will cover the modifications made to RAMSES not

related to the superbubble code, and discuss our general workflow, method, and goals for modifying the code.

2.1 Particle vs. grid codes

In the field of astrophysical hydrodynamics simulations, there are two primary types of code: particle-based, and grid-based (see Dale 2015 for a review). In particle-based codes, fluids are modeled using discrete mass elements. To evolve the fluid from one step to the next, the force on each mass element is calculated, and then they are moved around accordingly. A type of particle code used often in astrophysics is Smoothed Particle Hydrodynamics (SPH) codes, meaning that the masses of the particles are smoothed out over a certain spherical volume that contains a number of neighbouring particles. This smoothing is how fluid properties such as density and gradients are calculated. Some advantages of particle-based codes are:

- Since the particles are free to move anywhere, there is no need to set strict boundary conditions for the simulation.
- Gravitational calculations are easy to perform using tree code methods.
- It is possible for areas within the simulation to be truly empty, unlike in a grid code where each cell must have a defined value for each variable.
- Tracing fluid flows is extremely easy, since each particle is a unique object with a known history. This is very relevant when discussing solutions to the overcooling problem below.

Grid-based codes take a different approach. In this case, the simulation consists of a grid of volume elements, each with their own density, pressure, and so on. These elements fill a simulation volume defined ahead of time, which

must have boundary conditions set. The simulation proceeds by calculating the forces on the fluid in each cell, and moving the appropriate amount of material from each cell to its neighbours. Many grid codes, such as RAMSES, employ Adaptive Mesh Refinement to split or merge cells in the course of the simulation in response to ever-changing resolution requirements. Some advantages of using a grid code include:

- Refining the resolution is as simple as splitting some cells apart, which is much less disruptive than introducing smaller particles into an SPH code.
- Elements are defined by volume, not mass, which is useful for systems with a wide range of masses or densities as well as for keeping resolution high even in regions where density is low.
- Grid cells are regularly spaced, rather than the disorderly spread of particles. This regularity allows the use of higher-order approximation methods for solving equations (SPH is second order).
- Small scale mass flows and mixing are much easier, since mass is not discretized like in particle codes. In particle codes such as Gasoline (Wadsley et al. 2017), a mixing term is added to compensate for this deficiency, even for general (non-superbubble) cases.

However, these advantages come with some hefty downsides. Since each volume element must contain at least something, computing time can often be wasted on cells where nothing should be happening. Another issue is the existence of preferred directions. In a particle code, particles can move in any direction. But in a grid code, each cell has only six adjacent cells, forcing material to flow along the cardinal directions. This causes fluids to advect slightly more efficiently along the primary axis of the grid. This can lead to structures forming that are more rectangular than they should be. The final and most

noticeable problem is the existence of nonphysical diffusion. Grid codes can introduce additional diffusion through purely numerical effects, smearing the fluid out across neighbouring cells. This problem gets worse as the speed of the material increases. Particle codes, on the other hand, tend to have the opposite problem. Since they are limited by mass, not volume, particle codes struggle with very small mass flows.

One example of a situation where grid codes can struggle is the delayed cooling feedback method (Teyssier 2013), a model used by RAMSES which will be discussed further in the next section. These models attempt to solve the overcooling problem by temporarily turning off cooling when feedback occurs. In a particle-based code, this is trivial; simply flag the particles that represent the new, hot material, and turn cooling off for those particles. On a grid, however, it's much harder to track where the feedback energy is going from step to step. Passive scalars, discussed later in this chapter, can be used to trace fluid flow and will be crucial in our handling of the overcooling problem. However, due to the aforementioned diffusion present in grid codes, these tracers can get smeared into the wrong cells, falsely flagging them. It will be important to keep this effect in mind when building our two-phase model, as well as when discussing our simulation results in Chapter 4.

2.2 Generic Ramses

RAMSES is written in the Fortran language, widely used in physics due to its speed and suitability for scientific computing. The publicly available build of RAMSES features a powerful hydrodynamic solver, as well as options for magnetohydrodynamics and cosmology, though only the hydrodynamics solver was used in this project. RAMSES does not feature thermal conduction by default.

The current default stellar feedback method in RAMSES is described in Teyssier et al. (2013). Each star particle (formed when gas reaches a user-defined threshold) represents many thousands of solar masses clustered closely in space. After a specified amount of time (usually chosen to be approximately 10 Myr), the particle loses a (user-defined) portion of its mass, representing the large stars exploding as supernovae. When this happens, the supernova energy is deposited into the cell with the star particle. This energy is entirely thermal, except for whatever kinetic energy is required for conservation purposes as the particle loses mass. This energy is dumped into the star particle's cell, and the code evolves it normally from there. The exploded stars can also increase the metallicity of the cell accordingly. After this, the surviving mass in the star particle is inert save for its gravitational effects (each only explodes once, as a discrete event). If the `delayed_cooling` option is chosen, then RAMSES will add a passive scalar along with the feedback energy to trace its flow. Cells with this tracer scalar have cooling switched off. This gives the energy time to influence its surroundings before being immediately cooled away. This tracer's initial mass is proportional to the feedback ejecta mass, and decays according to $dM/dt = -M/\tau$, where τ is a user-defined parameter. Once the passive scalar's mass is sufficiently lower than the ordinary mass in the cell, cooling is allowed to turn back on for that cell. This option was not used in any of our final runs, as the superbubble model achieves the same results naturally. In Section 2.4, we will outline the changes we made to this feedback model to enable the superbubble.

While this delayed cooling method is the default, it is not the only method that has been attempted in RAMSES. Rosdahl et al. (2017) tested five differ-

ent sub-grid models for supernovae feedback in RAMSES galaxy simulations. These methods were basic thermal energy injection, kinetic feedback, delayed cooling, mechanical feedback (where feedback is deposited as either energy or momentum on a resolution-dependant basis), and a method of stochastic thermal energy injection. They conclude that none of these methods can fully model observed galaxies.

RAMSES is run by first compiling the code using a Makefile and then feeding it a namelist at runtime. The Makefile contains major settings unlikely to change from run to run, such as the number of dimensions and which external libraries are in use. The namelist contains user-defined variables that will change frequently, such as the initial conditions of the simulation and what physics is enabled. The output of a run is a set of binary files, one each recording the state of the grid, hydrodynamic variables, and particles. For this project, these files were read and analyzed using the python package `yt` (Turk et al. 2011). yt is an open-source analysis toolkit designed to work across several common codes to handle the challenges involved in analyzing complex data sets and to enable collaboration between users of multiple codes.

2.3 Ramses with Conduction

Thermal conduction is an essential part of the superbubble model, but is not a part of RAMSES by default. Fortunately, Dubois and Commerçon (2015) have implemented an implicit scheme for solving conduction in RAMSES. We would like to thank them for allowing us to use their code for this project.

Conduction increases the computing time enormously. An early attempt by

us to solve the heat transfer equation explicitly failed due to the impractically small timesteps required. This is why Dubois and Commerçon (2015) favour an implicit solver. Even with such a solver, our runs with conduction ran almost an order of magnitude slower compared to those without. This slowdown was especially impactful in our highest resolution runs, as the rate of conduction depends on the temperature gradient, which is much steeper at higher resolution. This longer computational time is a fair price to pay to enable the superbubble model, though it is understandable why it has taken so long for galaxy simulators to embrace conduction.

When Keller et al. (2014) implemented superbubbles into Gasoline, they had to introduce artificial mass flows to boost the effects of conduction. This is because Gasoline is a particle code, which are unable to advect mass between simulation elements because the masses are the simulation elements. Instead the mass particles themselves move, but this makes capturing small mass flows difficult. This leads to particle codes underestimate conductive mass flows. Fixing this required a sub-grid model to boost the effects of conduction, a model which our code will not require. On the other hand, it is possible that in a grid code, numerical diffusion could have the opposite effect, overdoing conduction.

2.4 Modification to Ramses

While preparing to implement the superbubble code, we made many changes to RAMSES to help with debugging as well as adding several useful features. These changes were tracked using a github repository, allowing us to keep a record of every change made over time. RAMSES is comprised of many different subroutines, each handling a different part of the code. The parts of the code

most relevant to our work are:

- `amr_step.f90`: The main driver of the code. Calls all of the routines listed below in order, as well as many other functions not relevant to this discussion.
- `newdt_fine.f90`: The routine that computes the timestep.
- `godunov_fine.f90`: The hydrodynamics solver. Calculates the forces on each fluid element and updates the values in each grid.
- `cooling_fine.f90`: The routine where cooling occurs.
- `feedback.f90`: This routine handles stellar feedback, as described above.
- `conduction.f90`: Added by Dubois and Commerçon (2015), this is where the conductive heat transfer is calculated and applied.

This order in our version of `amr_step` is different than in the default version of RAMSES. In that version, the feedback is applied just before the hydro step. We moved it to later in the process so that the timestep would always be determined in between the injection of feedback energy and the hydrodynamic calculations. This lets the timestep account for the large amount of new material from feedback, whereas in the original code the entire feedback-hydro-cooling process would operate on a single timestep calculated before the code knows how much feedback is coming. In the large-scale galaxy simulations RAMSES was designed for, this problem would rarely come up, as there would essentially always be some region of space demanding a low timestep. In our single-star-particle tests, however, it was common for nothing to be happening anywhere in the simulation box until feedback was switched on, which would be a disaster if we continued to use the very large timestep calculated before feedback occurred.

Figure 2.1 demonstrates the effect of calculating the timestep after feedback. The default timesteps (left panel) are larger than the cooling time, and so all of the energy from a feedback event cools away before it can impact the hydrodynamics if we let the code choose a timestep before feedback. By calculating the timestep after feedback (right panel), the code can see the new energy and cool it gradually instead.

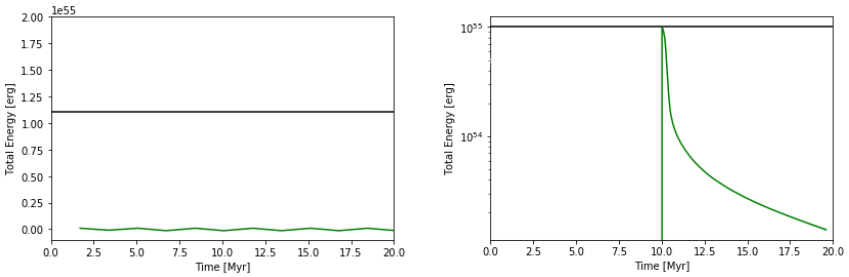

Figure 2.1: Demonstration of default RAMSES cooling of a singular feedback event (left) compared to the same event cooled after the timestep calculation was moved to occur after the feedback step (right). The horizontal line indicates the total energy of the supernovae. By default, the timesteps chosen before the explosion are larger than the cooling time, so the explosion is never even seen. Calculating the timestep later allows for the energy to cool gradually as it should.

In `newdt_fine.f90`, the timestep is calculated. By default, RAMSES selects its timesteps based on the state of the hydrodynamics on a given refinement level, with higher refinement levels having proportionally smaller timesteps. This was fine for our purposes, but we needed more control over what exactly the timestep was. RAMSES does not allow the user to manually choose the timestep, or change the rules for how the timestep is calculated. We added a namelist option that allows the user to define a maximum timestep which the code will never exceed. This was used for testing purposes, allowing us to test our superbubble code for convergence at small dt.

We did not make any significant changes to the way `godunov_fine.f90` operates, but did add some additional code related to our two-phase treatment of the ISM, discussed in Chapter 3.

`cooling_fine` is where a majority of the superbubble code is found, again discussed in Chapter 3. We added code allowing the cooling to be switched on and off independently of the hydro solver (for testing purposes) and most importantly switched from RAMSES' built-in chemical cooling package to `Grackle 3`. Grackle (Smith et al. 2016) is an open-source cooling library extracted from the cooling code of Enzo (Bryan et al. 2014), another grid-based astrophysical hydrodynamics code. Grackle allows for non-equilibrium cooling of typical ISM species as well as several UV background models for heating. In Figure 2.2, we show the cooling curve used for our simulations. For all densities, cooling is at its highest in the 10^4-10^6K temperature range. At lower temperatures, net cooling is actually negative (dashed lines) due to background photoelectric heating. The v-shaped dips where the dashed lines meet the solid ones are the equilibrium temperatures for that density.

