THE UNIVERSITY
of EDINBURGH

A New Residual Distribution Hydrodynamics Solver for Galaxy Formation Simulations

Ben Morton

Doctor of Philosophy
The University of Edinburgh
26/10/2020

"Your assumptions are your windows on the world. Scrub them off every once in a while, or the light won't come in."

– Isaac Asimov

Lay Summary

Much of the observed Universe is made up of gas, mostly hydrogen and helium. This gas is the material from which stars are formed. Stars, in turn, produce much of the light used to observe the Universe. The behaviour of the gas, the way it moves and evolves, can be described mathematically, but the exact solution cannot be found by hand. The motion and interaction of the gas is simply too complex to be found by solving the equations exactly, except in the most simple scenarios. Instead, we find approximate solutions to these problems by solving the equations numerically. To put it simply, the region of gas that we want to model is broken down into a grid of cells. The flow of material through each face of a given cell, for a set amount of time, is calculated. That amount of material is moved through the face, and the amount of gas in each cell is updated. This process is repeated for every cell, and for many time steps. Alternatively, the evolution of the gas is tracked by calculating the motion of a set of tracer particles, each representing a certain mass of gas. In this work, I present the implementation of a new method for modelling the behaviour of this gas. Built around triangular cells, this approach solves the flow of the gas in every dimension at once, instead of breaking the problem down into a series of flows through the faces of a cell, which effectively only solves the problem in one dimension at a time. This new approach captures the complex interplay of flows that have component flows in multiple directions at once. I show extensive testing of this new scheme, which is not widely used in astrophysical simulations. In addition, I cover my own derivation and implementation of a number of vital extensions to this approach, including going from two to three dimensions.

I also discuss my research into the behaviour of a region of gas as an object with a large amount of mass passes through it. The gravity from the travelling object pulls the gas into a wake behind it. The gravity from this wake produces a force that slows the object down, a process known as dynamical friction. This is a simplified version of a scenario found in many astrophysical systems, from merging galaxies, down to newly forming planets. I compare the theoretical predictions, for the formation and structure of this wake, to results found using a state-of-the-art numerical solver to calculate the evolution of the gas. I found that there were significant differences between the expected structure and the numerical results. The drag forces, that I found from the numerical results, were systematically lower than predicted, suggesting that current simulations may not be capturing the full effects of dynamical friction.

Abstract

Numerical simulations are key to our understanding the complex physical processes present in the formation and evolution of galaxies. The vast majority of the baryonic component is in a gaseous state, modelled by solving the fluid equations, using a variety of methods. I present a new implementation of the 2D residual distribution (RD) family of hydrodynamics solvers. Built around an unstructured mesh, RD solvers produce truly multi-dimensional solutions to the underlying fluid equations, with second order accuracy in both time and space. The implementation accurately reproduces the solutions to many standard hydrodynamics tests. I compare the RD results to solutions from state-of-the-art meshless finite mass (MFM) and meshless finite volume (MFV) solvers. I present extensions to the RD method, deriving an adaptive time stepping regime, and the 3D version of the solver. I also show a numerical study of idealised gaseous dynamical friction (DF) using the MFM solver, for both supersonic and subsonic flows, highlighting the need for accurate solvers. This solver produces a wake that systematically under-produces the expected retarding force in supersonic cases. The over-dense wake it forms does not replicate the expected sharp density profile and produces a bow shock where none is predicted. I compare this regime to that found in cosmological simulations, demonstrating that much of the dark matter substructure in the early universe will experience these conditions, suggesting DF driven mergers may be underestimated in current simulations. I propose a new standard gravo-hydrodynamical test based on the idealised DF setup. I add simulations that include molecular chemistry, showing how DF at early times can stimulate the formation of molecular hydrogen, critical to the formation of the first stars and structures.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Ben Morton, 26/10/2020*)

Acknowledgements

I am pleased to acknowledge the great support, advice and guidance provided by Prof. Sadegh Khochfar, without whom I would not have started this project, let alone completed it. I would like to thank Dr. Jose Oñorbe for all his help, and Dan, BK, and Andrea for many useful discussions, throughout the project. I would also like to thank Dr. Sijme-Jan Paardekooper for very helpful discussions about the RD solver. I compare results from my own work to test outputs supplied by Isaac Alonso Asensio. Thank you also to Fran, Adam, Ryan M, Joe K and everyone else at the IfA, who made it such a welcoming and fruitful place to work over the last four years.

Part of this work was performed using the Cambridge Service for Data Driven Discovery (CSD3), part of which is operated by the University of Cambridge Research Computing on behalf of the STFC DiRAC HPC Facility (www.dirac.ac.uk). The DiRAC component of CSD3 was funded by BEIS capital funding via STFC capital grants ST/P002307/1 and ST/R002452/1 and STFC operations grant ST/R00689X/1. DiRAC is part of the National e-Infrastructure. I acknowledge that the results of this research have been achieved using the DECI resource ARCHER based in the UK at EPCC with support from the PRACE aisbl. This work also used the Cirrus UK National Tier-2 HPC Service at EPCC (http://www.cirrus.ac.uk) funded by the University of Edinburgh and EPSRC (EP/P020267/1).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this work, I present simulations using a variety of numerical hydrodynamics modelling techniques. This includes results from a suite of idealised dynamical friction simulations, investigating the characteristics and effects of this process in simulations of early galaxy formation, and the comprehensive derivation, testing and extension of a new, truly-multidimensional, residual distribution hydrodynamics solver. The detail of the work revolves around the implementation and use of advanced methods for modelling the behaviour of baryonic gas, with some reference to the broader context of galaxy formation and evolution. In this chapter, I will first describe the observed characteristics of the galaxies that these techniques are used to model. Following this, I summarise the processes that impact the state of the baryonic gas, which require the development and use of these complex numerical tools. The final section of the chapter is a description of the fundamental equations which describe the physical behaviour of the baryonic gas that we seek to model, and the derivation of the underlying numerical methods upon which these tools are based.

1.1 Galaxy Formation and Evolution

Galaxies, gravitationally bound systems of stars, gas, and dust, are objects of great interest to astrophysicists. Since their identification as objects beyond the Milky Way (Hubble, 1929), they have been studied with both observations and numerical simulations (Somerville & Davé, 2015). They host the stars that

produce the light by which we study the universe. There is great complexity in the variety of their properties, and in the physics that governs their formation and evolution. Observations reveal a huge range in morphologies, colours, and luminosities, with more detailed analyses showing wide variation in mass, environment and stellar composition. If we are to understand these complex objects, including the Milky Way itself, we must first understand the physics of galaxy formation and evolution. This requires us to understand all the governing physical processes, from their origins within the large scale structure of the Universe, down to the details of gas dynamics and star formation.

Numerical simulations form a vital part of our current tools for investigating and understanding the universe. Under the standard model of cosmology, the cold dark matter model with a cosmological constant (ΛCDM), rapid progress has been made in our ability to simulate structure formation (White & Rees, 1978; Vogelsberger et al., 2014a; Schaye et al., 2015). To do this, several mass components must be considered: collisionless cold dark matter (DM), which experiences only gravitational interactions, stars, which can also be modelled as collisionless particles, and baryonic gas, which behaves as a collisional fluid. The components are set within a comoving coordinate scheme that encodes the expansion of the Universe dictated by the ΛCDM model. These simulations seek to predict observational properties of galaxies by accurately modeling the gravitational response of the dark matter, and the gravo-hydrodynamic response of gaseous baryons. All processes important in shaping the evolution of structure need to be considered. These include the fundamental forces, such as gravity on the dark and baryonic matter, and the hydrodynamics of the baryonic gas. Other forces and processes can also play an important role, such as electromagnetic radiation, which can be modelled using radiative transfer, and magnetic fields, which require the equations of hydrodynamics to be converted to handle magneto-hydrodynamics. Secondary processes, such as star formation, and the corresponding feedback mechanisms, must also be included to fully capture the evolution of galaxies. In this section, I describe the properties and features of observed galaxies (Section 1.1.1), particularly those that require advanced hydrodynamics solvers to model. I cover the basics of the standard cosmology, within which the baryonic gas, of these simulations, is modelled (Section 1.1.2). Finally, I focus on the processes affecting the baryonic gas in particular, most notably how the gas is cooled (Section 1.1.3), and how star formation impacts the state and distribution of the gas (Section 1.1.4).

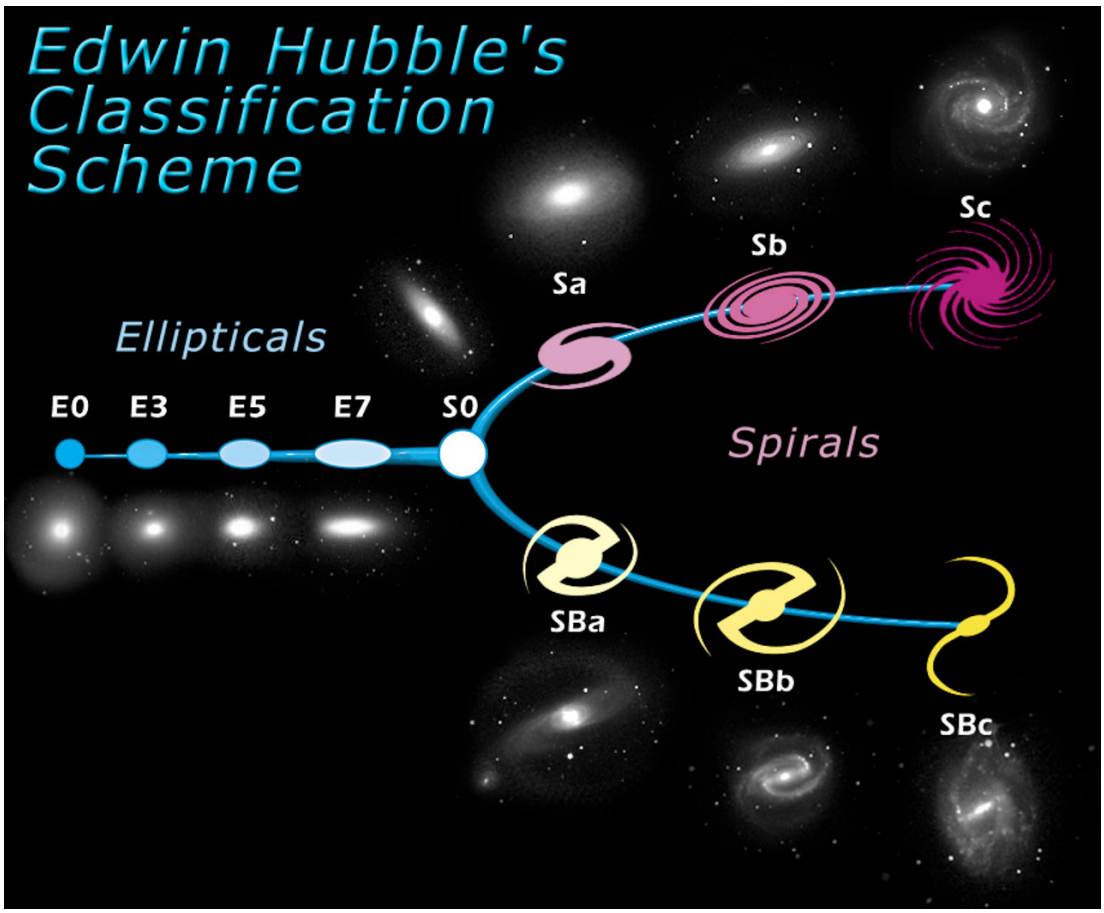

Figure 1.1 *Hubble tuning fork, showing ellipticals (E0 - E7), lenticulars (S0), spirals (Sa-Sc), and barred spirals (SBa - SBc). Image credit: ESA and NASA (https://www.spacetelescope.org/images/heic9902o/)*

1.1.1 Observed Characteristics

To understand the formation and evolution of galaxies, we must first understand the properties of those that we can observe. Galaxies can be divided by morphology into spirals, ellipticals, lenticulars, and dwarfs. These morphologies can be represented on the Hubble tuning fork, shown in Figure 1.1. Ellipticals from E0 to E7 become less spherical and more elongated, while the increasing letter classification of the spiral galaxies refers to how tightly wound the spiral arms are. Lenticulars (S0) share characteristics from both spirals and ellipticals. Galaxies can also be classified by their colour, mass, age, star formation history, and more. Here I briefly discuss these characteristics, and their relevance to the modelling techniques used in this work.

Classification

Spiral galaxies, also known as disk galaxies, are dominated by two structural features. The first is the rotationally supported disk of stars, neutral and molecular gas, and dust. They have mass from $10^9 M_\odot$ to $10^{12} M_\odot$. The mass fraction of gas varies significantly across the observed spiral population, from as little as 5% in the most massive spirals, to as much as 80% in the least massive (McGaugh & de Blok, 1997). The stars found here are typically young, metal-rich, with approximately circular orbits, and ongoing star formation within the disk (Martinsson et al., 2013). There is also a structurally separate thick disk component, where the gas has an exponential density profile away from the disk, with a characteristic scale height typically a factor three larger than the thin disk (Yoachim & Dalcanton, 2006). There are also spiral arm structures within the disk, most obvious when observing the young stars and the neutral and molecular gas (Nishiyama & Nakai, 2001). Second is the bulge of the galaxy, made of older, lower metallicity stars with randomised orbits. Within the bulge there can be stellar streams falling into the centre (Ferguson et al., 2002). The bulge is dispersion supported. Some spirals also have a bar structure that connects the bulge to the spiral arms (Erwin, 2019). The young population in the disk makes them appear blue, with more luminous galaxies appearing less blue. Lenticular galaxies have the disc structure of spiral galaxy, but with little or no star formation (van den Bergh, 2009).