In `feedback.f90,` we added a new method of stellar feedback: gradual energy injection. By default, all of the feedback from a given star particle occurred at once. This is analogous to a single star exploding in a supernova. However, star particles rarely represent only a single star. Instantaneous feedback is therefore a poor approximation. Releasing all of the energy at once does allow one to achieve very high gas temperatures, but it is not physically motivated. It is also highly sensitive to the simulation timestep, often requiring options like the previously mentioned `delayed_cooling` to work properly. A better method of

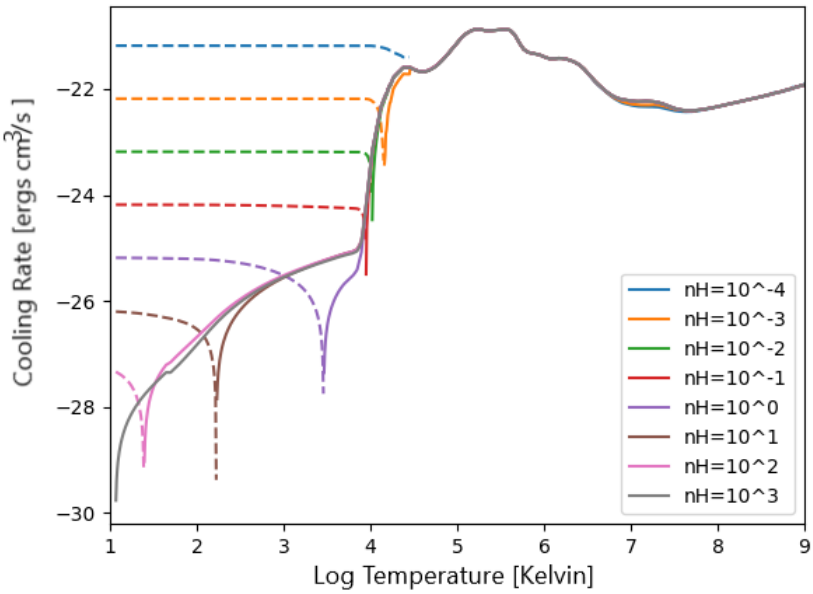

Figure 2.2: Cooling curve used for our simulations, plotted for a variety of densities with a solar metallicity. Cooling rate peaks in the middle of the temperature range, whereas both very hot and cold gas cool more slowly. Dashed lines represent heating (negative cooling) caused by a photoelectric background. The dip where the dashed lines meet the solid ones are equilibrium temperatures.

feedback is to take the energy that the star was slated to explode with, and spread that energy injection out across its entire lifetime. This accounts for both the continuous feedback from winds, and for the existence of a closely-clustered population of stars with slightly different ages and death dates, which is what a star particle really represents. Referring back to Figure 1.1, we can see that the energy output from young stellar populations is spread out across the first several million years of their lives, not released all at once. When the newly-added namelist option `gradual_sn` is switched on, this model is used. Each step, each star particle checks its initial birth mass, its current mass, and the portion of its life it has remaining, and converts an appropriate amount of its mass into thermal (and kinetic, if moving) feedback such that it emits energy at a constant rate throughout a 10 Myr feedback period. After that, it becomes inert, just as in the default method. This continuous energy injection is crucial for building superbubbles, but it is also a more accurate depiction of stellar feedback in general. Figure 2.3 compares gradual and instantaneous feedback, showing how we arrive at the same amount of total energy emitted by the end of the particle's 10 Myr feedback period, but at a constant rate.

As will be seen in Chapter 3, creating a superbubble required the addition of new variables tracking the properties of the hot and cold phases. RAMSES allows for the introduction of additional variables beyond the standard density, velocity, and internal energy, but these variables are treated as passive scalars by default. A passive scalar is something present in a fluid that has no direct effect on the dynamics of that fluid. It is simply advected around passively, going with the flow of the fluid. This can make them analogous to tracer particles, albeit messier. Adding these new variables will not change the dynamics of the existing ones at all other than whatever new code explicitly refers to the

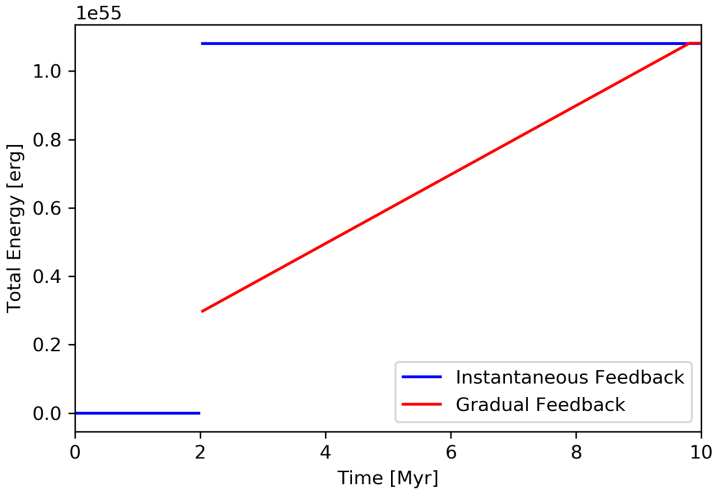

Figure 2.3: Comparison of feedback methods for a single star particle producing feedback without cooling. By default, all of the energy a star particle will ever output is deposited into the simulation all at once (blue line). In our simulations, we modify this so that the star particle emits gradually at a constant rate over its 10 Myr feedback period (red line). In this case, the first timestep was larger than the rest, and so the particle emits the correct amount of energy to "catch up" to where it should be.

new variable, making them a safe way of implementing new features to the code.

Other changes to the code were mostly minor tweaks for testing or debugging purposes, the most significant of which was an alteration to `init_part.f90`, the routine that reads in the initial conditions for the particles. By default, RAMSES can only have dark matter particles at the beginning of a run, with stars forming afterwards wherever gas is concentrated. We enabled the IC reader to recognize individual stars, allowing us to test small-scale scenarios involving one or more particles exploding.

Chapter 3

Building superbubbles in RAMSES

After familiarizing ourselves with RAMSES and adding the new features discussed in Chapter 2, we began work on implementing the physics required to form superbubbles. This included a sub-grid multiphase model for the ISM and for evaporation. This chapter will detail the specific implementation of these features, as well as some results demonstrating their function.

3.1 Two-Phase Model

In a grid-based code such as RAMSES, each cell has only one value for density, temperature, and any other physical variable the code is tracking. This presents a problem when trying to model something like the ISM, which is known to be comprised of multiple phases (such as in the three-component model of McKee and Ostriker 1977). For example, when a supernova releases material into the simulation, that material should be extremely hot. But there is also cold, dense material in that grid cell, from before the explosion. By default, Ramses would simply treat both of these phases as one fluid, with a single averaged temperature. This is the most frequent cause of the overcooling problem discussed in Section 1.2, since it results in too much material in the 10^4-10^6 K range. As seen in Figure 2.2, this is where typical ISM cooling curves are strongest. To avoid this, we implement a two-phase model where the gas is split into a "hot" and "cold" phase explicitly on a sub-grid level. The cold material will naturally sit at an equilibrium around 10^4 K, while the hot material will typically be well above 10^6 K, where it rarely cools. The two phases are assumed to be in pressure equilibrium with each other, as in Keller et al. (2014). This model

is distinct from previous two-phase attempts such as Agertz et al. (2013) in that the different phases have specific, separate physical qualities as opposed to simply being treated differently by processes such as cooling. The model is most crucial in the early moments of feedback where the hot bubble has not yet grown large enough to resolve.

The superbubble two-phase model required the addition of four new fluid variables: ρ_{hot} and ρ_{cold}, denoting the densities of the two phases, as well as $\rho\epsilon_{hot}$ and $\rho\epsilon_{cold}$, denoting the density times the thermal energy of each phase. In principle only the two new hot variables would be needed, with the cold phase simply being total - hot. Originally we built and tested such a model, with only the hot phase tracked explicitly. For reasons described below, we eventually abandoned that model and decided to track the cold explicitly as well. These new variables are passive scalars, as described in Section 2.4. By default, RAMSES will advect them around with the rest of the fluid, but they have no effect on the other variables other than what we add to the code. The RAMSES variables for total density and energy were not changed, and will continue behaving just as they did before these changes by default. At the beginning of the simulation, all ISM material is "cold". ρ_{cold} and $\rho\epsilon_{cold}$ are set to be equal to the total ρ and $\rho\epsilon$. Any material added later on by stellar feedback is considered "hot", with its own density and energy. This addition of new matter increases both total ρ, $\rho\epsilon$ as well as ρ_{hot}, $\rho\epsilon_{hot}$ by the same amount. The result is that each cell has three values of ρ and $\rho\epsilon$, where the hot and cold values add up to exactly equal the totals.

While these passive scalars will follow the rest of the fluid via advection, there is still other physics going on that we need to include them in. The first

is what is known as PdV work, applied during the hydrodynamics calculations (see Landau and Lifshitz 1959). An expanding or compressing gas causes the energy of a system to change according to the first law of thermodynamics:

$$work = PdV, \tag{1}$$

where P and dV are pressure and the change in volume, respectively. Each step, the code must calculate the local rate of change of volume for each cell, and adjust the energy of that cell accordingly. By default, this does not apply to passive scalars, and so we must include them ourselves. This was accomplished by calculating the fraction of the energy stored in the hot phase before PdV work is applied to the total energy, e_{frac}:

$$e_{frac} = \rho\epsilon_{hot}/(\rho\epsilon_{hot} + \rho\epsilon_{cold}). \tag{2}$$

Then, after the PdV step, the new $\rho\epsilon_{hot}$ is simply the new total $\rho\epsilon$ times e_{frac}. Similarly, the new cold energy is just the new total times $(1 - e_{frac})$. The amount of PdV work done on each phase is thus proportional to its share of the total energy, as in Keller et al. (2014). This method has the benefit of ensuring that $e_{hot} + e_{cold}$ always equals the total energy explicitly.

Previously, we attempted a model in which only two new variables were used, ρ_{hot} and $\rho\epsilon_{hot}$. In this model, the values for the cold phase would just be calculated as the total density or internal energy minus the hot density or energy whenever they were needed, rather than being stored explicitly. This would allow us to accomplish the same results with two fewer variables, reducing memory consumption and perhaps improving run time slightly. In this case, the fraction of energy stored in the hot phase before PdV work is:

$$e_{frac} = \rho\epsilon_{hot}/(\rho\epsilon_{total}). \tag{3}$$

Unfortunately, this method ran into an issue where this fraction would occasionally become greater than 1. This was caused by slight differences in how the passive scalars and the other variables are advected. Over time, these small changes added up until the sum energies of the two phases were significantly out of alignment with the total energy. By creating another passive scalar to store the cold phases' properties explicitly and calculating e_{frac} using that, this problem is avoided at the cost of requiring an additional couple of variables. Figure 3.1 compares these two methods. There are qualitative differences in the share of the energy between the two phases, even though these two methods should in theory produce identical results. This is because of the advection errors causing the sum of hot and cold energy to not always equal the total energy within a given cell (though globally there is no energy mismatch, the problem arises within individual cells). The difference is on the order of 10% at most, with the new method having higher values of $\rho\epsilon_{hot}$.