Elliptical galaxies are completely bulge dominated, with smooth brightness profiles (Chitre & Jog, 2002). They contain very little cold gas or dust, and have star formation rates 2-5 time below that of spiral galaxies (Martig et al., 2013). There is evidence for significant amounts of hot gas (10^7 K), which is observed through the emission of x-ray radiation (Roberts et al., 1991; Mathews et al., 2003). Their populations are old and metal-rich. There is also an observable red sequence, with brighter elliptical galaxies appearing more red. Colour, here, is defined by the difference in magnitude between two photometric bands (Johnson, 1966). It was initially believed that these galaxies did not show significant rotational velocities, with the stars moving on random orbits. However, more recent observations have shown there to be populations of both fast and slow rotating ellipticals (Emsellem et al., 2011), and some exhibit rotational flattening (Mo et al., 2010). The most massive galaxies are ellipticals, with masses up to $10^{13} M_\odot$.

Dwarf galaxies do not fit well into these categories. Some show ongoing star formation, so called dwarf irregulars (Parodi & Binggeli, 2003), and some entirely quenched, the dwarf ellipticals (Geha et al., 2012). They also show significant variation, across several orders of magnitude, in both mass and luminosity (Begum et al., 2008). Dwarf irregulars have highly asymmetric structures, and can have very high gas fractions, with regions of gas extending far beyond their stellar population. Dwarf ellipticals have more regular structure, with metallicities lower than standard ellipticals. They are usually found in galaxy clusters (Mistani et al., 2015).

Modern cosmological simulations can reproduce this wide range of galaxy types, in some cases producing images that are almost indistinguishable from genuine observations (Vogelsberger et al., 2014b; Bottrell et al., 2017). The variety in galaxy properties points towards complex formation and evolution processes. These simulations allow us to probe the history and environment, that create the various types of galaxy, in ways that are impossible to do with observations. All the observed characteristics described above come from observations of baryonic matter, be it in the form of gas, stars, or dust. Reproducing the different classifications of galaxy in self-consistent simulations requires accurate modelling of their varied evolutions. This is only possible if the behaviour of the baryonic gas, which is the progenitor of the stars and dust, is modelled with great accuracy.

Large Scale Structure

The three dimensional distribution of galaxies reveals an overall ordered structure, on large scales. This structure is found by galaxy redshift surveys, such as the 2dF Galaxy Redshift survey (Colless et al., 2001), the Sloan Digital Sky Survey (Doroshkevich et al., 2004), the future observations with the Dark Energy Spectroscopic Instrument (DESI, 2016) or the upcoming Euclid survey (Racca et al., 2016). As we see in Figure 1.2, there are nodes where large numbers of galaxies are clustered, joined to one another by narrow filaments of galaxies. Between these filaments are sparse voids with few galaxies.

Within this large scale structure, many galaxies are found in distinct galaxy clusters. These are gravitationally bound groups of galaxies, with a number density several orders of magnitude greater than the universal average. The clusters are on the scale of a few megaparsecs, and contain tens of bright galaxies, with many smaller, less luminous companions. The high density of bright galaxies

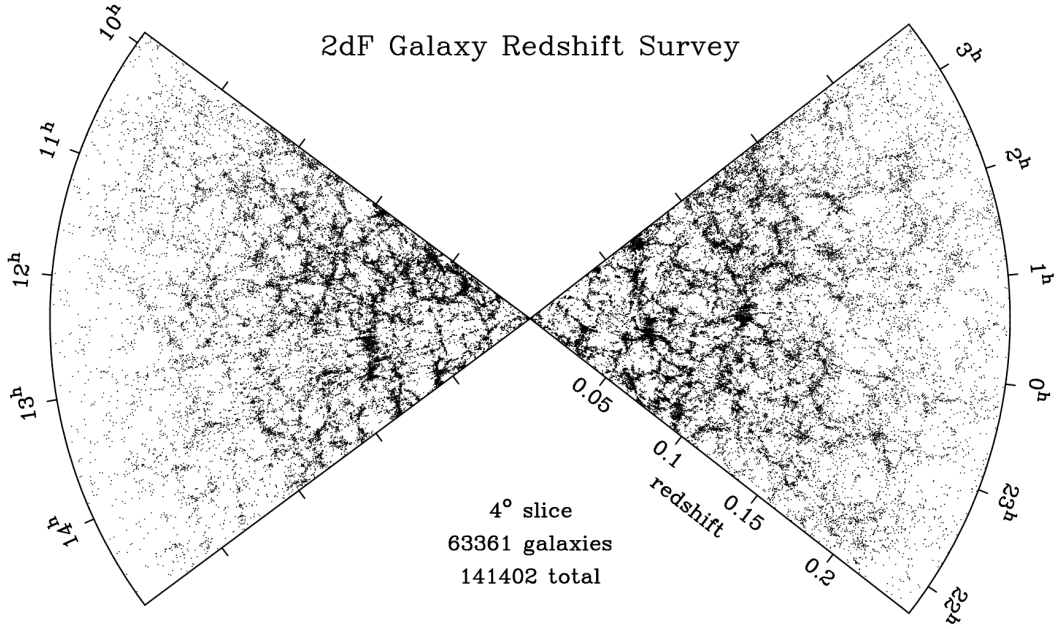

Figure 1.2 *Distribution of galaxies from 2dFGRS (Peacock et al., 2001)*

makes them observable at great distance, allowing them to be used to observe very high redshift. The strong ability of ΛCDM simulations to reproduce the observations of large scale structure is key evidence that this cosmology accurately describes our Universe (White & Rees, 1978).

Within this structure, gas exists across a huge range of densities. This is one of the most challenging aspects of simulating the behaviour of gas in this context. A simulation that encompasses a large segment of the structure at this scale must include a numerical approach capable of handling gas across this large dynamic range. This can be handled in a number of ways, some of which will be discussed in Section 1.2, but however it is done, accuracy must be maintained throughout the box.

1.1.2 Initial Conditions

The initial conditions, for galaxy formation, are derived from high precision observations of the cosmic microwave background (CMB), performed by space based telescopes such as WMAP (Spergel et al., 2003) and Planck (Planck Collaboration et al., 2016). The CMB is broadly isotropic, with a black body temperature of 3K, but also with fine anisotropies across different scales. Surveys look at temperature fluctuations in this light, which was emitted at

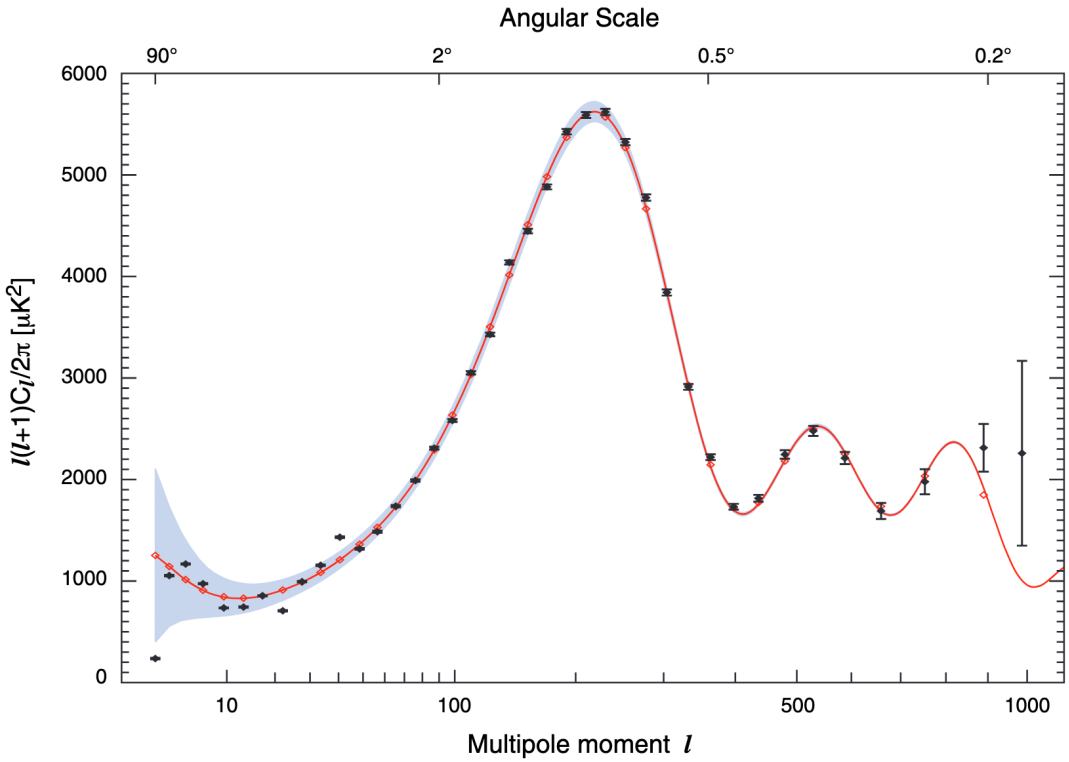

Figure 1.3 *Angular power spectrum of temperature fluctuations in the cosmic microwave background from WMAP (Hinshaw et al., 2007). Observations in black, best fit to ΛCDM cosmology in red.*

recombination, the point where the universe had expanded and cooled enough for ionized nuclei to combine with electrons, forming neutral atoms, creating a domain in which photons could travel large distances. Before this point, the physical conditions are such that photons could not travel far, before being absorbed. These temperature fluctuations in this surface are interpreted as fluctuations in the density of the universe at this time. The angular power spectrum of these fluctuations can be compared to theoretical predictions from cosmological models. Figure 1.3 shows the results from the WMAP satellite, with the best fit predictions from the standard ΛCDM cosmology. This provides strong evidence in favour of this cosmology.

The CMB power spectrum provides the distribution of the density perturbations to the cosmic average. These perturbations are the source of all structure formation within the Universe. Over-dense regions will attract the material around them slightly more than the average, and material will flow away from under-dense regions. This initial variation from the underlying isotropic distribution grows to form the large scale structure that we observe.

The density contrast $\delta = (\rho - \bar{\rho})/\bar{\rho}$ of any region is used to characterise its nature as an under- or over-density. The over-dense regions will increase in density over time, drawing material from their surroundings. As the universe expands, over-dense regions become more massive. The density of these regions is falling more slowly than the background thanks to the increased gravitational attraction, so the density contrast increases. The growth of the contrast takes the form of a power law, $\delta \propto t^{\alpha}$, where $\alpha > 0$ (Peebles, 1980). The exact form of the growth is dictated by the specific cosmology. Eventually the perturbation will have a density contrast of approximately unity. At this point, the turn around time, the over-dense region decouples from the universal expansion, stops expanding, and continues collapsing under gravity. Until this point, the growth in the over-density has been approximately linear, and the gas and DM density evolution does not differ significantly, because the DM potential dominates over the potential produced by the baryonic component.

Once a region has decoupled from the expansion of the universe, the dark matter component continues to collapse under gravity, forming a gravitationally bound structure known as a *halo*. The growth in the density becomes extremely non-linear. This collisionless material will eventually virialise through two body interactions, and the DM halo relaxes into an equilibrium state (Peebles, 1980). The DM collapse effectively halts once the random motion of the cold dark matter particles produces a distribution that is effectively supported by its own velocity dispersion.

The baryonic gas initially collapses along with the dark matter, as the pressure is negligible. As the density increases and the gas accelerates, the presence of these additional pressure forces leads to shocking, which significantly raises the temperature of the gas. This increased temperature produces pressure support, opposing the gravitational collapse. If the gas cannot cool efficiently, the gas will reach hydro-static equilibrium, with the DM component. However, if cooling is possible on suitably short timescales, then further collapse can occur. The instabilities in the gas structure that drive this early stage of galaxy formation require detailed modelling, reinforcing the need for advanced numerical techniques that can handle complex multi-dimensional flows.

1.1.3 Gas Cooling

A number of cooling mechanisms exist, with the physical conditions and chemical makeup of the gas dictating which mechanism dominates the cooling that the medium can experience. In order to cool, the gas must be able to lose energy. For cosmic gas, this is only achievable if it can radiate energy away. So, if a gas is to cool, it must be able to produce photons, and those photons must be able to escape the local environment.

Radiative Cooling

The process of energy escaping a region in the form of electromagnetic radiation is known as radiative cooling. In the context of galaxy formation, two-body interactions are the most relevant. After the first galaxies form, much of the gas in the Universe heats to the virial temperature of the halos into which they are collapsing. If this virial temperature is between 10^4K and 10^7K, atomic cooling allows the gas to cool to the 10^4K threshold (Somerville & Davé, 2015). This is where collisionally ionised and excited atoms release photons through recombination and decay. The cascade of transitions from ionised or excited state, to the ground level, release photons at wavelength for which the gas is optically thin. The collisional excitation and ionisation of these atoms occurs through two-body interactions.

For the gas to cool below the $T = 10^4$K threshold, cooling is only efficient if there are molecules, metals, or dust present. Atomic cooling alone is no longer efficient in this regime. There are not enough collisional excitations and ionisations to produce a significant number of escaping photons. As the first stars and galaxies form, the primordial metallicity does not provide enough metals for metal line cooling to be efficient. Metals provide many electron transitions, which can be excited and de-excited collisionally, to radiate energy from the gas. In their absence, the gas must instead cool by the collisional excitation and de-excitation of molecules of the primordial species, most notably hydrogen H_2. However, without any metals, which can catalyse the formation of molecular hydrogen, these molecules can only form through two-body interactions. The rate of formation is proportional to the square of the density, which can only increase if the gas cools, which can only happen with more molecules. Enough molecular hydrogen must form to allow the gas to cool to form the first stars, before $z \approx 6$,

when almost all the gas in the Universe is observed to have been reionised, most likely in part by the first stars. To solve this complex puzzle, the correct cooling physics must be combined with an accurate modelling of the gas. No matter which cooling channel is used, as the gas cools it will become unstable, and the accuracy with which the gravo-hydrodynamics of the resultant instability can be modelled, determines the accuracy of the end product, the galaxies we seek to understand.