Other than PdV work, the other physical process that must be accounted for is cooling. When the cooling step begins, RAMSES takes the average temperature of the cell and uses that to compute the cooling in that cell. We instead calculate the temperature of each phase separately using our new variables, and compute two cooling values, one for each phase. These temperature changes alter the energy of the hot and cold phase appropriately, which then changes the total energy. This change to total energy due to cooling is the only way that the two-phase model interacts with the hydrodynamics calculations driving the RAMSES code, but it is an absolutely crucial interaction. By splitting up the temperatures and cooling them seperately, we avoid the overcooling problem

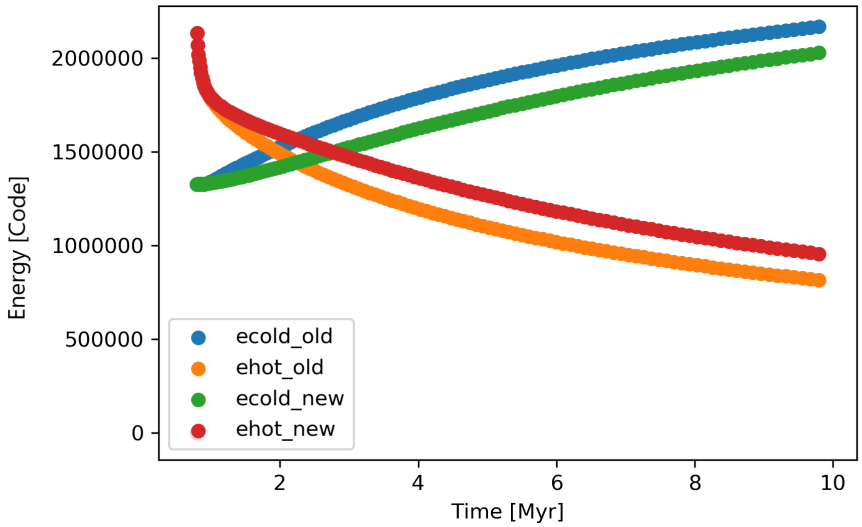

Figure 3.1: Energies of the two phases over time for a single, instantaneous explosion of a star particle, without cooling, for both e_{frac} models. As the hot bubble expands, hot material is shocked against the ISM to drive the expansion. This causes a gradual loss of hot energy (and corresponding increase in cold, as the total energy is conserved). In the old case, only the hot phase is tracked explicitly, and the hot fraction used to update the energy of the phases is calculated compared to the total energy before PdV work. The cold phase is calculated using the difference between the total and the hot. In the new case, the cold phase is tracked explicitly as well, and the hot fraction is calculated compared to the sum of the hot and cold energies. This changes the results slightly. In the old case, advection errors caused the sum of hot and cold to drift away from the total energy on a local cell level, leading to a loss of about 10% of the hot energy as hot energy values were being updated with incorrect e_{frac}.

elegantly.

Once the hot phase has been sufficiently cooled, it is no longer useful to continue tracking it separately from the cold phase. At this point, we convert it back into cold to both save on computing time by returning to one phase and to avoid confusion as to whether or not the hot phase is actually physically hot. The threshold for this was set at 10^5K (as in Keller et al. 2014) since it is the temperature below which evaporation effectively ceases. This temperature is also in the rapid cooling range, and so will quickly merge with the cold phase.

This two-phase model allows us to avoid the overcooling problem by cooling each phase separately. Since each phase is typically outside of the rapid cooling regime, we cool very little compared to simply averaging their temperatures. In addition to this benefit, the hot phase also serves as a useful tracer for the effects of feedback. For example, in Figure 3.2 below, it is easy to see that the new material injected by feedback has dominated the central region of the bubble, by blowing away all of the original cold material into the shock. Tracking the hot phase like this would be trivial in a particle code, where it amounts to nothing more than tracking the history of a given particle backwards in time. But in a grid code, the introduction of a passive scalar is required.

3.2 Evaporation

In Weaver's model, conduction of energy from the hot bubble into the shell is balanced by evaporation of cool gas from the shell into the bubble. Crucially, this means that thermal energy does not leave the bubble via this process. Rather, mass enters the bubble through conduction and the bubble's temper-

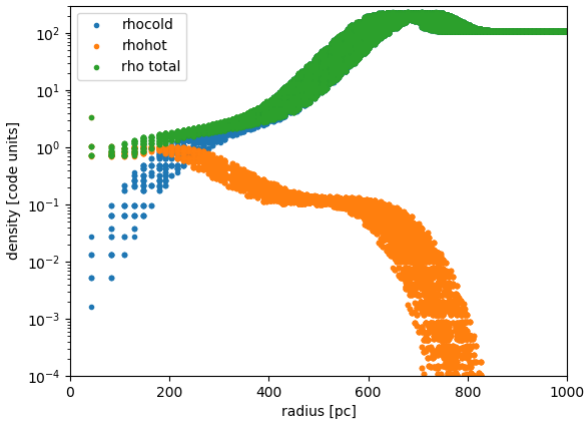

Figure 3.2: Demonstration of the two-phase model. A radial profile of the density of a typical shock wave produced by our superbubble model (see Chapter 4 for description of the initial conditions) is plotted at 10 Myr and at a resolution of 50 pc, but without cooling or evaporation. The densities of the two phases are shown along with the total density. The hot phase is negligible near the edge of the bubble, but dominates in the inner region where it has pushed out the cold phase.

ature lowers accordingly, reducing the rate of conduction over time. Weaver uses this balance to compute the rate of evaporation into the bubble. For this section, we will follow the El-Badry's (2019) treatment of the problem.

Conductive heat flux through a surface of radius r is -$4\pi r^2\kappa$(dT/dr), where dT/dr is the rate of change of temperature with radius and κ is the thermal conductivity as in Spitzer(1962). Following the math in El-Badry, we have for the evaporative mass flux:

$$dM/dt = \frac{-8\pi\mu m_p r^2 \kappa (dT/dr)}{5k_B T}. \tag{4}$$

If we assume no cooling in the bubble interior as well as the shell itself, then the temperature gradient can be approximated as $rdT/dr = (2/5)T_{hot}$, where

T_{hot} is the temperature of the hot bubble material at the shell/bubble interface. This derivation also assumes that the steep temperature gradient at that interface will remain unresolved, a safe assumption as resolving it properly requires sub-parsec resolution. We also write the conductivity as $\kappa = C{T_{hot}}^{5/2}$. Here, C is $1.84E-5$ divided by the Coulomb logarithm, which we take to be 40. This gives a final form for the evaporative mass flux of:

$$dM/dt = \frac{16\pi\mu m_p r C {T_{hot}}^{5/2}}{25 k_B}. \tag{5}$$

This evaporation should occur naturally via conduction, but in any galaxy-scale simulation, our resolution won't be nearly high enough for this to occur at the expected rate. This means we will have to implement evporation on a sub-grid level, using the approximations noted above. Our two-phase model is critical to this, as it allows us to model the hot/cold interface at sub-grid resolution.

The sub-grid model uses the equation above to calculate the amount of evaporation that should be occurring in any given two-phase cell. It then converts the appropriate amount of cold matter into the hot phase by increasing ρ_{hot} while lowering ρ_{cold} by the same amount. $\rho\epsilon_{hot}$ and $\rho\epsilon_{cold}$ are modified in the same way. This has the net effect of reducing the temperature of the hot phase as it increases in density until the two phases are equalized. It is worth noting that not all evaporation takes place on a sub-grid level. It can also occur between two adjacent cells. However, there is no need to model this explicitly as RAMSES can already handle it.

There are a couple potential issues with this approach. The first is that it

assumes no cooling in the interface, which was shown to occur by El- Badry (albeit in a restrictive 1-d model). Ignoring that cooling means that our results will represent an upper bound on the internal bubble temperature. El-Badry account for this cooling by modifying the equation for mass flux accordingly:

$$dM/dt \approx \frac{10\mu m_p(1-\theta)dE/dt}{33k_B T_{hot}} \tag{6}$$

where dE/dt is the rate of energy injection and θ is the fraction of this energy lost to cooling in the shell region where cooling can occur. If this fraction were known, we could modify our evaporation code accordingly. This may be a useful avenue for future work, as evaporation is the part of our superbubble model where we have made the most simplifying assumptions.

The second problem with this model is a tactical one: How do we decide what the effective radius, r, of the bubble looks like? One simple solution is to assume a sphere, although in reality these superbubbles are never perfectly spherical. Additionally, each cell in the interface region could require different treatments depending on its orientation and location. For this work, we have made the simplifying solution that the hot phase in each cell assumes the shape a sphere with volume $V_{hot} = m_{hot}/\rho_{hot}$, similar to Keller et al. (2014). Since the hot phase dominates the volume of any cell where it is present, the radius of this sphere, and hence the rate of evaporation, will go as dx, the width of the cell. However, the volume of a cell goes as dx^3. This means that the effective evaporation rate will increase as the resolution is improved. This is intuitively odd, and means that evaporation rate will never converge with resolution. There is some physical merit to this result in the context of an unresolved bubble, where smaller cells would have smaller bubbles that take proportionally less time to

evaporate. Still, there is likely a more robust treatment of evaporation to be considered in future work. Such a treatment would need to abandon our assumption of a spherical hot phase, instead determining the shape of the surface across which evaporation is occurring on a cell-by-cell basis. This might involve looking at the shape of the bubble as a whole and using the current trajectory of the hot phase in a region to determine what its orientation should be.

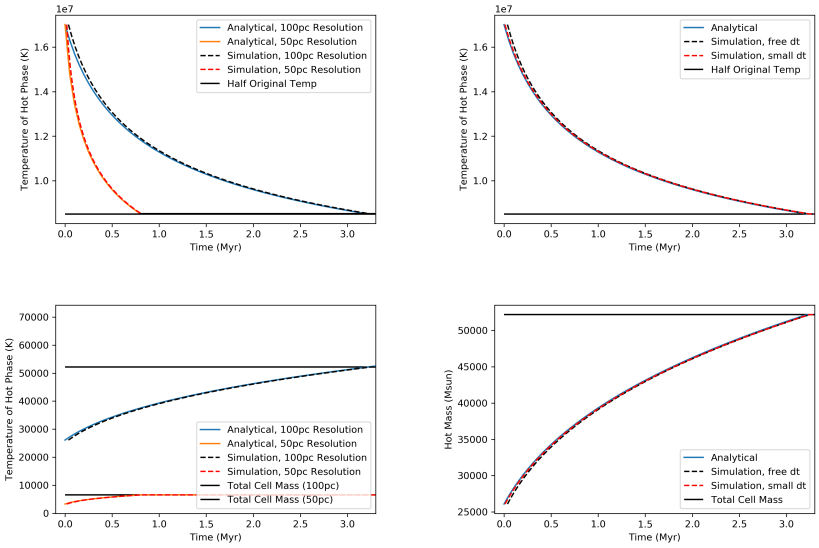

Figure 3.3: Demonstration of sub-grid evaporation model compared to analytical solution. Initial conditions consisted of a single grid cell with an even split between hot and cold phase masses in pressure equilibrium with each other. (Left panels) Temperature and mass of the hot phase over time at two different resolutions. Improving resolution by half increases the evaporation speed by a factor of four. Evaporation stops once the cold phase has been completely converted, when the hot phase has lost half its initial temperature and doubled its initial mass. (Right panels) Effects of varying timestep. As timestep decreases, we begin to converge on the analytical solution.

Figure 3.3 shows a demonstration of the evaporation model. This simulation consisted of a single grid cell with an even split between hot and cold phases by mass, in pressure equilibrium with one another. Cooling and hydrodynamics

were disabled, so the only physics occurring was evaporation. The hot phase proceeds to convert cold material into hot phase at the rate described in equation 5. This continues until all of the cold matter has been converted, which coincides with the hot material having been reduced to half its original temperature. When the resolution is increased by a factor of two, the evaporation rate drops by that factor, but the cell volume drops by a factor of eight. This has the net effect of increasing the effective evaporation rate by a factor of four. When the timestep is forced to be very small, as opposed to being chosen by RAMSES, the results begin to converge on the analytical solution determined by numerically integrating Equation (5). RAMSES chooses its main timestep based off of the hydrodynamics, meaning that processes like our evaporation code can be neglected when they would benefit from a smaller timestep. One avenue for future work would be to incorporate a way for evaporation to benefit from smaller timesteps without changing the overall hydro timestep, such as a substep model or an implicit approach.

Chapter 4

Results

This chapter will present the results of our superbubble code. First, we will discuss the initial conditions for our test scenario, and then compare slices, profiles and time evolution of our superbubble with the non-superbubble cases. We will then study the convergence of our three resolutions.

4.1 Initial Conditions

To test the superbubble code, a very simple test case was constructed. The initial conditions consisted of a single star particle at rest in the centre of a large box full of uniform gas. The gas had a density of 1 hydrogen atom per cc, an initial temperature of 4800 K, and solar metallicity. These conditions put it in equilibrium with a background photoelectric heating rate of 4^{26} erg/s per atom. The star particle had an initial mass of $10^6 M_\odot$. It began exploding immediately, releasing 10% of its mass as ejecta over the course of 10 Myr, with an energy injection rate of $3.17\text{x}10^{40}$ erg/s. This equates to the most massive 1% of the stellar population represented by the particle exploding as supernovae. The energy injection rate is the key parameter in this scenario, and is sensitive to the initial mass function of the star cluster. The one chosen here is a typical value, similar to the value listed for stellar winds in Figure 1.1. The actual mass of the star particle itself does not affect the final results much, however since we neglect self-gravity.

These conditions are intended to mimic a star cluster within a representative

patch of ISM. While in reality the cluster may be embedded within a cold molecular cloud at first (as in McKee and Ostriker 1977), we expect such a cloud to be quickly dispersed by UV and winds. Additionally, the scale of superbubbles is over a kiloparsec. A bubble so large will sample a representative patch of the ISM as it grows beyond whatever initial region of space it started in. The exception is bubbles that grow out of the disk, forming hourglass-like shapes (Kompaneets 1960). Testing scenarios with a less uniform ISM is a potential avenue for future work.

Three types of cases were simulated. The first are the no-superbubble cases. These are simply the default RAMSES physics, with no changes. The next case is the default RAMSES + conduction case. Finally, we have the superbubble case, which in addition to conduction includes the two-phase and evaporation models discussed in Chapter 3. Each case was run at three resolutions, with cell widths of 100, 50, and 25 parsecs. Resolutions of this fidelity are now standard in high-res galaxy simulations (see, for example, Kim et al. 2016). It is important that our superbubble model works at realistic resolutions so that it can be expanded to galaxy-scale tests. The total box width is 3.2 kiloparsecs, more than large enough to prevent the bubble from reaching the edges of the simulation space while the stars are still exploding.

4.2 Slices and Projections

We begin by presenting density slices and projection for each physical case. Figures 4.1 through 4.6 contains density slices and projections through the centre of the simulation box at 10 Myr, at our highest resolution. All display the familiar characteristics of blast waves, with a dense shockwave and a sparse

interior. The no-superbubble case (Figures 4.1 and 4.4) has a boxy shape to it. The conduction case (Figures 4.2 and 4.5) looks more spherical, though still has noticeably grid-like perturbations present. The superbubble case (Figures 4.3 and 4.6) is the most spherical, and more importantly is significantly (approximately 40%) larger than the non-superbubble cases. As we will see in the next section, this is a sign that we have successfully avoided the overcooling problem, retaining a large portion of the supernovae energy and heating up more mass. For now, let us turn our attention to the shapes of these bubbles.

As discussed in Section 2.1, grid codes have preferred directions, advecting more strongly along the primary axes of the grid. In the no-superbubble case, there is much less energy due to overcooling, and so the thermal energy driving bubble evolution is much weaker. This gives the numerical perturbations associated with the grid more time to grow compared to the slowly expanding bubble. A smaller bubble is also less well resolved, taking on a blockier appearance as fewer cells are available to model it. In the conductive case, the addition of conduction has provided a new physical method of moving material around, weakening the relative appearence of the underlying numerical perturbations. Even in the superbubble case, however, the effects of preferred directions can be seen in the low-density regions at the centre of the bubble. In Figure 4.6, an entire pattern of flower-shaped density perturbations can be seen. These perturbations are insignificant in the superbubble case, but are nonetheless an excellent demonstration of the effects numerical errors can create.

While these perturbations are certainly troubling, in reality superbubbles are hardly spherical. The real ISM is a patchwork of structures (McKee and Ostriker 1977) and these natural perturbations will results in a variety of shapes

for real superbubbles as they take the path of least resistance through a clumpy ISM. Even in a very uniform environment, Vishniac (1983) shows that superbubbles will grow small perturbations to the point where they cause the bubble to take on a more clumpy, fragmented shape. While the squarish artifacts in our runs are clearly caused by the grid-based code, real superbubbles will take on a variety of shapes for physical reasons. An important question to ask is whether these numerical perturbations are strong or weak compared to those caused by a real galactic environment. In the superbubble case, it is clear that the numerical perturbations have had little effect, and a non-uniform ISM would have a much greater impact on the final shape of the superbubble. In our non-superbubble cases, however, the perturbations are comparatively much stronger. In these cases, it is fair to say that the numerical effects have seriously impacted the result.

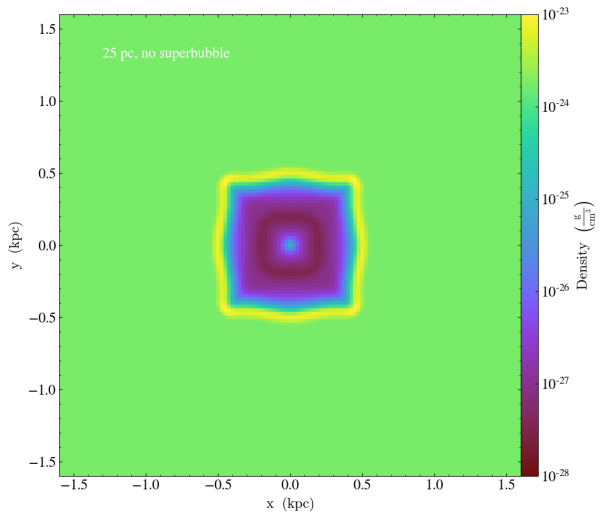

Figure 4.1: Density slice through the centre of the simulation box at 10 Myr for the non-superbubble case without conduction. The bubble has evolved along the preferred axes of the grid, giving it a box-like shape.

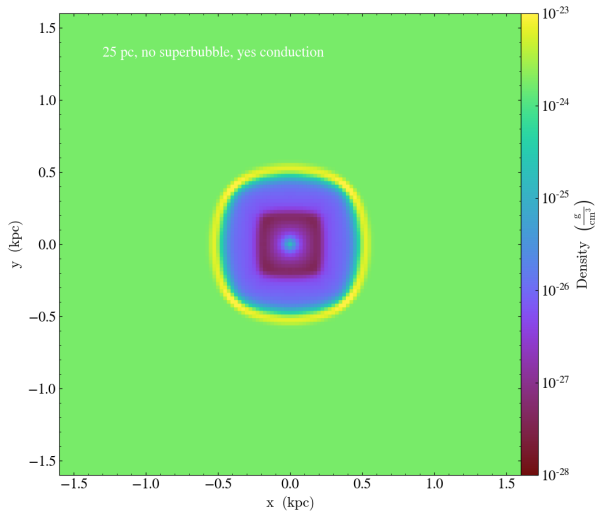

Figure 4.2: Density slice through the centre of the simulation box at 10 Myr for the non-superbubble case with conduction. The addition of conduction has provided additional physical methods of moving material around, reducing the effect of the numerical perturbations.

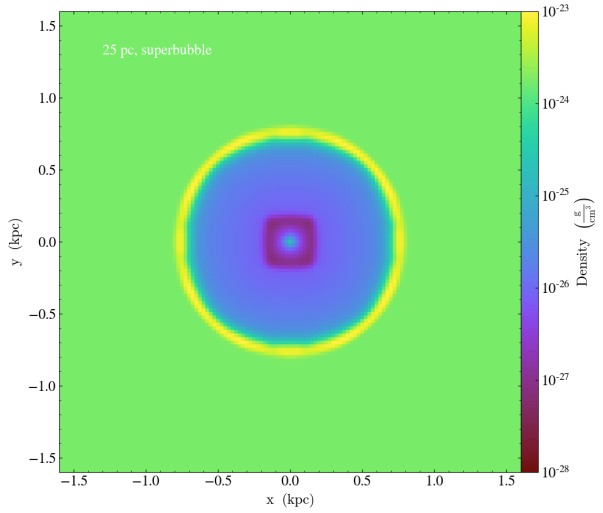

Figure 4.3: Density slice through the centre of the simulation box at 10 Myr for the superbubble case. The bubble is much large than in the previous cases, as it has retained more energy by avoiding overcooling. Grid-based perturbations are minimal.

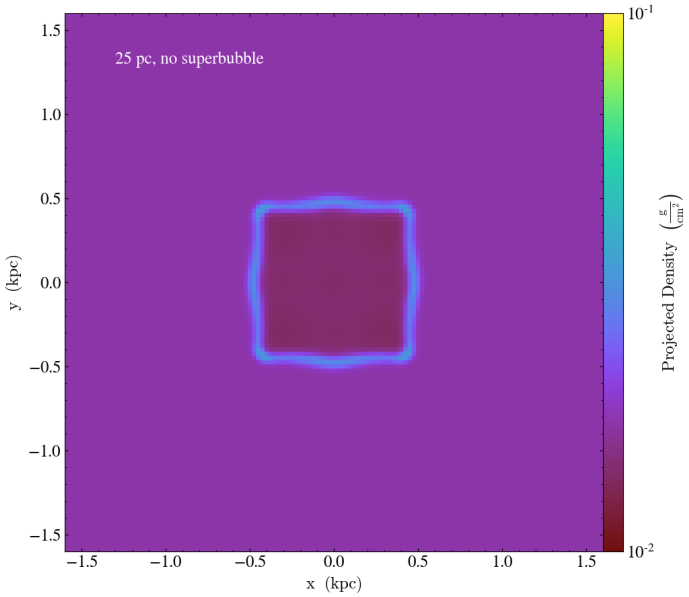

Figure 4.4: Density projection at 10 Myr for the non-superbubble case without conduction. The bubble has evolved along the preferred axes of the grid, giving it a box-like shape.

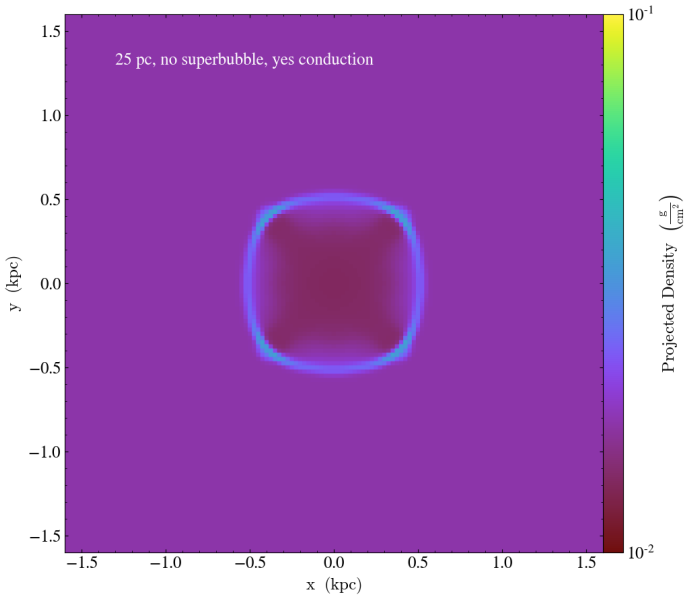

Figure 4.5: Density projection at 10 Myr for the non-superbubble case with conduction. An X-shaped numerical artifact can be seen inside the bubble.