Cooling Function

The cooling rate from each process can be calculated separately. However, it is useful to know the total cooling rate for some region of gas at a given temperature, independent of the density. This is expressed as a cooling function,

$$\Lambda(T) = \frac{r}{n_H^2}, \tag{1.1}$$

with the volume averaged cooling rate r, and the number density of hydrogen atoms n_H. This cooling function can be calculated for both primordial gas and gas that has been enriched with metals. The metallicity of the gas changes the form of the cooling function. Metal cooling can enhance the ability of a cloud to cool below 10^4K.

For primordial gas, the cooling function drops off rapidly below 10^4K if we only consider neutral and ionised hydrogen and helium. At this temperature the medium is almost entirely neutral, and the internal energy is not high enough to allow efficient cooling by collisional excitation. Molecular hydrogen H_2 is required to efficiently cool below this temperature (Anninos & Norman, 1996). For this to form, in the early universe, there must be interactions between neutral and negatively ionised hydrogen H^- (Abel et al., 1997). H_2 production therefore depends on the density and temperature of the gas, but also the ionisation fraction.

When the dynamical time is longer than the sound speed crossing time of the halo, any collapse will be halted as pressure equilibrium is reached throughout the cloud (O'Shea & Norman, 2007). If the opposite is true, then the new equilibrium conditions cannot be reached fast enough, and the gas collapses further, eventually becoming self gravitating. It can then collapse independently of the underlying dark matter distribution, eventually forming dense clouds of gas. These clouds

are the location of eventual star formation.

1.1.4 Star Formation and Feedback

As the gas in the early Universe cools by the processes described above, clouds of gas collapse under gravitational instabilities, eventually forming stars. Stars provide the majority of the light by which we observe the Universe. They reside within, and are a key component of, the galaxies which we seek to understand. A wide range of stellar types have been observed, with varying mass, metallicity and age. The details of star formation form an entire field of study by themselves, with individual star formation happening on a scale far below that which can be resolved in large scale galaxy formation simulations. However, the impact of star formation on the evolution of its host galaxy is significant. The formation of stars removes material from the ISM, and then produces the metals that drive further cooling and star formation. Outflows from stars shape the gas around them, resulting in a clumpy, turbulent medium. At the end of their lifetimes, massive stars explode as supernovae, injecting huge amounts of energy into the surrounding medium. While large scale galaxy formation simulations cannot resolve the details of the formation, and lifetimes, of individual stars, the wider impact of the stars can be included.

A number of process exist whereby the stellar population of a galaxy can radically transform the interstellar and circumgalactic medium (ISM and CGM respectively). There are three formats that this feedback takes: mass loading, electromagnetic radiation, and neutrino emission. Over the course of a star's lifetime, it emits a certain amount of energy as electromagnetic radiation. The amount is dependent primarily on the mass of the star. The mass dictates the amount of fuel available for fusion, the source of a star's energy. It is the most massive stars that dominate this energetic feedback. Some fraction of this radiation can be absorbed by the surrounding medium, depositing energy and momentum into the ISM. The radiation propagation within a star can also eject material from the star itself in the form of a stellar wind. This adds mass, alongside momentum and energy, to the medium.

At the end of their lifetimes, the most massive stars ($> 8M_{\odot}$) collapse into a supernova. These are extremely short lived events that emit huge amounts of energy. They have been shown to emit of order 10^{51}ergs , equivalent to approximately the total energy output of a solar mass star over its main-sequence

lifetime (Woltjer, 1972). A large fraction of this emission is in the form of a burst of neutrinos. The explosion also ejects a large fraction of the sources mass. This scenario includes Type Ib, Type Ic and Type II supernovae (Filippenko, 2005; Woosley & Janka, 2005). Supernovae can also be caused by the accretion of mass from a binary companion onto a white dwarf, a Type Ia supernova (Hillebrandt & Niemeyer, 2000).

Without the energy provided by these stars, simulations predict that the gas within halos cools rapidly, collapsing under gravity to form stars at rates far greater than those observed (Zamora-Aviles et al., 2012). The Kennicutt-Schmidt relation gives us a relationship between gas surface density and star formation rate. Together these show that the star formation efficiency is very low, of the order of a few percent. These same simulations show that, without feedback, the cold gas fraction in spiral galaxies is too high, so that even if stars are prevented from forming, there is too much gas resting in the galaxy. The collapse of cold gas on the scale of individual clouds requires an additional process to disrupt it, and material must be ejected from the galaxy itself.

Stellar Winds

In the context of galaxy evolution, only the winds from the most massive stars need be considered. While many classes of star produce winds, massive stars produce the most energetic, and so dominate the feedback effects (Crowther, 2001). The energy contribution from a stellar wind can be characterised by its mechanical luminosity

$$L_W = \frac{1}{2}\frac{dM_W}{dt}v_W^2, \tag{1.2}$$

using the mass loss rate of the star dM_W/dt, and the velocity of the wind v_W. Winds sweep up material around the source, creating hot diffuse bubbles, with dense shells surrounding them. The shape of these shells is highly dependent on the medium into which they are blown. The highly clumpy nature of GMCs and the ISM in general mean these wind blown bubbles follow the low density channels, with less significant disruption of the dense clumps. These winds can act to reduce and destroy star forming regions in the vicinity of their source. Conversely, they can also have a smaller positive feedback effect on the star formation rate, as the material that is swept into the dense shell can become unstable to collapse.

Supernovae

Supernovae are among the most intense events that have ever been observed, emitting extremely brightly for a very short period of time, of the order of a few weeks (Riess et al., 1999). There are a number of formation mechanisms that lead to the supernova explosion, but they share similar final stages and products. Current models suggest a supernova can be caused by either the collapse in the final stages of a massive star's life, or by the accretion of material onto a lower mass white dwarf from a binary companion. In either case, supernovae dump vast amounts of energy very rapidly into the ISM.

Other than the dramatic effect they have in disrupting the material around them, supernovae can also act to drive galactic winds. Simulations have shown these winds drive the outflows that are required to either remove material from the galactic disk, or to prevent it from accreting in the first place (Hopkins et al., 2012). Supernovae are a primary source of these galactic outflows.

Radiative Feedback

Stars are obvious sources of electromagnetic radiation. This radiation interacts with the surrounding medium, with the key effect in the context of galaxy evolution coming from ionising radiation. Massive stars produce high levels of radiation energetic enough to ionise the surrounding hydrogen. The material between galaxies, the intergalactic medium (IGM), is neutral at the time of recombination, but is completely ionised by $z \approx 6$ (Fan et al., 2006). Since star formation, and so galaxy evolution, is inextricably linked to the amount of neutral and molecular hydrogen, our models require sources of ionising radiation that can effectively ionise a huge volume of space. Massive stars are obvious candidates (Simon-Diaz & Stasinska, 2008).

Photons with high enough energies ionise neutral atoms when they are absorbed. The freed electron has kinetic energy equal to the excess energy photon, after the ionisation energy has been used up. This is how the ionising radiation both ionises and heats the surrounding medium. The increase in temperature and reduction of the neutral fraction can suppress the ability of the surrounding gas to cool, shutting down potential star formation.

The effects of this ionising radiation, and the other processes described above,

contribute to a medium, within and around galaxies, that has huge variation in density and temperature, with complex interactions between flows in chaotic orientations. These complex flows contain many unstable structures, which must be accurately captured by the numerical methods used to model them.

1.2 Numerical Modeling of Galaxy Formation

As we have seen above, galaxies are highly complex systems, that are embedded within larger cosmological structures, with a wide range of masses, morphologies, and compositions. The physics that governs their formation and evolution is equally complex, with different processes dominating across the large variation in scale. Our understanding of how these processes combine to produce the observed properties of galaxies have been greatly informed by numerical simulations. Instead of trying to solve the underlying physical equations analytically, which is an impossible task for all but the simplest, most idealised, scenarios, numerical approximations of the solutions to these equations are used to test our understanding of the observed galaxy properties, and to make testable predictions for future observations.

Numerical solutions, to astrophysical problems, have been utilised for many decades (Efstathiou et al., 1985; Springel, 2005; Bryan et al., 2014). The requirements of ever more detailed physics simulations have been one of the driving forces behind the development of larger and larger computational machines. When one sets out to produce a numerical approximation of the solution to a physical solution, there are several competing factors that must be considered in choosing an appropriate method. First of all, the method must be accurate. If the numerical solution does not come close to reproducing the true solution, then it is obviously of no use. Different numerical methods will often recover different facets of a solution with varying accuracy, so choice of method is often dictated by what behaviours are being investigated. This is often in conflict with the second factor: the computational cost of the method. If a method is extremely accurate, but also extremely expensive, it is potentially of much less use than a less accurate method that can be run at a fraction of the computational cost. The trade off between these two factors is at the heart of the choice of numerical method.

A third factor follows somewhat from the first two. Based on the choice

of numerical method, and the computational resources available, a choice of resolution is effectively made, be this resolution in space, mass, frequency, temperature, etc. The resolution dictates the detail with which the solution can be recovered. A finer resolution allows for more detailed structure to be recovered, but can significantly increase computational cost. The increase comes from both having more fluid elements to track and update, and from the time-step required by the numerical method being smaller (see Sections 1.2.3 and 1.2.4 for details of time-step limitations). Any increase in cost with resolution is most acute with 3D models, since where increasing the number of cube cells on an edge will increase the total number by the same factor cubed. For this reason, it is critical to consider the computational efficiency of any method one is considering using.

Alongside more complex structure recovery, increasing resolution can also change what physical processes must be modelled. Assumptions made about the internal structure and properties of a gas, that are acceptable when the resolution is at kilo-parsec scale, may no longer produce accurate results when used with single digit parsec resolution. Turbulent flows are a good example. On large scales the turbulence could simply be smoothed out into a contribution to internal energy, but a finer resolution might need to be able reproduce the turbulence itself. This idea also brings us to what processes are modelled by sub-grid algorithms. These are models that emulate processes not handled self consistently by solving the fundamental physical equations. Star formation is an obvious example. No cosmological or galaxy formation simulation can hope to follow all the complex physics of star formation. Instead a star formation sub-grid model is applied, which uses the information provided by the base solver to estimate how star formation would occur. This is just one example of many, but they all share the idea of smoothing over complex physics that cannot realistically be included, and instead provide estimates of the major effects of the desired process.

All of these ideas play a role in the development and implementation of the various numerical methods required for running accurate galaxy formation simulations. In this section, I describe the fundamental equations that must be understood if one is to model the core physics of dark matter and gas dynamics. I cover the initial conditions used in galaxy formation simulations, and the approach used to generate them. This is followed by a description of a number of numerical methods for solving the physical equations, particularly those widely used in computational astrophysics, which includes both hydrodynamics and gravity

solvers. Finally I outline some of the approaches used to include sub-grid physics in these simulations.

1.2.1 Fluid Equations

Here I derive and discuss the fundamental equations that describe the behaviour of the different media. This includes the Boltzmann equation, the moments of the Boltzmann equation, which form the fluid equations, and the equations of motion. I briefly cover the equivalent equations in comoving coordinates, as these coordinates can be used to encode the expansion of the Universe, within which simulations of large scale galaxy formation are run.

Boltzmann Equation

One key component of most astrophysical systems is baryonic gas, made up of many billions of moles of particles. It is inconceivable that we could calculate the motion and state of each particle individually. Instead, we use fluid elements, which track the statistical properties of the gas, with each element representing an ensemble of physical particles. For this method to work, the size of each fluid element l should be significantly larger than the mean distance between particle collisions λ, or $l \gg \lambda$. To find the overall statistical behaviour, one defines the particles in phase space, taking a three dimensional body of N identical gas particles of mass m, with positions $\mathbf{x}_i$ and velocities $\mathbf{u}_i$, splitting by dimensions $i = (0, 1, 2)$. This defines the conditions of a particle in six dimensional phase space $(x_1, x_2, x_3, u_1, u_2, u_3)$. One also takes the specific force $\mathbf{F}_i$ acting upon each particle. These give us the evolution of the particle positions, where the change in the position $d\mathbf{x}_i$ of particle i, over time dt, is simply

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{u}_i, \tag{1.3}$$

and velocities

$$\frac{d\mathbf{u}_i}{dt} = \mathbf{F}_i(\mathbf{x}_i, \mathbf{u}_i, t), \tag{1.4}$$

with time t. The particles have momentum $\mathbf{q}_i = m\mathbf{u}_i$, and bulk velocity $\mathbf{v}$, where the bulk velocity refers to any general motion that all particles share. We define a distribution function that describes this body of gas in phase space such that the

number of particles dN, in the volume of phase space, within $\mathbf{x} + \mathbf{dx}$ and $\mathbf{u} + \mathbf{du}$ is given by

$$dN = f(\mathbf{x}, \mathbf{u}, t)\mathbf{dxdu}. \tag{1.5}$$

If this volume element contains many particles, effectively required by the mean distance condition, then a statistical approach is viable. If this is not the case, then a larger fluid element must be used. Having too few particles in the element results in the conditions within being distorted by extreme outliers. The change in this distribution function over time dt is simply the change in the positions $\mathbf{x}+\mathbf{u}dt$ and velocities $\mathbf{u}+\mathbf{F}dt$ of these particles. These changes are the trajectories of the particles in phase space.