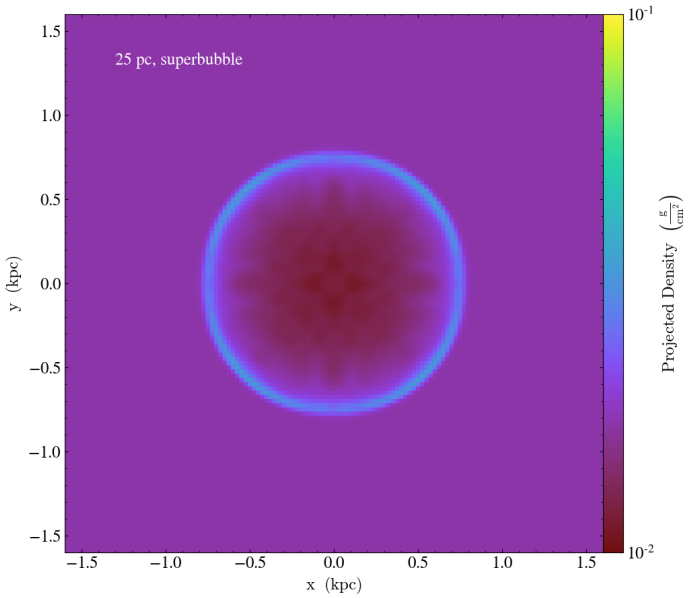

Figure 4.6: Density projection at 10 Myr for the superbubble case. An flower-shaped perturbation has grown out from the centre of the box..

4.3 Time Evolution

Here we will plot the evolution of the bubble with time, looking at its radius, energy, and mass. It is important to clearly define what counts as "inside the bubble" to avoid confusion. As we will see, there is no easy answer for this due to numerical diffusion smearing out the thin cold shell. As illustrated in Figure 1.3, there are three relevant regions in a superbubble; the innermost region containing the energy source, the hot bubble, and the cold, dense shell. The inner region is simply our star particle, and is treated as a point in this simulation. This is in line with Weaver's assumption that the various supernovae from a star cluster would blend together such that they provide a constant spherically-symmetric thermal energy injection. The outer region, the cold shell, is extremely thin in reality. In fact, Weaver (1977) assumes it has no width at all. As can be seen in Figure 4.3, for example, in our simulations it can be smeared out across a few grid cells in either direction. Due to our shell having a finite width often extending to 2 or 3 grid cells, our highest resolutions runs could be underestimating the radius by 50-75 pc compared to a fully-resolved thin shell at sub-parsec resolutions. This underestimation would only get worse as resolution worsens, as we will see in Section 4.5. Finally, as the non-superbubble cases are less spherical, it may be less correct to call this value a "radius" in those cases. Instead, it represents the furthest extent of the bubble in any direction. This distance should be analogous to a sphere's radius anyway, only just along a few preferred directions rather than every direction.

We define the radius of the hot bubble as the distance from the centre of the simulation box where the temperature first rises above 10^6 K. For counting up the mass and energy within the bubble, we simply sum our passive scalars denoting the mass and energy of the hot phase. This will count hot material in

cells with overall temperatures below 10^6 K, which means we will count some material outside of our defined radius. This mainly impacts our low-resolution runs, which will be discussed in Section 4.5. While this disagreement is not desirable, the alternatives are excluding some of the hot material just because it is mixing with the smeared-out cold shell, or defining our radius as the furthest extent of the hot phase. While that second option may sound reasonable, due to numerical leakage there exists some hot phase even at large radii (see Section 4.4), making this method unworkable. We will simply have to proceed with the knowledge that no clean definition of the radius is available. The definition we settled on, one based on temperature, equates to choosing the radius where the hot phase begins to dominate. We tested several other temperatures in the 10^5-10^7 K range as our cutoff before settling on 10^6 K. Lower values lead to a larger radius that indicated we were including too much of the cold shell, while higher values were clearly excluding too much of the bubble when compared to the temperature profiles presented in Section 4.4.

Figure 4.7 compares the radius of the bubble over time in all three physical cases at a resolution of 25 pc. The radius is dramatically larger with our superbubble model, increasing by approximately 40% compared to the non-superbubble cases. This is due to the two-phase model preventing the overcooling problem.

In Figure 4.8 we plot the energy of the hot bubble over time as a fraction of the total feedback energy released up until that point in time. The overcooling problem is clearly evident in the non-superbubble cases, where only about 10% of the energy has survived at later times without being cooled away. With the superbubble model, that number is closer to 40%. This value is still less than

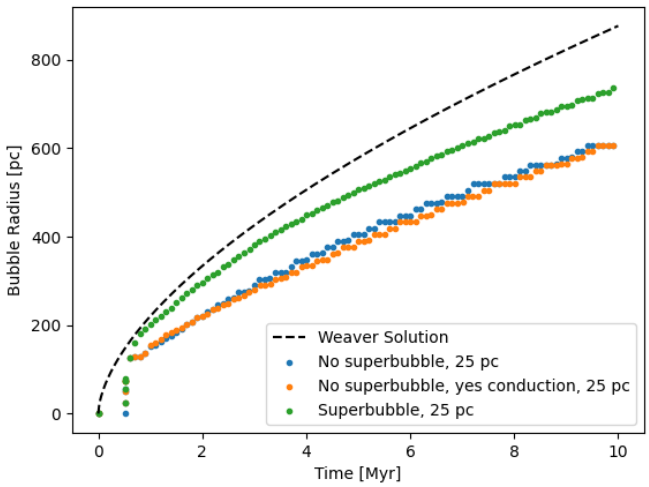

Figure 4.7: Time evolution of bubble radius, excluding the cold shell, for each of the three difference cases, compared to the analytical solution from Weaver at al. (1977). The radius is much larger in the superbubble case as excessive cooling is prevented, though it still falls short of the predicted value due to retaining less energy than Weaver predicts.

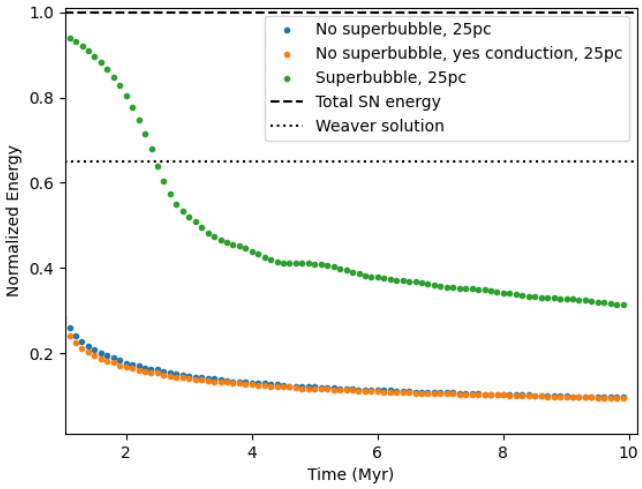

Figure 4.8: Time evolution of bubble energy, excluding the cold shell, for each of the three difference cases, compared to the analytical solution from Weaver at al. (1977). Energy is normalized against the total feedback energy released at a given time to illustrate Weaver's prediction that the bubble would assume a self-similar state where 65% of the energy is maintained by the bubble, while the rest is lost to cooling. This prediction is plotted as a dotted line. Our superbubble model retains only 40% of the energy, though this is still much more than the non-superbubble cases which loose almost everything to cooling. This energy loss could be explained as radiative cooling in the shell/bubble interface (El-Badry et al. 2019) or as a byproduct of our evaporation model.

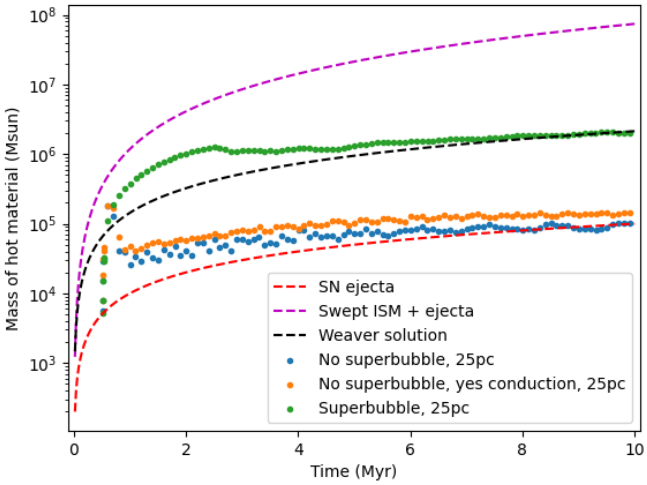

Figure 4.9: Time evolution of bubble mass, excluding the cold shell, for each of the three difference cases, compared to the analytical solution from Weaver at al. (1977) as well as an upper and lower theoretical limit. We match predictions well for the superbubble case. This is due to conduction drawing mass inward into the hot bubble as well as sub-grid evaporation converting large amounts of cold material to the hot phase.

the 65% predicted by Weaver (1977), however. We plot the energy in this way to draw attention to the self-similarity of the superbubble model. Once the bubble reaches its self-similar phase, it will retain the same fraction of the supernova energy for as long as the energy source survives. A bubble that ideally replicates Weaver's answer would come to rest on the dotted line and stay there. Our model has reached a lower equillibrium where approximately 40% of the injected energy is protected from cooling. This missing energy is one reason why the superbubble's radius is still less than Weaver's analytical solution, with the other reason being the smearing of the cold shell. In all three cases, the thermal energy after 10 Myr was within 2% of the 70% share of the total energy calculated by Weaver. While we do expect some amount of radiative cooling to occur even within the hot bubble, the timescale involved is typically much larger than the dynamical timescales of superbubbles in a galactic environment, and for our bubble is nearly 80 Myr (Mac Low and McCray, 1988). One possible explanation for this energy loss comes from El-Badry (2019) who predict energy loss from cooling in the interface between the cold shell and the hot interior. Whereas Weaver (1977) assumed a sharp interface where material did not have time to cool as it was being evaporated into the bubble, in reality there will be a temperature gradient in which some gas will exist at the right temperature for rapid cooling. As will be shown in Section 4.5 (in Figure 4.22), we do have some material in this transitional region. Another contributing factor may be our treatment of evaporation, specifically the way it scales with resolution. This will be discussed more in relation to our convergence study in Section 4.5.

Figure 4.9 plots the hot mass within the bubble over time, alongside the Weaver solution and two extreme limits. The lower limit is the supernovae ejecta; the mass physically flung off of the star cluster as its stars explode. The

upper limit is that ejecta plus the mass of the shocked ISM, i.e. all of the material that the bubble has displaced as it spreads through the galaxy. In the default case, the bubble mass is little more than the ejecta mass, as the bubble fails to heat up any material from the ISM it impacts. When conduction is turned on, the mass is larger by about a factor of two, even though the radius and energy are the same. This is due to conduction dragging mass inwards from the cold shell into the hot interior. This effect is increased by an order of magnitude when the superbubble model is turned on. Part of this is due to the extra energy allowing the bubble to expand further. The rest is caused by our subgrid evaporation model, which explicitly converts cold mass into hot near the edge of the bubble. The end result is a superbubble that matches Weaver's prediction for the amount of mass affected, but falls slightly short on energy. Even with this missing energy, it is a dramatic improvement on the standard case.

4.4 Radial Profiles

In this section we present radial profiles of density, temperature, and pressure. Profiles describe the shape of the bubble at a given time. These profiles are snapshots of the simulation taken at a time of 10 Myr, just before the feedback is switched off.

Figures 4.10-4.12 show the radial profiles for the three different physical cases at our highest resolution. Immediately it is clear that the profiles for the superbubble case are much less spread-out than those for the non-superbubble cases. This is because the superbubble case is more spherically symmetric, as discussed in Section 4.2.

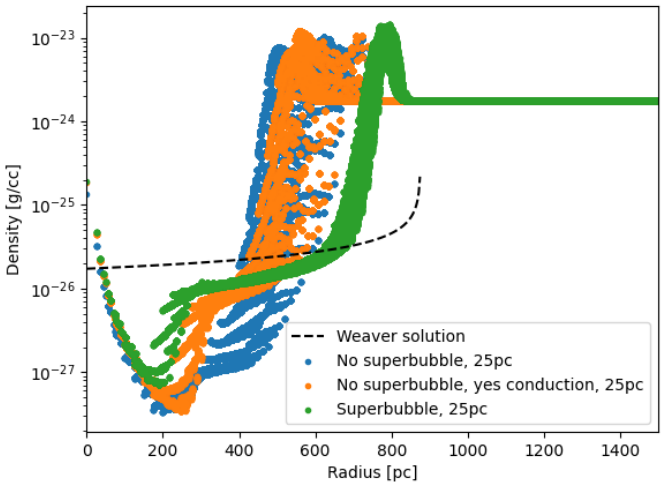

Figure 4.10: Radial profile of density for each of the three physical cases at 10Myr, compared with the analytical solution from Weaver et al (1977). The Weaver solution cuts off at the point where it has reached the cold shell, as the ISM beyond is not part of the solution.

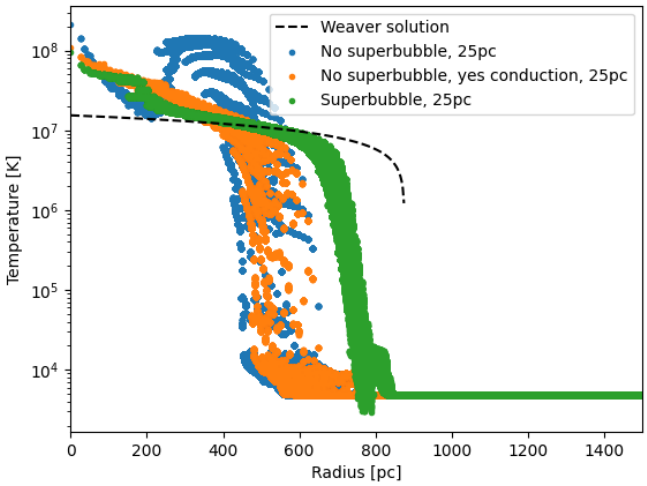

Figure 4.11: Radial profile of temperature for each of the three physical cases at 10Myr, compared with the analytical solution from Weaver et al (1977).

The cold shell can be seen in the superbubble case as the "bump" near 800 pc where the temperature dips and the density peaks. A similar feature is seen in the non-superbubble cases, but it is weaker and spread across a larger region of space. Comparing the two non-superbubble cases shows the effects of conduction; it has increased density and lowered temperature inside the bubble by conducting material inwards. These effects are only more pronounced when evaporation is introduced.

All of these profiles show features not seen in the simplistic Weaver predictions, especially in the central region of the bubble. The key to this discrepancy lies in the pressure profile. Weaver assumes a completely uniform pressure within the bubble, whereas we find a dip in the centre. This dip explains why our density and temperature values undershoot and overshoot Weaver's predictions respectively. The other main difference between our results and the Weaver prediction is the lower radius, which is primarily caused by the extra energy loss discussed in the previous section.

4.5 Convergence Study

In this section, we plot our superbubble results at each of three different resolutions; 25, 50, and 100 pc. We aim to demonstrate convergence towards the Weaver (1977) solution, as well as determine what qualitative effects changing the resolution has, if any.

Figures 4.13-4.15 plot the time evolution of the superbubble's radius, energy, and mass at each of our three resolutions. First, let us turn our attention to the energy evolution (Figure 4.14). At the lower resolutions, less energy is

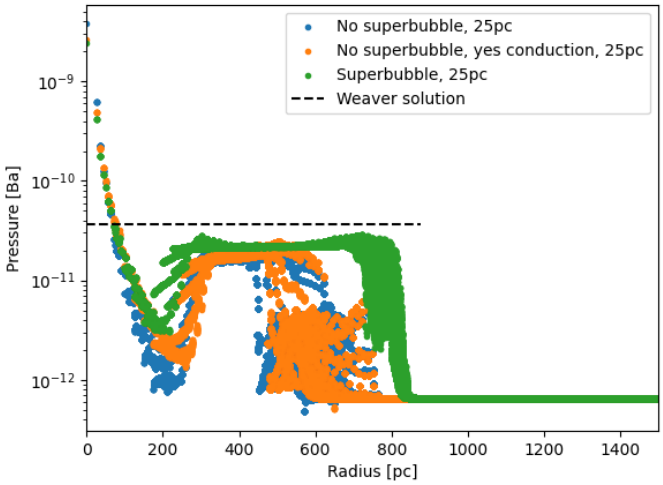

Figure 4.12: Radial profile of pressure for each of the three physical cases at 10 Myr, compared with the analytical solution from Weaver et al (1977).

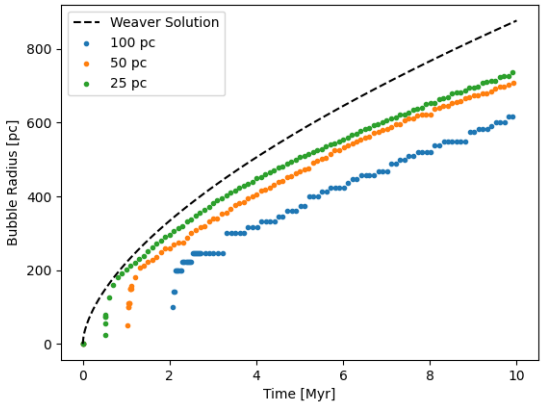

Figure 4.13: Time evolution of superbubble radius at three different resolutions, compared to the analytical solution from Weaver et al. (1977). Radius is larger at higher resolution, even though there is less energy as resolution increases. This is due to our definition of radius as the distance where temperature first rises above 10^6 K. This choice leads to unintuitive results at low resolution, and is likely an underestimate. For more discussion of this problem, refer to Section 4.3.

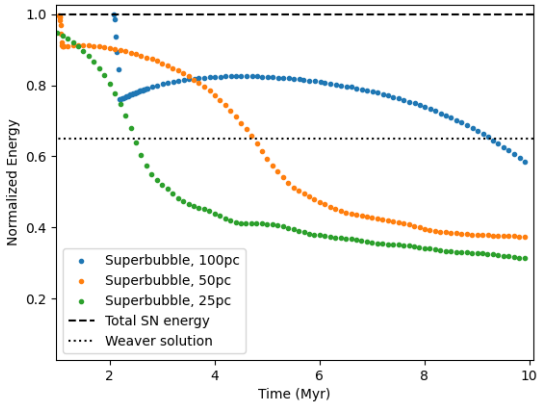

Figure 4.14: Time evolution of superbubble energy at three different resolutions, compared to the analytical solution from Weaver et al. (1977). More energy is lost at higher resolutions. This is likely due to the fact that evaporation rate in our model scales with resolution. Evaporation brings material within the temperature range where rapid cooling occurs, leading to energy loss.

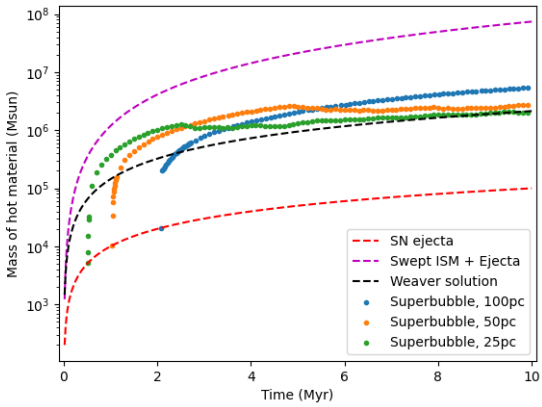

Figure 4.15: Time evolution of superbubble mass at three different resolutions, compared to the analytical solution from Weaver et al. (1977) and to theoretical upper and lower limits. Lower resolutions have higher mass due to their increased energy and size.

lost. This is likely due to the increased effect of evaporation at higher resolution (see Section 3.2). Evaporation reduces the temperature of the hot phase and raises it in the cold, often causing material to fall within the range where it will cool rapidly. Therefore, in higher-resolution simulations, where we have more evaporation due to the way our model scales with cell size, we will lose more energy to cooling. If our higher-resolution runs have overestimated the rate of evaporation, it could explain the energy gap. On the other hand, it is possible that this level of evaporation is fine and the energy gap is explained by the extra radiative cooling predicted by El-Badry et al. (2019). More investigation into alternative evaporation models is needed before any definitive conclusions can be drawn here.

Despite retaining more energy, the lower-resolution runs have smaller radii (Figure 4.13). This is a result of us defining radius as the distance where temperature first rises above 10^6 K, as discussed in Section 4.4. Since the cold shell is smeared across much larger distances at lower resolution, the effective radius seems to shrink. However, a quick glance at the profile plots in Figures 4.16-4.18 show that lower resolution clearly results in a larger radius. Our definition of radius has underestimated the extent of the bubble, and this underestimation is more severe at lower resolutions. The lower resolution runs do indeed have a larger extent and heat up more hot mass accordingly.

The profile plots in Figures 4.16- 4.18 show a snapshot at 10 Myr of the density, temperature, and pressure, for the superbubble case at three different resolutions. Convergence is clearly seen as the jump from 100 pc to 50 pc is much larger than the jump from 50 pc to 25 pc. Improving the resolution makes features such as the density peak and cold shell sharper, and also significantly

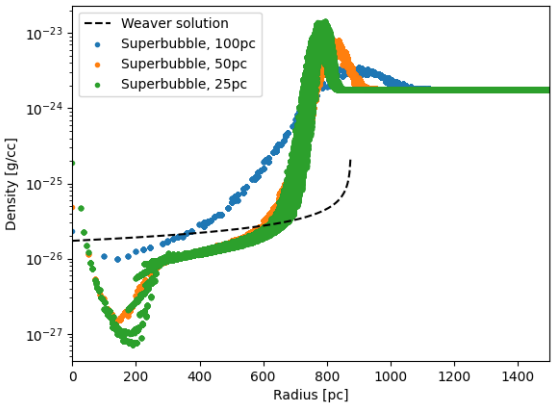

Figure 4.16: Radial profile of superbubble density at three different resolutions at 10 Myr, compared to the analytical solution from Weaver et al. (1977). Structure of the inner bubble is noticeably different at the higher resolutions, dipping below the Weaver solution at the very centre.

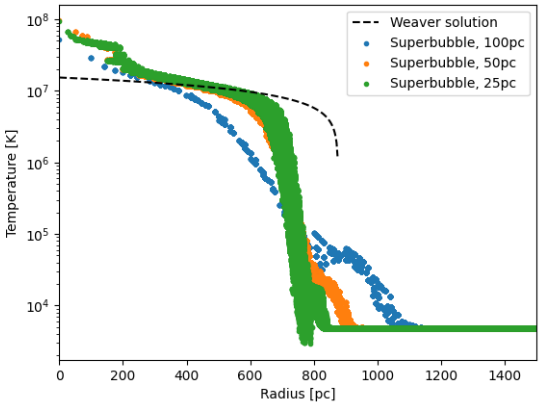

Figure 4.17: Radial profile of superbubble temperature at three different resolutions at 10 Myr, compared to the analytical solution from Weaver et al. (1977). Steepening of temperature gradient with increases resolution is very prominent.

changes the shape in the inner bubble region. This is likely due to the energy loss caused by stronger evaporation at higher resolutions, as discussed in the previous section.

The most important change in these profiles with resolution is the steepening of the temperature gradient. We expect the transition between the cold shell and the inner bubble to be extremely sharp, as indicated by the Weaver (1977) solution (dashed lines). As our resolution improves, the temperature gradient sharpens to reflect this. Even still, we are nowhere close to being able to resolve the shell. This steepening increases the effective radius of the bubble, since we have defined the radius to be the distance at which the temperature first reaches 10^6 K. This is why the lowest-resolution has the lowest radius in (Figure 4.13) despite clearly having a greater extent in Figures 4.16- 4.18; the shallow increase in temperature lowers the effective radius in our definition. As discussed in Section 4.3, there is no definition of radius that does not either exclude some material we want to include, or include material that we do want. The main takeaway from these figures is that lower resolution does indeed result in a larger explosion that heats more mass, but that it does so via a more shallow and spread-out bubble.