From Liouville's theorem, we know that for a Hamiltonian system, such as this one, the phase space volume occupied by a set of particles does not change as they move along their phase space trajectories (Taylor, 2005; Bodenheimer et al., 2006). This assumes that no particles are are created or destroyed, and that there are no collisions between particles. While the volume that they occupy does not change, the shape of the volume can deform. The number of particles in the distribution function at time t and $t + dt$ is constant. We can put a corollary on this statement; that collisions between particles can create discontinuous changes in phase space, as a particle might change velocity without changing position. If one considers the exact moment two particles collide, both will move to new velocity coordinates as they deflect one another, but no time has passed. The effect of collisions is not captured by the continuous trajectories of the particles, so the effects of these collisions are included as Δf_{coll}. We can summarise these statements as

$$f(\mathbf{x} + \mathbf{u}dt, \mathbf{u} + \mathbf{F}dt, t + dt) - f(\mathbf{x}, \mathbf{u}, t) = \Delta f_{\text{coll}}, \tag{1.6}$$

which can be written in partial derivative form as the Boltzmann Equation

$$\frac{\Delta F}{\Delta t} = \frac{\partial f}{\partial t} + \frac{\partial x_i}{\partial t}\frac{\partial f}{\partial x_i} + \frac{\partial u_i}{\partial t}\frac{\partial f}{\partial u_i} = \frac{\partial f}{\partial t} + u_i\frac{\partial f}{\partial x_i} + F_i\frac{\partial f}{\partial u_i} = \left[\frac{\partial f}{\partial t}\right]_{\text{coll}}, \tag{1.7}$$

with implied summation over indices i. This effectively describes the evolution, with time, of this distribution function, in phase space. We can define a number of physical quantities from this distribution function. The number density of

particles

$$n(\mathbf{x}, \mathbf{t}) = \int f(\mathbf{x}, \mathbf{u}, t) d\mathbf{u} \tag{1.8}$$

and mass density

$$\rho(\mathbf{x}, t) = m \int f(\mathbf{x}, \mathbf{u}, t) d\mathbf{u} = mn \tag{1.9}$$

at position $\mathbf{x}$, naturally follow by integrating over all velocities $\mathbf{u}$, as does the bulk velocity

$$\mathbf{v}(\mathbf{x}, t) = \frac{m}{\rho} \int \mathbf{u} f(\mathbf{x}, \mathbf{u}, t) d\mathbf{u}. \tag{1.10}$$

We also define the random peculiar velocity, relative to the bulk velocity, as

$$\tilde{\mathbf{u}} = \mathbf{u} - \mathbf{v}. \tag{1.11}$$

These peculiar motions contribute to the internal energy of the gas, and are used to define the specific internal energy ϵ

$$\epsilon(\mathbf{x}, t) = \frac{m}{2\rho} \int \tilde{u}^2 f(\mathbf{x}, \mathbf{u}, t) d\mathbf{u}. \tag{1.12}$$

If we assume the particles are at equilibrium, then the time derivative terms disappear. We can then use the Maxwellian velocity distribution (Landau et al., 1980)

$$f(\mathbf{x}, \mathbf{u}, t) d\mathbf{u} = n(\mathbf{x}, t) \left(\frac{m}{2\pi k_B T(\mathbf{x}, t)} \right)^{3/2} \exp\left(-\frac{m(\mathbf{u} - \mathbf{v})^2}{2 k_B T(\mathbf{x}, t)} \right) \tag{1.13}$$

as the suitable distribution function, where k_B is the Boltzmann constant and T is temperature. This describes the particles per unit volume at position $\mathbf{x}$ with velocities from $\mathbf{u}$ to $\mathbf{u} + \mathbf{du}$. This statistical description of the set of identical gas particles provides the framework from which we can derive the equations that describe the conservation of mass, momentum and energy as the gas evolves.

Fluid Equations - Eulerian Form

We can use the above formulation to consider the state of a fluid at a fixed position $\mathbf{x}$. The fluid equations, for this case, are derived by taking the first three

velocity moments $k = (0, 1, 2)$ of the Boltzmann equation. In other words, we are finding out how the Boltzmann equation dictates the motion of an ensemble of fluid elements, by expanding it about the motion of the fluid. The motion is described by the dependence on the velocity terms of various orders. The kth moment is found by multiplying Equation (1.7) with the appropriate velocity term $U_k = (1, \mathbf{u}, u^2)$, and integrating over all velocities $\mathbf{u}$

$$\int U_k \left[\frac{\partial f}{\partial t} + u_i \frac{\partial f}{\partial x_i} + F_i \frac{\partial f}{\partial u_i} \right] d\mathbf{u} = \int U_k \left[\frac{\partial f}{\partial t} \right]_{\text{coll}} d\mathbf{u}. \tag{1.14}$$

The integration over all velocities, at a fixed position, is the key factor that dictates the Eulerian nature of the equations that will result from this approach.

One can first consider the collisional part of the Boltzmann Equation, to observe some important relationships. If one assumes that the particles are neither destroyed nor created in collisions, then the integral of the collisional term alone (i.e. the $0th$ order term of Equation (1.14)), over all velocities, must be zero. Collisions have not changed the total number of particles, just their velocities. If one also assumes that the total momentum vector of the particles is conserved, then the first moment in each dimension is also zero, and the same goes for the second moment, if total energy conservation is assumed. The right hand side (RHS) of Equation (1.14) therefore vanishes for all moments. It can also be shown that (Bodenheimer et al., 2006)

$$\int \frac{\partial f}{\partial u_i} d\mathbf{u} = 0, \quad \int u_j \frac{\partial f}{\partial u_i} d\mathbf{u} = -\delta_{ij} \frac{\rho}{m} \quad \text{and} \quad \frac{1}{2} \int u^2 \frac{\partial f}{\partial u_i} d\mathbf{u} = 0. \tag{1.15}$$

The first conservation equation is found by multiplying both sides of the $0th$ velocity moment of (1.14) by the particle mass m, to give

$$m \int \frac{\partial f}{\partial t} d\mathbf{u} + m \int u_i \frac{\partial f}{\partial x_i} d\mathbf{u} + m F_i \int \frac{\partial f}{\partial u} d\mathbf{u} = 0. \tag{1.16}$$

We can see from (1.15) that the third term vanishes. The partial derivatives with respect to t and x_i are independent of $\mathbf{u}$, so they can be moved outside the integral. This allows us to substitute in (1.9) and (1.10). Together (1.16) reduces to

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho v_i) = 0, \tag{1.17}$$

which describes the conservation of mass in some volume element dV. This

conservation can be seen qualitatively by observing that the change in density at a given position, over time, is equal to the net flux of material at that position. Any material that flows away reduces the density, and incoming material increases it. The change with time, combined with the flux of material, balance out, so mass is conserved across some volume. Explicitly, by integrating over a volume V, and using the divergence theorem, the equation becomes

$$\int_V \frac{\partial \rho}{\partial t} dV = -\int_V \frac{\partial}{\partial x}(\rho v_i) dV = \int_S \rho v_i \hat{n}_i dA = 0, \tag{1.18}$$

where S is the surface of the volume V, the infinitesimal area element of this surface is dA, and $\hat{\mathbf{n}} = (\hat{n}_1, \hat{n}_2, \hat{n}_3)$ is the unit normal of the area element. The surface is closed, hence the integral of the flow through it producing zero. This is true for a suitably small volume. The integral over the temporal part is all that is left, with

$$\int_V \frac{\partial \rho}{\partial t} dV = \frac{\partial}{\partial t} \int_V \rho dV = \frac{\partial M}{\partial t} = 0. \tag{1.19}$$

The mass enclosed within the volume does not change with time, so mass is conserved. This does not preclude the shape of the volume element changing with time, so the density can still change, but mass is still conserved.

The same procedure is applied for $k = 1, 2$, multiplying Equation (1.14) by m for both. In the $k = 1$ case, this becomes

$$\frac{\partial}{\partial t}(\rho v_i) + \frac{\partial}{\partial x_i} \int m u_i u_j f d\mathbf{u} - \rho F_i = 0, \tag{1.20}$$

with the third term transformed using the second equation of (1.15). We can substitute the peculiar velocity for the velocity in the second term using (1.11). We also need to use

$$\int \tilde{\mathbf{u}} f d\mathbf{u} = \int \mathbf{u} f d\mathbf{u} - \int \mathbf{v} f d\mathbf{u} = \frac{\rho}{m}\mathbf{v} - n\mathbf{v} = 0, \tag{1.21}$$

so any term with this as part of it will vanish. This is used to transform the second term of (1.20) into

$$\int m u_i u_j f d\mathbf{u} = \rho v_i v_j + P_{ij}, \tag{1.22}$$

where we introduce the pressure tensor. In the case of interstellar or circum/intergalactic gas, as in many other astrophysical scenarios, the pressure can be

assumed to be isotropic, i.e. acting equally in all directions. This is true for an ideal gas, where the energy that contributes to the pressure, the internal energy, is equally distributed across velocities in each dimension. The pressure tensor then reduces to

$$P_{ij} = \int m\tilde{u}_i\tilde{u}_j f d\mathbf{u} = P\delta_{ij}. \tag{1.23}$$

This pressure P is then given by

$$P = \frac{1}{3}\int m\tilde{u}^2 f d\mathbf{u} = \frac{2}{3}\rho\epsilon. \tag{1.24}$$

This leaves us with

$$\frac{\partial}{\partial t}(\rho v_i) + \frac{\partial}{\partial x_j}(\rho v_i v_j + P) = \rho F_i, \tag{1.25}$$

which describes the conservation of momentum in each dimension i=(0,1,2).

Finally, we can solve for the second. We restart from (1.14), multiplying the $k = 2$ case by mass m. Combined with the third equation from (1.15), we end up with

$$\frac{\partial}{\partial t}\int \frac{1}{2}mu^2 f d\mathbf{u} + \frac{\partial}{\partial x_i}\int \frac{1}{2}mu^2 u_i f d\mathbf{u} - \rho v_i F_i = 0. \tag{1.26}$$

Once again we substitute the velocity with the peculiar velocity (1.11), and use (1.21) to eliminate the terms that depend linearly on $\tilde{\mathbf{u}}$. This gives us

$$\frac{\partial}{\partial t}\int \frac{1}{2}m\tilde{u}^2 f d\mathbf{u} + \frac{\partial}{\partial t}\int \frac{1}{2}mv^2 f d\mathbf{u} + \frac{\partial}{\partial x_i}\int \frac{1}{2}m\left(\tilde{\mathbf{u}} + \mathbf{v}\right)^2(\tilde{u}_i - v_i) f d\mathbf{u} - \rho v_i F_i = 0. \tag{1.27}$$

We can substitute (1.12) into the first term, and (1.9) into the second. The third term can be expanded to

$$\frac{\partial}{\partial x_i}\int \frac{1}{2}m\left(\tilde{u}^2\tilde{u}_i + 2\tilde{\mathbf{u}}\mathbf{v}\tilde{u}_i + \tilde{u}^2 v_i + v^2 v_i\right) f d\mathbf{u}, \tag{1.28}$$

again using (1.21) to discard terms. This reduces to

$$\frac{\partial}{\partial x_i}\left[\int \frac{m}{2}\tilde{u}^2\tilde{u}_i f d\mathbf{u} + \mathbf{v}\int m\tilde{\mathbf{u}}\tilde{u}_i f d\mathbf{u} + \int \frac{m}{2}\tilde{u}^2 v_i f d\mathbf{u} + \int \frac{m}{2}v^2 v_i f d\mathbf{u}\right], \tag{1.29}$$

where the first term in the bracket is defined as the conduction heat flux h_i,

the second term can be substituted with (1.23) to be Pv_i, the third term is $\rho v_i \epsilon$ using (1.12), and the fourth becomes $v^2 \rho v_i/2$ using (1.10). Heat conduction is negligible for astrophysical problems, since we are modelling diffuse gases where the speed of individual particles within the gas is not large when compared to the size characteristic length scale of the gas. The rate at which particles can carry energy around is therefore negligible, so this term can be ignored. A notable exception is heat transport within some stars, such as white dwarfs (Bodenheimer et al., 2006), but this does not need to be considered in the context of gases on galaxy evolution scales.

We reintroduce this reduced form into (1.26), giving

$$\frac{\partial}{\partial t}\left[\rho\left(\frac{v^2}{2}+\epsilon\right)\right]+\frac{\partial}{\partial x_i}\left[\rho v_i\left(\frac{v^2}{2}+\epsilon\right)\right]=-\frac{\partial}{\partial x_i}(Pv_i)-\rho v_i F_i, \tag{1.30}$$

which can be simplified using the total specific energy

$$E=\frac{1}{2}v^2+\epsilon, \tag{1.31}$$

leaving us with

$$\frac{\partial}{\partial t}(\rho E)+\frac{\partial}{\partial x_i}(\rho E+P)v_i=\rho v_i F_i. \tag{1.32}$$

This describes the conservation of total energy, and forms a key part of the equations governing the behaviour of a fluid. It can be used to derive equivalent equations that describe the evolution of the kinetic and potential components.