An important aspect of the plots in the time evolution convergence tests (Figures 4.13- 4.15) is that each resolution appears to begin at a different time. This is due to a quirk of the way timesteps are determined in RAMSES. When the simulation begins, our simulation box is completely uniform. This causes the first timestep to be much bigger than the others, since the star has not started exploding yet. This does not affect the energy of the feedback, as the appropriate amount of energy is still injected based on the amount of time passed since

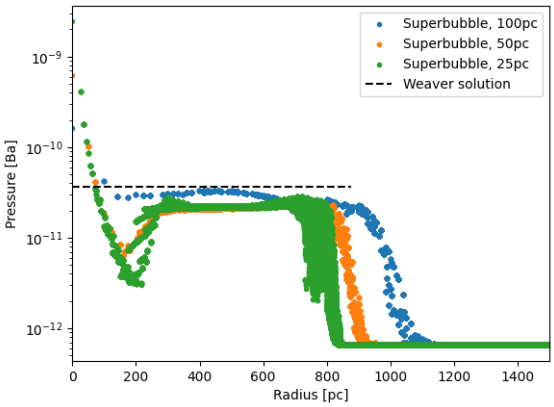

Figure 4.18: Radial profile of superbubble pressure at three different resolutions at 10 Myr, compared to the analytical solution from Weaver et al. (1977). As resolution improves, the shock between the ISM and the cold shell steepends dramatically.

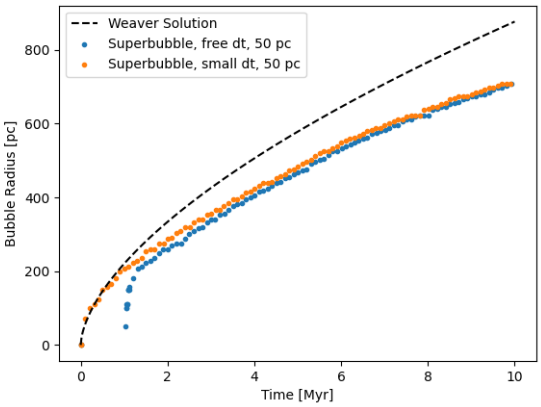

Figure 4.19: Effects of timestep on the radius of the superbubble. With a lower timestep, the explosion proceeds more gradually at early times, more closely matching the Weaver (1977) scenario of constant energy injection. However, at later times, the two scenarios converge to the same answer.

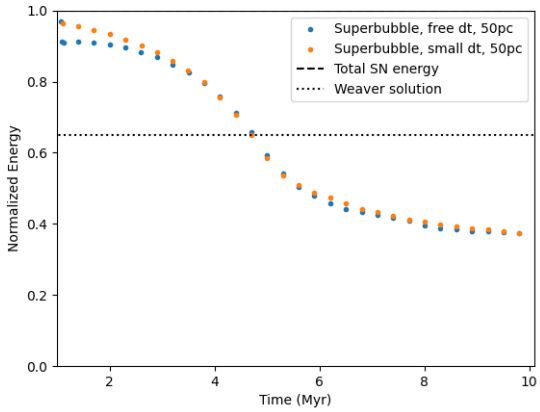

Figure 4.20: Effects of timestep on the energy of the superbubble. The small timestep and free timestep cases rapidly converge. Since the gradual supernova feedback model inputs the correct amount of energy regardless of timestep, this is not a surprise. The initial gap between the two cases at early times is most likely due to a larger amount of cooling occuring in the first, overly-large timestep.

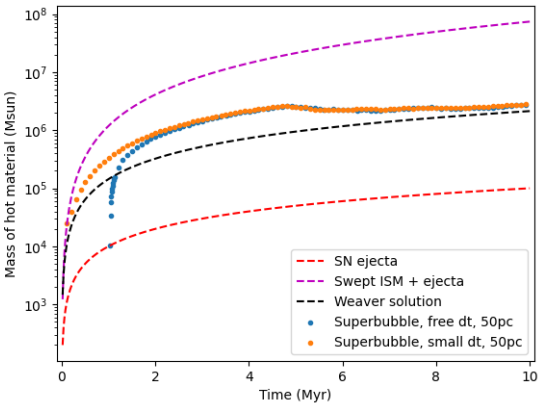

Figure 4.21: Effects of timestep on the mass of the superbubble. Much like the radius case, the explosion originally proceeds more gradually but eventually converges with our standard run.

t=0. However, it does mean that more energy is injected in the first step than in any other, which could impact the overall results by making the energy injection less smooth than desired. To investigate this effect, a new run was performed with a very small timestep enforced at all times at our medium resolution of 50 pc. Figures 4.19- 4.21 plot the time evolution of this new run against the 50 pc run with regular timesteps (chosen by RAMSES). In all cases, the results converge by 10 Myr. Our simulations are therefore physically comparable to the Weaver (1977) scenario of constant energy injection, except at early times when the initial burst of energy still dominates. Still, a fix for this problem is desired. In a galaxy-scale simulation, this problem would be harder to notice as there usually would be something demanding a small timestep. It is conceivable that it could come up though, and it is dangerous to rely on the default timesteps always working out in our favour. Future work in this area should focus on modifying the way RAMSES determines its timesteps, possibly by allowing the code to predict upcoming feedback events and adjust accordingly. Another option would be to simply keep enforcing a tiny timestep manually, but this would cost computing time and come across as too much of a cheap hack, as if we did not trust the code to do its job properly.

Finally, in Figure 4.22 we present plots of the density and temperature of our superbubbles separated into their hot and cold phases, to demonstrate the model's applicability even at lower resolution. We show radial profiles of both the hot and cold phases for each superbubble resolution at 10 Myr. In every case, the hot phase dominates inside the bubble. In the higher-resolution cases, the cold phase has been almost completely evaporated out to the cold shell. Hot temperature is capped at 10^5, the temperature where evaporation becomes insignificant. The lack of material sitting at this temperature in the low-res case

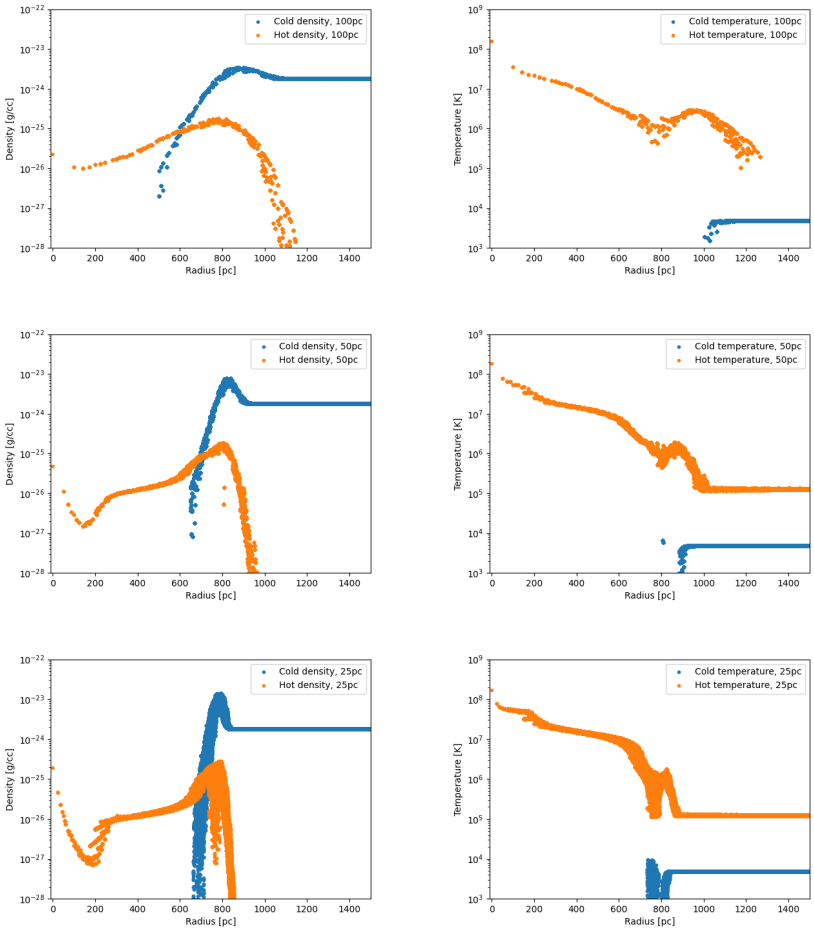

Figure 4.22: Profiles of density and temperature of both the hot and cold phases for each superbubble resolution at 10 Myr. In every case, hot phase dominates inside the bubble, and cold phase dominates outside. Hot temperature is capped at 10^5, the temperature where evaporation becomes insignificant. The lack of material sitting at this temperature in the low-res case could be due to weaker evaporation at that resolution. There does exist a small amount of material in the 10^4 - 10^6 K rapid cooling region, which could result in extra cooling compared to Weaver (1977). Note that the hot phase has travelled beyond the edge of the bubble via numerical diffusion. This does not affect the evolution of the bubble because the hot density is negligible in that region.

could be due to weaker evaporation at that resolution. There does exist a small amount of material in the 10^4 - 10^6 K rapid cooling region, which could result in extra cooling compared to Weaver (1977) as noted earlier. Note that in every case the hot phase has travelled beyond the edge of the bubble via numerical diffusion. This does not significantly affect the evolution of the bubble because the density of the hot phase is negligible in that region.

Chapter 5

Conclusions

Accurate, first-principle-based simulations of galaxy evolution have long been hampered by a lack of physically motivated feedback models without too many free parameters. With powerful modern computers and a few clever tricks, that era may soon be coming to an end. By implementing a two-phase model for the ISM and including thermal conduction, we have built a model for feedback that comes closer to reproducing the superbubbles seen in nature than the rest of the field without relying on ad hoc adjustments or free parameters. Our bubbles are significantly larger than those made without these changes, and includes more prominent cold shells and sharper shocks. These results mirror those in Keller's earlier work in Gasoline, demonstrating that the superbubble model has merit in both particle-based and grid-based codes.

Our superbubble resembles the predictions of Weaver et al. (1977) in broad terms. By splitting the gas into a cold phase and a hot phase with separate physical properties, we are able to avoid the overcooling problem, especially early in the bubble's evolution before it can be resolved. We generate roughly spherical self-similar bubbles with a cold shell, an interior hot enough to avoid rapid cooling, and conductive mass flux from the shell into the interior. There is one key area where our results disagree with Weaver's, however; the amount of energy retained within the bubble. Where Weaver predicts 65% of the feedback energy remains in the bubble, our bubble keeps only 40%. There are two likely factors involved. The first comes from El-Badry et al. (2019) who predict that as mass evaporates from the cold shell into the bubble, it will experience some

amount of radiative cooling as it passes through the intermediate temperature range where cooling rate peaks. Weaver, in contrast, assumes that this evaporation is rapid enough to prevent any cooling from occurring. Our superbubble does contain some (small amount of) matter in this temperature range, and so we should expect to lose some energy compared to the analytical solution in the no-cooling assumption. However, as El-Badry's simulations were both 1-dimensional and extremely high resolution, it is difficult to use them to predict exactly how strong this expected energy loss really is. One way to test this would be to record specifically any cooling that occurs from recently-evaporated material.

The other factor likely contributing to the energy loss is our treatment of sub-grid evaporation. Our evaporation model scales with resolution, growing more powerful as the resolution increases. This is a result of our assumption that the hot gas within each evaporative cell forms a sphere. This assumption means that when the resolution is increased and a single cell is split into multiple, the overall surface area through which evaporation can occur increases. This somewhat unintuitive result is not a crucial part of our evaporation model, and a different treatment of the shape of the evaporative interface would lead to different results. The fact that our superbubble retains less energy than Weaver's prediction means that we have a smaller radius over time. However, the mass of our bubble does not suffer the same blow, matching the theoretical prediction very well. Since evaporation works to both lower the hot bubble's energy and increase its mass, this is further evidence that our evaporation is too strong at higher resolutions. Before moving forward with this superbubble model, we would suggest taking a second look at evaporation, specifically the effective radius of the evaporating surface. A more robust method of determin-

ing what this surface looks like on a sub-grid level is needed.

Other than evaporation, one other area where the code could use additional work is in the timestep selection. Our resolution convergence tests (Section 4.5) display a problem that was a recurring issue throughout this project: the timestep is chosen purely based off of the current state of the fluid without regard for upcoming feedback events. This causes issues when a large timestep skips over a feedback event and our gradual supernova model has to play catch-up, dumping a large amount of energy step. It also makes comparing runs of different resolutions more difficult, as they will appear to have different start times for feedback events. While this didn't affect our results significantly, it would be safest to fix this issue before proceeding with larger runs. We also encountered situations where it would be beneficial for a certain process (such as evaporation) to operate on a different timestep than the fluid dynamics (see Figure 3.3) or to at least be considered as part of the timestep calculations, but this is not possible in default RAMSES. RAMSES already uses AMR to selectively lower its timesteps in regions where a smaller step is called for. This functionality could be used to solve these problems by writing a new refinement method that takes future feedback events and strong evaporation into account.

After sorting out these issues, the next step would be to apply this superbubble model to larger and more complex regions of space. We would follow a path similar to Keller et al. (2014), who applied their superbubble model to a Milky Way-sized galaxy and a dwarf galaxy. We would like to see what effect a less uniform ISM has on our bubble, as well as their effect on galaxy evolution over larger timescales. We expect to see much less spherical results, whether that be through fragmentation caused by Vishniac instabilities (Vishniac 1983)

or the formation of chimneys and other elongated structures as the bubble expands out of the galactic disk (Kompaneets 1960, Norman 1989). This non-spherical behaviour makes the need for a evaporation model not built around assumptions of spherical surfaces even more important moving forward. In a galaxy-scale simulation, key indicators of success would be long-term regulation of the star formation rate and the overall mass of the gaseous disk. Keller et al. then expands their work to include cosmological galaxy evolution (2015) and later probes the limits of stellar feedback by testing very massive galaxies (2016) where they conclude that some additional form of feedback (likely AGN) is needed to fully explain observations for these galaxies. RAMSES is already equipped to handle cosmological cases if we wished to follow this route.

While the superbubble model easily beats RAMSES' default feedback method of basic thermal energy injection, there are still other methods we could compare it to. In a summary of the field, Scannapieco et al. (2012) compare 13 state-of-the-art (at the time) cosmological hydrodynamics simulations across nine different codes, each trying to model the evolution of a Milky Way-like galaxy. They feature a variety of the feedback methods described in Section 1.2; thermal injection, kinetic feedback, delayed cooling, decoupling of outflow gas from the rest of the ISM, and even a model that includes AGN feedback. They conclude that none of these models can satisfyingly replicate observed galaxies. In future work where we will test the superbubble in galaxy-scale environments, we must be sure to test not just against default RAMSES feedback, but against these other methods and codes as well. The different models used in Rosdahl et al. (2017) would be perfect comparisons, as they are already implemented within RAMSES.

As we head into a new decade of galaxy simulation, the need for dependable feedback models is greater than ever. Upcoming projects like the James Webb Space Telescope (Garner et al. 2006) will shed new light on the formation and evolution of galaxies, and so there is no better time than now for a model with strong predictive power and wide applicability. Now that the superbubble method has been demonstrated in both Gasoline and RAMSES, two wildly different codes, we believe that it is the model simulators have been waiting for. Having access to a superbubble model in both types of codes will greatly increase the availability of the model by allowing anyone to use it regardless of their preferences for particle-based or grid-based techniques. With just a few more tweaks, the model described in this work will be ready for application to full-blown cosmological galaxy simulations. We hope that the superbubble will soon grow in popularity to earn its place among every simulator's repertoire.

Bibliography

Agertz, O. et al. (June 2013). Toward a Complete Accounting of Energy and Momentum from Stellar Feedback in Galaxy Formation Simulations. ApJ 770(1), A25

Bryan, G. L. et al. (March 2014). Enzo: an adaptive mesh refinement code for astrophysics. ApJ Supplement Series 211(2).

Cash, W. et al. (June 1980). The X-ray superbubble in Cygnus. ApJ 238, L71-L76.

Castor, J., McCray, R., Weaver, R. (Sept 1975). Interstellar Bubbles. ApJ 200, L107-L110.

Cecil, G., Bland-Hawthorn, J., Veilleux, S. (Sept 2002). Tightly Correlated X-Ray/Hα-emitting Filaments in the SUperbubble and Large-Scale Superwind of NGC 3079. ApJ 576(2), 745-752.

Chabrier, G. (July 2003). Galactic Stellar and Substellar Initial Mass Functions. PASP 115, 763-796

Conselice, C. J., Wilkinson, A., Kenneth, D. and Mortlock, A. (Oct 2016). The evolution of galaxy number density at z $<$ 8 and its implications. ApJ 830(2)

Dale, J. E. (Aug 2015). Modelling of feedback in star formation simulations. New Astron. Rev. 68, 1-33.

Dalla Vecchia, C. and Schaye, J. (Oct 2012). Simulating galactic outflows with thermal supernova feedback. Mon. Notices Royal Astron 426(1), 140-158.

Dubois, Y. and Commerçon, B. (Oct 2016). An implicit scheme for solving the anisotropic diffusion of heat and cosmic rays in the RAMSES code. Astron. Astrophys. 585, A138.

Durier, F. and Dalla Vecchia, C. (Jan 2012). Implementation of feedback in smoother particle hydrodynamics: towards concordance of methods. Mon.

Notices Royal Astron 419(1), 465-478.

El-Badry, K., Ostriker, E. C., Kim, C., Quataert, E., and Weisz, D. R. (Dec 2019). Evolution of supernovae-driven superbubbles with conduction and cooling. Mon. Notices Royal Astron 490(2), 1961-1990.

Gardner, N. P. et al. (Nov 2006). The James Webb Space Telescope. Space Science Reviews 123, 485-606.

Godunov, S. K. (1959). A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. Mat. Sb. 47(89). 271-306

Grond, J. J., Woods, R. M., Wadsley, J. W., and Couchman, H. M. P. (Feb 2019). TREVR: a general $Nlog^2N$ radiative transfer algorithm. Mon. Notices Royal Astron 485(3), 3681-3695.

Ho, S. H., Martin, C. L. and Turner, M. L. (April 2019). How Gas Accretion Feeds Galactic Disks. ApJ 875

Katz, N. (June 1992). Dissipational Galaxy Formation. II. Effects of Star Formation. ApJ 391, 502.

Keller, B. W., Wadsley, J., Benincasa, S. M., and Couchman, H.M.P. (June 2014). A superbubble feedback model for galaxy simulations. Mon. Notices Royal Astron 442(4), 3013-3025.

Keller, B. W., Wadsley, J., and Couchman, H.M.P. (Nov 2015). Cosmological Galaxy Evolution with Superbubble Feedback I: Realistic Galaxies with Moderate Feedback. Mon. Notices Royal Astron 453(3), 3499-3509.

Keller, B. W., Wadsley, J., and Couchman, H.M.P. (Dec 2016). Cosmological Galaxy Evolution with Superbubble Feedback II: The Limits of Supernovae. Mon. Notices Royal Astron 463(2), 1431-1445.

Kennicutt, R. C. (May 1998). The global Schmidt law in star-forming galaxies ApJ 498, 541-552.

Kim, J. et al. (Dec 2016). The Agora high-resolution galaxy simulations comparison project. II. isolated disk test. ApJ 833, 202-236.

Kompaneets, A. S. (July 1960). A Point Explosion in an Inhomogeneous Atmosphere. Soviet Physics Doklady, 5, 46.

Landua, L. D. and Lifshitz, E. M. (1959). Fluid Mechanics. Pergamon Press, London.

Leitherer, C. et al. (July 1999). Starburst99: Synthesis models for galaxies with active star formation. The Astrophysical Journal Supplement, 123, 3-40.

Mac Low, M. and McCray, R. (Jan 1988). Superbubbles in disk galaxies. ApJ 324, 776-785.

Marri, S. and White, S. D. M. (Oct 2003). Smoothed particle hydrodynamics for galaxy-formation simulations: improved treaments of multiphase gas, of star formation adn of supernovae feedback. Mon. Notices Royal Astron 345, 561-574.

Mathew, W. G. (1966). On the central hole in NGC 2237-2246. ApJ 144, 206-215.

McKee, C. F. and Ostriker, J. P. (Nov 1977). A theory of the interstellar medium: three components regulated by supernova explosions in an inhomogeneous substrate. ApJ 218, 148-169.

Menon, T. K. (1962) A study of the rosette nebula NGC 2237-46. ApJ 135, 394-407.

Naab, T. and Ostriker, J. P. (Aug 2017). Theoretical Challenges in Galaxy Formation. Annu. Rev. Astron. Astrophys. 55, 59-109

Navarr, J. F. and White, S. D. M. (May 1993). Simulations of dissipative galaxy formation in hierarchically clustering universes - I. Tests of the code. Mon. Notices Royal Astron 265, 271-300.

Norman, C. A. and Ikeuchi, S. (Oct 1989). The Disk-Halo Interaction: Super-

bubbles and the Structure of the Interstellar Medium. ApJ 345, 372.

Rosdahl, J. et al. (April 2017). Snap, crackle, pop: sub-grid supernova feedback in AMR simulations of disc galaxies. Mon. Notices Royal Astron 466(1), 11-33.

Scannapieco, C. et al. (June 2012). The Aquila comparison project: the effects of feedback and numerical methods on simulations of galaxy formation. Mon. Notices Royal Astron 423(2), 1726-1749.

Shapley, H. and Curtis, H. D. (May 1921). The Scale of the Universe. Bulletin of the National Research Council, Vol. 2, Part 3, No. 11, 171-217.

Smith, B. D. et al. (Dec 2016). GRACKLE: a chemistry and cooling library for astrophysics. Mon. Notices Royal Astron 466, 2217-2234.

Sommer-Larsen, J., Martin, G., Laura, P. (Oct 2003). Galaxy Formation: Cold Dark Matter, Feedback, and the Hubble Sequence. ApJ 596(1), 47-66.

Spitzer, L. (1962). Physics of Fully Ionized Gases. Interscience Publishers, New York.

Springel, V. and Hernquist, L. (Feb 2003). Cosmological smoother particle hydrodynamics simulations: a hybrid multiphase model for star formation. Mon. Notices Royal Astron 339(2), 289-311.

Stinson, G. et al. (Nov 2006). Star formation and feedback in smoothed partile hydrodynamic simulations - I. Isolated galaxies. Mon. Notices Royal Astron 373(3), 1074-1090.

Teyssier, R. (April 2002). Cosmological hydrodynamics with adaptive mesh refinement. A new high resolution code called RAMSES. Astron. Astrophys. 385, 337-364

Teyssier, R., Pontzen, A., Dubois, Y., and Raed, J. I. (March 2013). Cusp-core transformations in dwarft galaxies: observational predictions. Mon. Notices Royal Astron 429(4), 3068-3078.

Thacker, R. J. and Couchman, H. M. P. (Dec 2000). Implementing Feedback in Simulations of Galaxy Formation: A Survey of Methods. ApJ 545, 728-752.

Turk, M. J. et al. (Jan 2011). yt: A Multi-code Analysis Toolkit for Astrophysical Simulatino Data. The Astrophysical Journal Supplement, 192(1) A9.

Vishniac, E. T. (Nov 1983). The dynamic and gravitational instabilities of spherical shocks. ApJ 274, 152-167.

Wadsley, J. W., Keller, B. W., and Quinn, T. R. (June 2017). Gasoline2: a modern smoothed particle hydrodynamics code. Mon. Notices Royal Astron 471(2), 2357-2369.

Weaver, R., McCray, R., Castor, J., Shapiro, P., Moore, R. (Dec 1977). Interstellar bubbles. II. Structure and evolution. ApJ 218, 377-395.