The evolution of the kinetic energy can be found by subtracting the conservation equation, multiplied by component wise velocity v_i, from the momentum equation

$$\frac{\partial}{\partial t}(\rho v_i)+\frac{\partial}{\partial x_i}(\rho v_i v_j+P)-v_i\frac{\partial \rho}{\partial t}-v_j\frac{\partial}{\partial x_i}(\rho v_i)=\rho F_i, \tag{1.33}$$

where the expansion of the various terms produces

$$\rho\frac{\partial v_i}{\partial t}+v_i\frac{\partial \rho}{\partial t}+\rho v_i\frac{\partial v_j}{\partial x_i}+\rho v_j\frac{\partial v_i}{\partial x_i}+v_i v_j\frac{\partial \rho}{\partial x_i}+\frac{\partial P}{\partial x_i}-v_i\frac{\partial \rho}{\partial t}-\rho v_j\frac{\partial v_i}{\partial x_i}-v_i v_j\frac{\partial \rho}{\partial x_i}=\rho F_i, \tag{1.34}$$

which in turn cancels to

$$\rho\frac{\partial v_i}{\partial t}+\rho v_i\frac{\partial v_j}{\partial x_i}=\rho F_i-\frac{\partial P}{\partial x_i}. \tag{1.35}$$

This is essentially a rearrangement of the momentum conservation equation, but can be multiplied again by v_i, and combined with the continuity equation multiplied by the specific kinetic energy $v^2/2$, which has the form

$$\rho v_i \frac{\partial v_i}{\partial t} + \rho v^2 \frac{\partial v_j}{\partial x_i} + \frac{1}{2} v^2 \frac{\partial \rho}{\partial t} + \frac{1}{2} v^2 \frac{\partial}{\partial x_i}(\rho v_i) = \rho v_i F_i - v_i \frac{\partial P}{\partial x_i} \tag{1.36}$$

Expanding this via the product rule for the third term, and then grouping terms by their derivative dependence, allows these terms to be combined, producing a form that describes the evolution of kinetic energy only

$$\frac{\partial}{\partial t}(\rho v^2) + \frac{\partial}{\partial x_i}(\frac{1}{2}\rho v^2 v_i) = \rho v_i F_i - v_i \frac{\partial P}{\partial x_i}. \tag{1.37}$$

The above equation is known as the bulk energy equation, as it describes the kinetic terms, or the energy in the macroscopic motion of the gas. The evolution of the internal motion of the gas, which produces the internal energy, is found by subtracting the bulk energy equation from the total energy equation. This is effectively removing the kinetic components to leave only those related to the internal energy. The combined equations take the form

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_i}(\rho E + P) v_i - \frac{\partial}{\partial t}(\rho v^2) - \frac{\partial}{\partial x_i}(\frac{1}{2}\rho v^2 v_i) - v_i \frac{\partial P}{\partial x_i} = 0. \tag{1.38}$$

Substituting Equation (1.31), for the total specific energy e, allows us to cancel out the kinetic terms of each derivative, leaving only

$$\frac{\partial}{\partial t}(\rho \epsilon) + \frac{\partial}{\partial x_j}(\rho v_j \epsilon) = -P \frac{\partial v_j}{\partial x_j}, \tag{1.39}$$

which is clearly analogous to the continuity equation, with the addition of a source term. Together with the bulk energy and total energy equations, these three equations describe the evolution of the energy of a fluid.

To summarise the above derivations, we now have equations that describe the conservation of mass, momentum and energy for a fluid that can be broken down into fluid elements, where each element is made of an ensemble of physical particles. Specifically, the equations trace the Eulerian evolution of the fluid, keeping the tracer position fixed and following the flow of fluid past these positions. These equations can be rewritten compactly as

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{Q}) = \mathbf{S}, \tag{1.40}$$

where

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho\mathbf{v} \\ \rho E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho\mathbf{v} \\ \rho\mathbf{v}^2 + P \\ (\rho E + P)\mathbf{v} \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 0 \\ \rho\mathbf{F} \\ \rho\mathbf{v}\mathbf{F} \end{pmatrix}. \tag{1.41}$$

This set of partial differential equations describe the evolution of the fluid, but are dependent on a specific relationship between the pressure and the state variables of the system, via the equation of state. For an ideal gas, with adiabatic gas constant γ, this can be written as

$$E = \frac{1}{2}(\mathbf{v} \cdot \mathbf{v}) + \frac{P}{\rho(\gamma - 1)}. \tag{1.42}$$

The RHS of the fluid equations are often described as source and sink terms, as they represent the change in momentum and energy created by additional forces. In the astrophysical context, this will often include gravity, as well as various sub-grid physical processes, such as star formation feedback. Many numerical hydrodynamics solvers are built out of these equations, some of which I describe in more detail in Section 1.2.3. These methods are used to model a wide variety of astrophysical scenarios, and have seen extensive development across several decades.

Fluid Equations - Lagrangian Form

The above formulation of the conservation equations describes the evolution of the fluid state at a fixed point in space. A number of numerical methods instead follow the motion of a fluid element as it moves with the flow. In this approach, the fluid equations are recast with comoving coordinates. For this we use the instantaneous position, using components of the radius vector $\mathbf{r} = (r_1, r_2, r_3)$. The Lagrangian time derivative defines the change with time of any representative variable of the fluid element, as it moves within the simulated region. The definition of this derivative is given as

$$\frac{dQ}{dt} = \lim_{\delta t \to 0} \frac{Q(\mathbf{x} + \mathbf{v}\delta t, t + \delta t) - Q(\mathbf{x}, t)}{\delta t}. \tag{1.43}$$

The rate of change in the instantaneous position $\mathbf{r}$ is equivalent to the velocity of the gas at that position. The first order Taylor expansion of the change in the

fluid quantity Q is simply $Q(\mathbf{x} + \mathbf{v}\delta t, t + \delta t) = Q(\mathbf{x}, t) + (\partial Q/\partial t)\delta t + v_j(\partial Q/\partial x_j)\delta t$, where, as before, the index j includes the implied summation over all dimensions. Thus the Lagrangian derivative can be written in terms of the Eulerian derivatives

$$\frac{dQ}{dt} = \frac{\partial Q}{\partial t} + v_j \frac{\partial Q}{\partial x_j}. \tag{1.44}$$

Going through the fluid variable vector $\mathbf{Q}$, we can derive the Lagrangian form of Equation (1.40). The Lagrangian derivative of the density is

$$\frac{d}{dt}\rho(\mathbf{r}, t) = \frac{\partial \rho}{\partial t} + v_j \frac{\partial \rho}{\partial x_j}. \tag{1.45}$$

The product rule gives

$$\frac{\partial}{\partial x_j}(\rho v_j) = v_j \frac{\partial \rho}{\partial x_j} + \rho \frac{\partial v_j}{\partial x_j}. \tag{1.46}$$

Combining these we get

$$\frac{d\rho}{dt} = \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho v_j) - \rho \frac{\partial v_j}{\partial x_j}, \tag{1.47}$$

where the first and second terms of the RHS are shown to be zero by Equation (1.17). Thus we have

$$\frac{d\rho}{dt} + \rho \frac{\partial v_j}{\partial x_j} = 0, \tag{1.48}$$

which is the Lagrangian form of the continuity equation. It can be shown to represent the conservation of mass by integrating over a finite volume region V, which gives

$$\int_V \frac{d\rho}{dt} dV + \int_V \rho \frac{\partial v_j}{\partial x_j} dV = 0. \tag{1.49}$$

The density is constant across the whole volume, so this is equivalent to

$$\frac{d}{dt}(\rho V) + \int_V \rho \frac{\partial v_j}{\partial x_j} dV = 0, \tag{1.50}$$

where the second term can be broken down using integration by parts

$$\int_V \rho \frac{\partial v_j}{\partial x_j} dV = \rho \int_V \frac{\partial v_j}{\partial x_j} dV - \int_V \frac{d\rho}{dV} \left(\frac{\partial v_j}{\partial x_j} dV \right) dV. \tag{1.51}$$

We can assume that the velocity of the gas is non-zero, but that there is a finite extent of this moving region, at the surface of which the velocity is zero. From the divergence theorem we know that

$$\int_V \frac{\partial v_j}{\partial x_j} dV = \int_S v_j \hat{n}_j dA = 0, \tag{1.52}$$

where $\hat{n}_j$ is the outward facing normal to the surface. It is clear form this that $d(\rho V)/dt = 0$. The above is equal to zero because we required that the velocity at the surface S of the volume is zero, so the integral becomes zero. This effectively states that the mass of a fluid element is constant with time.

The next term, the momentum of the fluid state vector, gives

$$\frac{d}{dt}(\rho v_i) = \frac{\partial}{\partial t}(\rho v_i) + v_j \frac{\partial}{\partial x_j}(\rho v_i), \tag{1.53}$$

which, again, can be expanded using the product rule. This leaves

$$\rho \frac{dv_i}{dt} + v_i \frac{d\rho}{dt} = \rho \frac{\partial v_i}{\partial t} + v_i \frac{\partial \rho}{\partial t} + v_j \left(\rho \frac{\partial v_i}{\partial x_j} + v_i \frac{\partial \rho}{\partial x_j} \right). \tag{1.54}$$

The Lagrangian continuity equation shows that the second term of the LHS is zero. Rearranging the above equation produces

$$\frac{dv_i}{dt} = \frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} + \frac{v_i}{\rho} \left(\frac{\partial \rho}{\partial t} + v_j \frac{\partial \rho}{\partial x_j} \right). \tag{1.55}$$

The terms in the brackets are clearly zero, based on Equation (1.17), leaving us with

$$\frac{dv_i}{dt} = \frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial P}{\partial x_i} + F_i, \tag{1.56}$$

with the right most part comes from the Eulerian momentum equation (see Equation (1.25)). The rate of change in the velocity, or acceleration, is simply shown to be equal to the net force acting upon it. This equation is the Lagrangian form of the momentum equation, and gives the acceleration of the fluid element.

The final equation is found by the same process, this time using the energy term of the fluid state vector. As before, we substitute it into Equation (1.44), and expand as before, giving

$$\frac{d}{dt}(\rho E) = \frac{\partial}{\partial t}(\rho E) + v_i \frac{\partial}{\partial x_i}(\rho e) \tag{1.57}$$

The Eulerian energy equation can be substituted in for the first term on the RHS of this equation, producing

$$\frac{d}{dt}(\rho E) + \frac{\partial}{\partial x_i}(\rho E + P)v_i - v_i \frac{\partial}{\partial x_i}(\rho E) = \rho v_i F_i \tag{1.58}$$

which in turn gives

$$\frac{d}{dt}(\rho E) + (\rho E + P)\frac{\partial v_i}{\partial x_i} + v_i \frac{\partial}{\partial x_i}(\rho E + P) - v_i \frac{\partial}{\partial x_i}(\rho E) = \rho v_i F_i. \tag{1.59}$$

The expansion of the third term on the LHS cancels with the forth term, leaving

$$\frac{d}{dt}(\rho E) + (\rho E + P)\frac{\partial v_i}{\partial x_i} + v_j \frac{\partial P}{\partial x_j} = \rho v_j F_i, \tag{1.60}$$

which describes the Lagrangian conservation of energy.

As before, we can summarise the Lagrangian fluid equations that describe the evolution of the mass, momentum and energy of a fluid element, this time with a tracer that moves with the fluid flow. We also therefore include the change in position, which is equal to the Eulerian velocity. Together these are

$$\frac{dr_i}{dt} = v_i, \tag{1.61}$$

$$\frac{d\rho}{dt} = -\rho \frac{\partial v_j}{\partial x_j}, \qquad \text{[continuity equation]} \tag{1.62}$$

$$\frac{dv_i}{dt} = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + F_i, \qquad \text{[momentum conservation]} \tag{1.63}$$

$$\frac{d}{dt}(\rho E) = \rho v_j F_i - (\rho E + P)\frac{\partial v_i}{\partial x_i} - v_j \frac{\partial P}{\partial x_j}. \qquad \text{[energy conservation]} \qquad (1.64)$$

The fluid elements have fixed mass, and so mass is conserved by construction. These equations form the basis of a number of widely used numerical methods for solving the evolution of fluids, most notable smoothed particle hydrodynamics (Gingold & Monaghan, 1977; Monaghan, 1992; Springel & Hernquist, 2003). These approaches have been applied extensively to astrophysical problems. I will discuss the specifics of these implementations in Section 1.2.3.

Fluid Equations - Viscosity

Up to this point we have assumed that no physical particles are moving between fluid elements. This assumption is not always applicable, so it is important to be able to include this possibility by introducing viscosity in our fluid equations. This effectively models the microscopic transport of momentum caused by friction between fluid elements. This is done by adding the viscous stress tensor Π, which captures these microscopic friction effects where adjacent fluid elements are moving in different directions. Unlike the pressure term, this is not isotropic. As such the viscous stress tensor must depend on the variation of velocity across space, the so called shear viscosity, but should also fall to zero when neighbouring elements are rotating together (with the same angular velocity about a common centre). It should also include the effects of bulk viscosity, which comes from the time required for injected to energy to be distributed across different energetic degrees of freedom. The viscous stress tensor therefore takes the form

$$\Pi_{ij} = \eta\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3}\frac{\partial v_k}{\partial x_k}\delta_{ij}\right) + \zeta\frac{\partial v_k}{\partial x_k}\delta_{ij}, \qquad (1.65)$$

where η is the shear viscosity coefficient and ζ the bulk viscosity coefficient. In most scenarios, the bulk velocity is negligible, and only the shear viscosity needs to be taken into account.

The Eulerian momentum and energy equations become

$$\frac{\partial}{\partial t}(\rho v_i) + \frac{\partial}{\partial x_j}(\rho v_i v_j) = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\Pi_{ij} + \rho F_i \qquad (1.66)$$

and

$$\frac{\partial}{\partial t}\left(\frac{1}{2}\rho v^2\right) + \frac{\partial}{\partial x_j}\left(\frac{1}{2}\rho v^2 v_j - \Pi_{ij} v_i\right) = -v_j \frac{\partial P}{\partial x_j} + \rho v_j F_j - \Pi_{ij}\frac{\partial v_i}{\partial x_j}. \tag{1.67}$$

These, together with the unchanged continuity equation, are collectively known as the Navier-Stokes equations, and are used in place of the inviscid fluid equations in scenarios where viscosity is significant, such as proto-planetary disks (Rafikov, 2017), where viscosity of the gas provides a mechanism to transport angular momentum across the disk.

Gravity

The fundamental long range force that drives much of the motion of all mass in the Universe is gravity. For simplicity, I will only discuss non-relativistic, Newtonian gravity, as this describes the vast majority of astrophysical scenarios. The motion of collisionless objects, such as dark matter or stars, can be described by the solution to a set of partial differential equations, analogous to the Lagrangian fluid equations, with

$$\frac{d\mathbf{r}}{dt} = \mathbf{v}, \tag{1.68}$$

$$\frac{d\mathbf{r}^2}{d^2t} = \frac{d\mathbf{v}}{dt} = \mathbf{g}, \tag{1.69}$$

where $r = (x, y, z)$ is position, $\mathbf{v} = (v_x, v_y, v_z)$ is velocity, and $\mathbf{g}$ is the acceleration due to gravity, or specific force. This acceleration is found from the gradient of the gravitational potential $\mathbf{g} = -\nabla\Phi$. For a density distribution $\rho(\mathbf{r}, t)$, with mean density $\bar{\rho}$, the Poisson equation gives

$$\nabla \cdot \mathbf{g} = -4\pi G\left(\rho(\mathbf{r}, t) - \bar{\rho}\right). \tag{1.70}$$

This effectively says that the acceleration is created by over and under densities in the density field. These equations can be solved numerically to calculate the evolution of an ensemble of dark matter particles that represent the underlying dark matter density distribution.

In practice, gravitational force, and so acceleration, is often calculated directly

between particles. The absolute value of the Newtonian force F_{ij} between two point masses, labelled i and j, with masses m_i and m_j is defined as

$$F_{ij} = \frac{Gm_i m_j}{r_{ij}^2}, \tag{1.71}$$

where r_{ij} is the distance between the two objects. However, this only describes the size of force exerted on each body by the other. To calculate the change in motion of the massive objects, one needs to know the acceleration $\mathbf{a}_i = d\mathbf{v}_i/dt$. As before, the velocity is simply the change in position $\mathbf{r}_i$ with time $\mathbf{v}_i = d\mathbf{r}_i/dt$, where position, velocity and acceleration have x, y and z components, such that $\mathbf{r} = (x, y, z)$, $\mathbf{v} = (v_x, v_y, v_z)$ and $\mathbf{a} = (a_x, a_y, a_z)$. Newton's second law relates the acceleration of i to the net force acting on i, from all sources, via

$$\mathbf{F}_i = m_i \mathbf{a}_i = m_i \frac{d\mathbf{v}_i}{dt} = m_i \frac{d\mathbf{r}_i^2}{d^2 t}. \tag{1.72}$$

The force vector is given generally as $\mathbf{F} = F\mathbf{r}/|\mathbf{r}|$, where r, in this case, is the direction of the force, and $|\mathbf{r}|$ the length of $\mathbf{r}$. The distance between two positions is given by $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$. The gravitational force on i, from j, in component form, can be written as

$$\mathbf{F}_{ij} = \frac{Gm_i m_j}{r_{ij}^2} \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} = Gm_i m_j \frac{\mathbf{r}_j - \mathbf{r}_i}{|\mathbf{r}_j - \mathbf{r}_i|^3}. \tag{1.73}$$

The acceleration caused by this force is then simply

$$\mathbf{a}_{ij} = Gm_j \frac{\mathbf{r}_j - \mathbf{r}_i}{|\mathbf{r}_j - \mathbf{r}_i|^3}. \tag{1.74}$$

The motion of a massive object can thus be updated based on its direct gravitational interaction with another object. The equivalent action on the other object is, of course, equal and opposite. Alternatively, the acceleration from a gravitational potential can be calculated using the same component approach. The force is the gradient of the potential, and the components are found in the same way as before.

The effect of gravity on the fluid equations is handled by introducing source terms like those described in the derivation of the Euler equations. These source terms

act on the momentum and energy conservation equations, with

$$\mathbf{S} = \begin{pmatrix} 0 \\ \rho\mathbf{g} \\ \rho\mathbf{v}\mathbf{g} \end{pmatrix}. \tag{1.75}$$

Comoving Coordinates

In our current standard cosmological model of dark matter, baryonic matter, and an expanding Universe, the evolution of the various components are tracked using comoving coordinates. The comoving frame handles the expansion of the Universe, and is included in the fundamental equations in the following ways. The fluid equations are modified with the cosmological expansion scale factor $a = 1/(1+z)$, where z is redshift, and the Hubble parameter $H = d\ln a/dt$. They become (Bertschinger, 1998)

$$\frac{\partial\rho}{\partial t} + \frac{1}{a}\nabla\cdot(\bar{\rho}\mathbf{v}) = 0, \tag{1.76}$$

$$\rho\frac{\partial\mathbf{v}}{\partial t} + \frac{1}{a}\rho\mathbf{v}\cdot\nabla\mathbf{v} + H\rho\mathbf{v} + \frac{1}{a}\nabla P = 0, \tag{1.77}$$

and

$$\rho\frac{\partial}{\partial t}\frac{P}{(\gamma-1)\rho} + \frac{1}{a}\rho\mathbf{v} + \frac{P}{a}\nabla\cdot\mathbf{v} = 0. \tag{1.78}$$

The velocity $\mathbf{v}$ is now the peculiar velocity, or the velocity relative to the cosmological expansion. For clarity, t is still proper time. The density is from the baryonic gas only, since the collisionless components do not contribute. For collisionless matter, the set of partial differential equations that describe the evolution of the initial conditions become (Bertschinger, 1998)

$$\frac{d\mathbf{r}}{dt} = \frac{1}{a}\mathbf{v}, \tag{1.79}$$

$$\frac{d\mathbf{v}}{dt} + H\mathbf{v} = \mathbf{g}, \tag{1.80}$$

and

$$\nabla \cdot \mathbf{g} = -4\pi G a \left(\rho(\mathbf{r}, t) - \bar{\rho}\right). \tag{1.81}$$

1.2.2 Initial Conditions

In the standard ΛCDM cosmological paradigm, simulations of cosmological scale systems are performed in boxes with comoving coordinate schemes that captures the expansion of the Universe. Within this box, a distribution of dark matter and gas is set up to replicate the conditions inferred from the CMB. Setting up this distribution is not entirely simple. The non-linear nature of gravity means any initial perturbations to the uniform background, be they intentional or caused by the discreet sampling of the underlying density, can grow rapidly. Initial conditions must model both the background uniform density, and the perturbation to this background.

A regular grid of particles can be used for the uniform component, but this introduces preferential directions and spurious periodicities on small scales (Bode et al., 2001). It also cannot easily reproduce the perturbations to the uniformity. A solution is to create what is known as a *glass* (White, 1994). Particles are distributed randomly within the box, and then the gravitational force on each particle is calculated. There are a variety of ways to calculate this force, discussed in more detail in Section 1.2.4. The direction of this force is reversed, effectively making the gravitational force repulsive, and the evolution is applied. This system is run for a sufficiently long period of time that the particles reach a quasi-equilibrium state where the net force on each particle falls to near zero. This method produces a distribution that has no regular structure, so no preferential directions, and so significant perturbations on scales larger than the mean particle separation. Evolving this distribution under gravity produces no small scale structure.

The desired distribution, however, is not uniform. The initial perturbations grow with time as the over dense perturbations contract. Eventually the gravitational contraction becomes non-linear, but before this point, the fluctuation in the background density can be produced using the Zel'dovich approximation (Zel'Dovich, 1970; White, 2014). One starts by describing the underlying density distribution, both dark matter and gas components, with a set of Lagrangian fluid elements. The fluid element at comoving position $\mathbf{x}_0$, at time t_0, has a position at some later

time that is the initial position plus a displacement $\mathbf{s}$, or explicitly

$$\mathbf{x}(\mathbf{x}_0, t) = \mathbf{x}_0 + \mathbf{s}(\mathbf{x_0}, t). \tag{1.82}$$

By construction, the displacement fully describes the evolution of the cosmological fluid. The Zel'dovich approximation is the linear approximation of this displacement, from Lagrangian perturbation theory (White, 2014). Once overdense regions decouple from the expansion, the trajectories of their component particles are no longer described by a linear perturbation. Before this point, however, the Zel'dovich approximation can accurately reproduce the desired perturbations (White, 1994). The perturbed gravitational potential, taken from the approximation, is applied to the uniform glass, moving the particles in a similar manner to before, but this time to reproduce the desired density. The elements also require peculiar velocities consistent with their perturbed trajectories, which are supplied by the gravity solver used to produce the glass. This can be used to recreate distributions with the mean cosmological density, or to sample regions of over or under density.

The above approach produces a distribution of matter that accurately reproduces the observed conditions at extreme redshift, close to the surface of last scattering. The evolution of this distribution can then be calculated by solving the equations that govern the critical physical processes that describe the behaviour of the key components. In the standard cosmological model, these are baryonic gas, and cold dark matter.

1.2.3 Gas Dynamics

Modelling the behaviour of the baryonic gas component of the universe is a complex problem. As discussed above, the gas is described by the fluid equations, a set of partial differential equations, that represent the conservation of mass, momentum, and energy, for a compressible, inviscid fluid. In their one dimensional Eulerian form, with no source or sink terms, they are written compactly as

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{Q}) = 0, \tag{1.83}$$

where $\mathbf{Q}$ is the conserved fluid variables, and $\mathbf{F}(\mathbf{Q})$ their corresponding fluxes, given by

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho\mathbf{v} \\ \rho E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho\mathbf{v} \\ \rho\mathbf{v}^2 + P \\ (\rho E + P)\mathbf{v} \end{pmatrix}. \quad \begin{matrix} \text{[mass conservation]} \\ \text{[momentum conservation]} \\ \text{[energy conservation]} \end{matrix} \tag{1.84}$$

Here the symbols have their usual physical meanings: ρ is mass density, $\mathbf{v}$ is velocity, E is specific energy, and P is pressure. For an ideal gas, we use the polytropic equation of state, with the adiabatic index γ,

$$P = \rho(\gamma - 1)\left(E - \frac{\mathbf{v}\cdot\mathbf{v}}{2}\right). \tag{1.85}$$

The corresponding sound speed c_s of this gas is

$$c_s = \sqrt{\frac{\gamma P}{\rho}}. \tag{1.86}$$

Solving these equations analytically, for all but simplest of cases, is effectively impossible. Modeling the behavior of a fluid numerically has been a field of study for many years, but the advent, in recent decades, of increasingly powerful computers has allowed new techniques to be implemented. The available methods are highly varied, but can be divided into broad categories, based on their fundamental approach. One way to separate them is by the nature of their *measurement tool*. Modeling schemes are either Eulerian, meaning the positions where the state of the fluid is *measured* (i.e. where we track these values) are stationary in space, or Lagrangian, meaning the measurement positions move with the fluid. Techniques can also be divided by how they formulate the fluid, broadly, either as a mesh of cells or as a set of particles. Mesh methods break the domain down into a series of finite volume cells, with the fluid variable stored in these cells. Particle methods, such as smoothed particle hydrodynamics (SPH), use a set of particles to represent the fluid elements. The condition of the fluid is modeled using smoothing kernels around each of the particles. These are then used as the variables in the Euler equations. The methods are discussed in more detail in Section 1.2.3, followed by a brief discussion of the solver schemes that are used in mesh based methods.

Eulerian Schemes

A fundamentally simple approach is to break the space up into a series of finite volume cells. This can be done in a wide range of ways, from a uniform Cartesian grid, to an unstructured mesh. The fluid variable values, such as density, are held within these cells. The initial conditions are known across the space, so the problem becomes an initial value problem (IVP), discretised into a finite volume scheme. The solution to the numerical approximation of fluid equations is found at the boundaries of the cells, effectively breaking the problem down into a series of one dimensional Riemann problems, the specifics of which will be addressed shortly. The discretised form of Equation (1.40) is found from the Taylor expansion of the solution, over time-step Δt, with

$$\mathbf{Q}(x, t + \Delta t) = \mathbf{Q}(x, t) + \Delta t \frac{\partial \mathbf{Q}}{\partial t} + O(\Delta t^2). \tag{1.87}$$

Substituting the initial differential equations, this can also be written as

$$\mathbf{Q}(x, t + \Delta t) = \mathbf{Q}(x, t) - \Delta t \frac{\partial \mathbf{Q}}{\partial x} + O(\Delta t^2). \tag{1.88}$$

The order of expansion effect the temporal order of the method, and can vary with scheme. The basic methods take only the linear and lower terms, leaving the approximation of the solution, for time-step n to $n + 1$, as

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n + \frac{\Delta t}{\Delta x} \Delta \mathbf{Q}_i, \tag{1.89}$$

where $\Delta \mathbf{Q}_i$ is the change across the cell. This change in the fluid state $\Delta \mathbf{Q}$ for a given cell i, at position l, is

$$\Delta \mathbf{Q}_i = -\frac{\Delta t}{\Delta x} \sum_{j=1}^{D} [\mathbf{F}_j(l_{j+1/2}, t) - \mathbf{F}_j(l_{j-1/2}, t)]. \tag{1.90}$$

The flux is calculated at the cell boundaries at $l_j \pm 1/2$ from the centre of the cell. Here, D denotes number of dimensions. In 1D, each cell only has one pair of faces, one on either side of the cell centre, but in 2D and 3D, the flows through the other faces must also contribute to the change in state. The update effectively sums the one dimensional fluxes perpendicular to each face. At these faces, a variety of schemes can be used to approximate the condition of the fluid on either side. These range from simple piece-wise linear reconstructions to more complicated parabolic up-winding schemes.

Riemann solvers are commonly used in this context. An initial value problem, with one discontinuity and two uniform regions either side, is known as a Riemann problem, and a wide array of exact and approximate Riemann solvers have been developed to solve it (Godunov & Bohachevsky, 1959; Glimm, 1965; Harten, 1983; Leer, 1984). As mentioned before, each cell interface is effectively a 1D Riemann problem, even when the number of dimensions is greater than one. The Riemann problem is simply aligned to the normal of the face. The solution is the combination of the solutions to the whole set of Riemann problems. The net result is a relatively versatile approach, which scales easily, by simply adding more cells, and handles aspects such as periodic boundary conditions well (Berger & Oliger, 1984). These are often handled by creating a set of ghost cells around the edge. These ghosts act as surrogates for the cells at the other side of the mesh. Any change made to the ghost is copied to this other cell.

There are some significant downsides to using this approach. These include a lack of Galilean invariance, their tendency for over-mixing, and the flows only occurring in the directions normal to the faces (Springel, 2010). In the uniform grid example, this means any flow that should be spherically symmetric will have significant anomalous fringe effects at small radii. Unstructured meshes are much less susceptible to this problem. This will be discussed in greater detail, along with other topics mentioned here, such as periodic boundaries, in Chapter 3.

Adaptive Mesh Refinement

To combat the poor resolution in areas with high density, which are usually the areas of interest, a method known as adaptive mesh refinement (AMR) has been developed (Berger & Oliger, 1984). This is a method for solving partial differential equations (PDE), such as the fluid equations, where a coarse grid is overlaid by a finer grid that covers a smaller area which requires higher resolution. This could be a region where there is a rapid change in the solution to the PDE. The refinement can be done a large number of times, producing a highly versatile algorithm. The meshes are independent, so it is possible to have a stationary coarse grid with a moving fine grid overlaid to track a moving feature, such as a shock. Some additional error is introduced at the boundary between grids (Berger & Oliger, 1984). This approach improves resolutions in places, but is computationally intensive and does not improve the overall accuracy order of the simulation, as it does not refine everywhere. AMR has been utilised by several

cosmological simulation codes, including RAMSES (Teyssier, 2002), and ENZO (Bryan et al., 2014).

MUSCL-Hancock

The Monotonic Upstream-Centered Scheme for Conservation Laws (MUSCL-Hancock) (van Leer, 1979; Leer, 1984; van Leer, 2006) is a commonly used approach for solving hydrodynamics in schemes with moving meshes. This method builds on the simple one dimensional Godunov problem with a linear reconstruction concept for solving the hydrodynamics at a boundary, to produce a fast and accurate solver (Springel, 2010; Bryan et al., 2014; Hopkins, 2015). For situations with a grid in more than one dimension, the same approach can be used, with some additions. In the basic setup, there are a series of cells that have a set of variables attached to them that represent the state of the fluid in that region. These variables are assumed to be constant throughout the cell.

Linear reconstruction replaces these piece-wise constant variables throughout each cell by introducing a linear slope passing through the center of two neighboring cells. Using this scheme a value for each variable is interpolated on either side of the boundary. Extending into three dimensions converts this to a surface integral to calculate the flux through the face.

A slope limiter is added to prevent spurious oscillations from appearing in the linear reconstruction where there are shocks in the simulated system. This limiter applies a limiting function to the slope of the linear reconstruction, used in the interpolation of the left and right states at the boundary (Sweby, 1984). It is used to force the piece-wise reconstruction to be total variation diminishing (TVD), preventing the before mentioned oscillations. There are a variety of slope limiter functions available, some of which are not TVD, but which are still used for as they are computationally cheaper (Springel, 2010). The fundamentals are described by Sweby (Sweby, 1984). It is also possible to produce a parabolic, rather than linear, reconstruction scheme using a similar method, which adds spatial accuracy, but obviously this is more computationally expensive. Hancock's addition to this method was to include a predictor-corrector step (van Leer, 2006).

The fundamental mathematics of the MUSCL scheme can be laid out in the following way (Sohn, 2005). For a constant space step (Δx), such that $x_i = i\Delta x$ and $x_{i+1/2} = (1/2)(x_i + x_{i+1})$, we get cell i defined as $(x_{i-1/2}, x_{i+1/2})$. Q_i is the

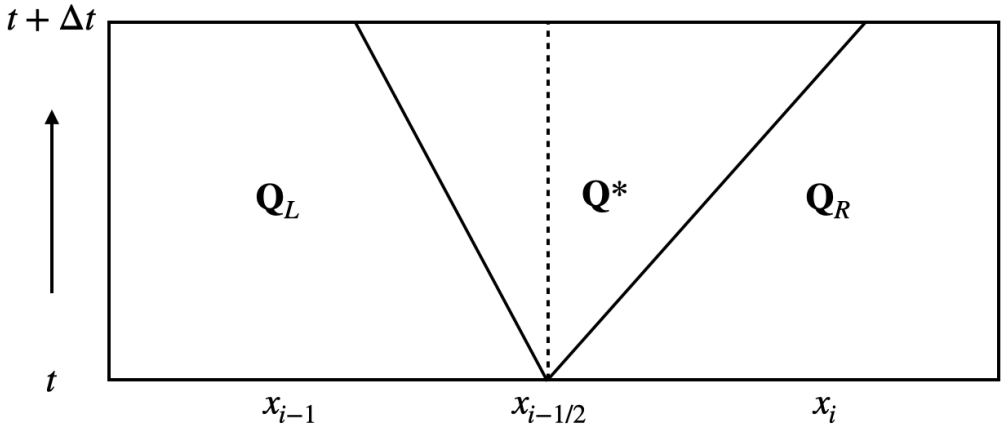

Figure 1.4 *Fundamental waves from the boundary between two 1D cells. The Riemann problem at the boundary between cells $i-1$ and i, with the boundary shown as the dashed line, produces two waves that propagate into the cells either side of the boundary, shown by the solid lines. The initial states either side of the boundary are $\mathbf{Q}_L$ and $\mathbf{Q}_R$, with the intermediate state between the waves denoted by $\mathbf{Q}^*$.*

average value of Q over cell i at time t^n. We can use Equation (1.90), where $F_{i+1/2}$ is is the flux at the face at $x_{i+1/2}$, given as (van Leer, 1979)

$$F_{i+1/2} = F(Q^n_{i+1/2}) + \frac{\Delta t}{2}\left(\frac{\partial}{\partial t}F(Q)\right)^n_{i+1/2}. \tag{1.91}$$

The right hand side of this equation can be solved in variety of ways, often requiring the solving of a *generalized Reimann problem*, which involves resolving the discontinuity with some kind of piece-wise reconstruction (van Leer, 1979; Sohn, 2005).

Harten-Lax-van Leer-Contact Method

The Harten-Lax-van Leer-Contact (HLLC) method of solving the hydrodynamics of a problem is another commonly used approach (Toro et al., 1994). It solves the Riemann problem at the boundary between two cells by assuming the solution can be approximated as two waves moving away from the face of the cell. These waves are assumed to be moving at constant speed. In this context, the waves are the flow of material from either side of the face, without reference to the opposing side. A third, middle wave is added to fully capture the effects of contact discontinuities. The precursor Harten-Lax-van Leer (HLL) method (Harten, 1983), uses the same approach, but does not contain the central third wave, so

does not resolve contact discontinuities well. The base method has three regions: $\mathbf{Q}_L$, $\mathbf{Q}^*$, and $\mathbf{Q}_R$. This setup is shown in Figure 1.4, where time increases in the y-direction. The dashed line gives the positions of the boundary, and the solid lines the position of the two waves. The state between the two waves, which move with speed S_L and S_R, for the boundary $i-1/2$, is defined as

$$\mathbf{Q}^*_{i-1/2} = \frac{S_R\mathbf{Q}_R - S_L\mathbf{Q}_L - (\mathbf{F}_R - \mathbf{F}_L)}{S_R - S_L}, \tag{1.92}$$

where $\mathbf{F}_R$ and $\mathbf{F}_L$ are the fluxes evaluated for the left and right states. The flux for this region is given by

$$\mathbf{F}^*_{i-1/2} = \frac{S_R\mathbf{F}_R - S_L\mathbf{F}_L - S_RS_L(\mathbf{Q}_R - \mathbf{Q}_L)}{S_R - S_L}. \tag{1.93}$$

The flux at the boundary is then

$$F^{HLL}_{i-1/2} = \begin{cases} \mathbf{F}_L & \text{if } S_L \geq 0 \\ \mathbf{F}^* & \text{if } S_L \leq 0 \leq S_R \\ \mathbf{F}_R & \text{if } S_R \leq 0 \end{cases}. \tag{1.94}$$

The determination of the wave speed is now all that is left. There are several approximations that are used, but a common one is to assume the solution is isentropic and that the two waves are rarefaction waves. These assumptions lead the velocity v^* and sound speed c^* in the central region to be

$$v^* = \frac{1}{2}(v_L - v_R) + \frac{c_L - c_R}{\gamma - 1}, \tag{1.95}$$

$$c^* = \frac{1}{2}(c_L + c_R) + \frac{1}{4}(\gamma - 1)(v_L - v_R). \tag{1.96}$$

The wave speeds are then set as

$$\begin{aligned} S_L &= \min(u_L - c_L, u^* - c^*) \\ S_R &= \max(u_R + c_R, u^* + c^*) \end{aligned} \tag{1.97}$$

The introduction of the contact wave splits the central region in two. This produces four constant states separated by these three waves. There are four regions: $\mathbf{Q}_L$, $\mathbf{Q}^*_L$, $\mathbf{Q}^*_R$ and $\mathbf{Q}_R$. The wave fronts that define these regions have speeds S_R, S^* and S_R. The speeds are found in the same manner as before. The

flux is then calculated for the central two regions (Q_{*L} and Q_{*R}) based on the fluxes created by the left and right moving waves. The integral over these regions gives the final estimate of the flux at the cell boundary.

Lagrangian Schemes

An alternative approach is to allow the tracers to move with the gas flow. The Lagrangian form of the adiabatic fluid equations, with no gravity, are given by

$$\frac{d\rho}{dt} + \rho\nabla \cdot \mathbf{v} = 0, \tag{1.98}$$

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho}\nabla P, \tag{1.99}$$

$$\frac{de}{dt} = -\frac{P}{\rho}\nabla \cdot \mathbf{v}. \tag{1.100}$$

With this approach, we discretise the numerical domain by mass, instead of space. The material in the box is represented by a set of N particles of given mass, which are tracers of the underlying fluid. This method is analogous to N-body particle approach to modelling the gravitational interaction of cold dark matter. The motion of these particles describes the evolution of the gas. The fluid nature of the gas is captured by the presence of a pressure term in the acceleration of the particles. Gradients in the pressure accelerate the particles, an effect not present with collisionless media.

Smoothed Particle Hydrodynamics

A well established Lagrangian approach is smoothed particle hydrodynamics (SPH) (Gingold & Monaghan, 1977). This approach uses a sampling of the mass elements of a numerical region to reconstruct the physical conditions. A region of gas is discretised in to N particles of mass m. The basic SPH formulation calculates the motion of the i^{th} particle using

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i, \tag{1.101}$$

$$\frac{d\mathbf{v}_i}{dt} = -\frac{1}{\rho_i}\nabla P_i \tag{1.102}$$

where $\mathbf{r}_i$ is the position of the particle, $\mathbf{v}_i$ the velocity of the particle, ρ_i the density, and P_i is the pressure. The density and pressure at an arbitrary position r are found by summing contributions from a set number N_k of nearest particles

$$\rho(\mathbf{r}) = \sum_{j=1}^{N_k} m_j W(\mathbf{r} - \mathbf{r}_j, h), \tag{1.103}$$

$$P(\mathbf{r}) = \sum_{j=1}^{N_k} m_j \frac{P_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h), \tag{1.104}$$

using a smoothing kernel $W(\mathbf{r}, h)$ of scale length h, similar to that used in the PM gravity method. This kernel is chosen such that it goes to zero at its edge $2h$. The scale length h is chosen to contain N_k particles, and so varies with the distribution. The more particles in a region the greater the spatial resolution. A typical kernel used in SPH codes is the cubic spline (Monaghan, 1992)

$$W(p) = \begin{cases} \sigma\left[1 - \frac{3}{2}q^2\left(1 - \frac{q}{2}\right)\right] & \text{if } 0 \le q \le 1 \\ \frac{\sigma}{4}(2-q)^3 & \text{if } 1 < q \le 2\,, \\ 0 & \text{if } q > 2 \end{cases} \tag{1.105}$$

where $q = r/h$, and σ is dependent on the dimensionality of the problem. In three dimensions $\sigma = 1/\pi h^3$.

The Lagrangian nature of this method allows for natural resolution adaptation, with more particles providing greater spatial resolution in areas of greater density. This is a powerful advantage, but there are drawbacks to using this approach. Sparse regions are not well resolved as there are few particles left there. Particles in crossing streams can move through one another, leading to spurious oscillations in these circumstances, such as in shocks. This can be counteracted by introducing artificial viscosity, which essentially allows the SPH particles to dissipate kinetic energy into thermal energy. A number of state-of-the-art multiphysics simulation codes (Hernquist & Katz, 1989; Springel, 2005) use SPH implementations to great effect.

In recent years, a number of moving mesh methods have been developed and implemented. These seek to combine the advantages of the two classical methods, while removing the disadvantages. The fluid is represented by a set of cells that move and deform with the bulk motion of the gas they are modelling. This approach takes the resolution adaptation of the SPH methods, and combines it with the stability and shock handling capabilities of the grid based methods.

Solvers such as MUSCL-Hancock can be converted for use on a moving mesh. Fluid can move between cells, and the cells themselves can move and deform. The flux problem is often transformed into the rest frame of the boundary (i.e. the cell wall). This only requires an adjustment of the relevant variables, such as velocity, and has little effect on the construction of this solver. It does, however, simplify the formulation of the conservation equations. In this way, the method is ideally suited to a moving mesh scheme, such as AREPO (Springel, 2010).

The GIZMO (Hopkins et al., 2014) code uses this method (Lanson & Vila, 2008; Gaburov & Nitadori, 2011), in a slightly different way to AREPO, by implementing it over a so called 'mesh-less' scheme. In GIZMO the step function at the boundaries is not used. Instead the boundaries are effectively smoothed by the implementation of a weighting kernel that changes the contribution from the particle/cell. The whole volume of these cells is then integrated over to get the appropriate value for the conserved quantity (e.g. mass). This value is then used in the Euler equations, using the solver to solve for the flux, but the cell face is replaced by an 'effective face' that is defined by the volume integral.

The HLLC method is used in codes that solve purely advective problems. These are problems where there is constant pressure and velocity, but different densities, such as with the relativistic TESS code (Duffell & MacFadyen, 2011). It is also possible to use the contact (central) wave to introduce the effects of a moving face. This is again useful in relativistic problems where converting to the rest frame of the face does not simplify the problem. The concentric cylindrical cell code DISCO (Duffell, 2016) uses the method for this reason, as well as the well resolved contact discontinuities.

Time Step Limitations

No matter which discretisation of the fluid is used, be it in space or mass, the numerical solution to the fluid equations is found by numerical integration over time. This requires a discretisation of time into a set of time-steps of size ΔT, over which the numerical integration is incrementally performed. The choice of time-step is limited by the discretisation of the fluid, defined by the Courant-Friedrichs-Lewy (CFL) condition, which states that the numerical domain of dependence of the method must enclose the physical domain of dependence (Holmes, 2007). In other words, when calculating the change in the fluid at a given position, the numerical method must include information from every part of the fluid that can physically influence that position. This physical region is defined by how fast information about the fluid can travel, which is the sound speed.

In the case of a structured grid based solver, including those using AMR, this manifests itself as an upper limit on the time-step, found using the time taken for information to cross the cells of size Δx, with

$$\Delta t < C_{CFL} \frac{\Delta x}{c_s + |v|}, \tag{1.106}$$

where c_s is the sound speed and $|v|$ is the speed of the gas. The C_{CFL} coefficient is applied to guarantee that the time-step is consistent with the condition, with $C_{CFL} < 1$. An equivalent calculation can be performed in particle based methods, where the kernel radius is used in place of cell size, to estimate distance. Methods built on unstructured meshes use a similar approximation, calculating the radius of a sphere of volume equal to the volume of the cell. These approximations each provide a time-step limitation for a given cell or particle.

The sound speed and velocity will inevitably vary across the domain, as can the size of the cells, and smoothing lengths. In their basic formulations, the various methods take the smallest time-step required by any element. This inevitably means that many elements are evolved at time-steps shorter than is required. The computational expense is therefore larger than strictly required. Many modern codes implement varied time-step mechanisms, typically by binning elements into groups by the required time-steps, with bin limits increasing as powers of two of the minimum (Katz et al., 1996; Springel, 2010). These improve the efficiency of the method, particularly in scenarios with extreme variation in time-step across the domain, such as in cosmological simulations, where the conditions in galaxies

and voids require very different time-steps.

1.2.4 Gravity

All bodies with mass must experience, and produce, the gravitational forces that influence the evolution of the universe. In the case of cold dark matter, this is the only effect that needs to be accounted for, since the dark matter particles are assumed to be collisionless. There are several, widely used, methods for performing this calculation. Some of the most common are outlined below. First it is important to clarify where the gravity modelling of gravity starts.

Particle-Particle Algorithms

The gravity calculation can be done with a direct summation of particle-particle interactions (Mo et al., 2010), using a particle-particle (PP) algorithm, where the gravitational force on particle i is given by

$$\mathbf{F}_i = -\sum_{j \neq i} G m_i m_j \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3}. \tag{1.107}$$

The gravitational force on any particle i, at position $\mathbf{r}_i$, can be calculated, based on N point masses at positions $\mathbf{r}_{0,1,\cdots,i,\cdots,N}$. Integrating the resultant force over a given time step produces the gravitational evolution of the particles. This is a very intuitive approach that can be made very accurate when large numbers of particles and small time steps are used. However, it is computationally intensive, scaling for N particles as $O(N^2)$, especially for systems where distant contributions are negligible, so more efficient schemes have been developed.

Tree Algorithm

In modern simulations, the number of particles being simulated makes calculation of the forces between every particle pair prohibitively expensive, while many of the particles are at such high spatial separation that they have little effect on one another. This requires approximation methods to be used for particles beyond a certain distance. Each particle must be able to efficiently find all its neighbors within this distance. A common approach to solve this problem is to use a

tree algorithm. The algorithm decomposes the domain into a tree that divides particles into smaller and smaller groups, known as *hierarchical grouping*. A number of solutions to this problem have been found, starting in the 1980s (Barnes & Hut, 1986), with other more recent examples from GADGET-2 (Springel, 2005) and PKDGRAV (Potter et al., 2017), all based around the same principle.

A simple method is to first enclose all particles in a cube. This cube is then divided into eight regions. Each of these boxes has an edge 1/2 the length of the original box. This is known as an octree. The division is done repeatedly for all the sub-regions, until every sub-region has, at most, one particle in it. These are the leaves of the constructed tree. A visual representation of the 2D quadtree equivalent of this is shown in Figure 1.5. The gravitational force of the particle in each node is replaced by the multipole expansion of the force. A relatively small number of the expansion terms can be used to create an accurate force calculation (Mo et al., 2010). The further away a node is from the target particle, the fewer terms can be used. A tree-walk can then be used to calculate the total force on any particle. This is done by summing up the contributions from tree nodes. Only particles within a certain distance are inside this sum, and those are innately identified by the tree-walk. The center of mass of each cell is tested to see if it is within the required distance of the particle of interest. If it is then the moment is included in the sum. If it is not then the walk opens the node and continues down to the next layer. These methods can reduce the scaling to $O(N \log N)$ (Barnes & Hut, 1986).

Particle-Mesh Algorithms

Another option for gravitational scheme is the particle-mesh (PM) method (Hockney & Eastwood, 1988). The gravitational potential is represented on a Cartesian grid of M^3 points with position $\mathbf{X}_q = \mathbf{q}l$, $\mathbf{q} = (q_1, q_2, q_3)$. Here l is the side length of grid cell, set by the size of the box L, through $l = L/M$. The mass at each grid point is calculated by assigning contributions from the particle distribution. This is found via the density at the position of each grid cell, such that

$$\rho(\mathbf{q}) = \frac{m}{L^3} \sum_{i=1}^{N} W(\mathbf{r}_i - \mathbf{X}_q). \tag{1.108}$$

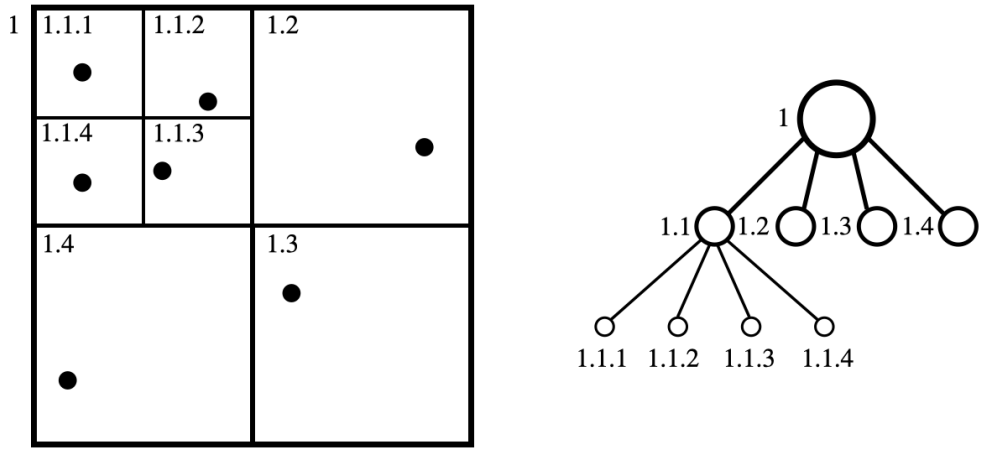

Figure 1.5 *Visual representation of a tree algorithm constructing a gravity tree, in this case showing a quadtree, the 2D equivalent to the 3D octree. Black dots represent particles, with the tree dividing up the domain such that each leaf contains, at most, one particle.*

The weighting kernel W introduced here has an integral normalised to unity. For periodic boxes, such as those used in cosmological scale simulations, the force from this grid of masses can then be efficiently found by solving the Poisson equation using fast Fourier transforms (FFT). The force at the position of a particle is then found by interpolating the force at that position from the grid

$$\mathbf{F}(\mathbf{r}_i) = \sum_{\mathbf{q}} W(\mathbf{r}_i - \mathbf{X}_q)\mathbf{F}(\mathbf{q}). \tag{1.109}$$

Here the force is found by summing over the grid points that have a non-zero kernel value. The kernel used in the density and force calculations does not have to be the same one, and there are a variety to choose from. The resultant algorithm scales as $O(N \log N)$, with spatial accuracy limited by the size of the grid cells.

Particle-Particle-Particle-Mesh

The above two methods can be combined into a Particle-Particle-Particle-Mesh (P^3M) algorithm (Hockney & Eastwood, 1988). Such a scheme uses direct summation to calculate contributions from particles closer than two grid cell lengths. The contribution from particles beyond this limit are found using the PM method. This approach combines accuracy with computational efficiency.

Gravitational Softening

A separate problem arises at small distances. The particles used in these simulations are typically treated as point masses. If the distance between two particles becomes very close to zero, in the internal units of the simulation, then the gravitational force behaves asymptotically, since force is inversely proportional to the square of the radius $F \propto r^{-2}$. The point masses therefore create diverging gravitational potentials within the simulation. A softening length is employed to prevent this (Aarseth, 1963). In the PP algorithm, this transforms the normal Newtonian force, on i from j, into

$$\mathbf{F}_i = -\sum_{j \neq i} G m_i m_j \frac{\mathbf{r}_i - \mathbf{r}_j}{\left(|\mathbf{r}_i - \mathbf{r}_j|^2 + \epsilon^2\right)^{\frac{3}{2}}}, \tag{1.110}$$

where ϵ is the softening length. This no longer behaves asymptotically as r$\rightarrow$0, removing the divergence problem. This softening length creates an effective maximum spatial resolution for the simulation. Interactions that happen at scales close to the softening length will not be physically correct, so structure formed by gravity on these scales will not be recovered. At distances much greater than the softening length $|r_i - r_j| \gg \epsilon$, the softened force is essentially identical to the full force, so it is only interactions within or close to this length that are significantly effected.

Time-Step Limitations

When numerically calculating the acceleration of a particle due to gravity, the numerical solver is effectively assuming that acceleration is acting constantly for some amount of time. The numerical scheme will only remain accurate if the time step is suitably small, such that the numerical error does not introduce catastrophic inaccuracies. In other words, the time-step must be small enough that the change in the acceleration is much smaller than both the velocity and the acceleration. This leads to a time step criterion that sets a tolerance α, with the time-step given by

$$\Delta t = \alpha \frac{\sigma}{|\mathbf{a}|}, \tag{1.111}$$

where σ is the velocity dispersion of the massive particles, and $|\mathbf{a}|$ is the net acceleration. The choice of tolerance, along with the chose numerical scheme,

effectively controls the accuracy of the simulation.

1.3 Summary

I introduce the wide variety of observed galaxy characteristics and properties, focusing on how we require advanced hydrodynamics modelling techniques, if we are to simulate their formation and evolution accurately. This includes the observations of epoch from which we draw the initial conditions for the simulations, and our understanding of the physics by which the gas can cool. I derive the fundamental equations that describe the behaviour of the baryonic gas, for which numerical solutions are required. This is followed by an overview of these numerical methods, with detailed examples for the most widely used approaches. In subsequent chapters, I present my idealised study of gaseous dynamical friction (Chapter 2), and show the derivation and extensive testing of my implementation of the residual distribution family of hydro solvers (Chapter 3). I then cover the extensions to the basic residual distribution method that I have designed and implemented, including a variable time step mechanism, and full 3D flow modelling (4). Finally I show dynamical friction work, built on that described earlier, with the addition of advanced cooling and chemistry models (Chapter 5), designed to study how the inclusion of the additional physics changes the dynamical friction results.

Chapter 2

Dynamical Friction of Satellites in Early Galaxy Evolution

2.1 Introduction

As a massive body moves through a region of lower mass bodies, gravitational interaction builds an over-density in the wake of the massive traveller. Momentum and energy are transferred from this massive perturber to the surrounding medium. This process is known as dynamical friction (DF) (Chandrasekhar, 1943), and is key to our understanding of the evolution of a number of astronomical systems at very different mass and length scales. It drives processes within structures from galaxy clusters (El-Zant et al., 2004; Kim et al., 2005; Adhikari et al., 2016), to satellite galaxies orbiting within their host halo (Zhao, 2004; Fujii et al., 2006; Ogiya & Burkert, 2016), super-massive black hole formation (Beckmann et al., 2018), compact object binaries and mergers (Just et al., 2010; Dosopoulou & Antonini, 2017; Tagawa et al., 2018), and planets within disks (Teyssandier et al., 2012).

In the case of galaxy clusters, DF plays a key role in driving the accretion of dark matter substructure. Hierarchical collapse, within the standard ΛCDM cosmological model, predicts the presence of extreme numbers of low mass sub-halos within higher mass hosts (van den Bosch et al., 2005). The number density of satellites found in simulations follows a power law, with a slope of approximately -2, giving a large number of low mass satellites (Moore et al., 1999).

These structures will experience both collisionless and gaseous DF, as they move within extended dark matter (DM) and circumgalactic medium (CGM) structure of the host. DF provides a mechanism by which they can shed angular momentum, allowing them to merge with the central structure (Boylan-Kolchin et al., 2008). In this way, DF drives the build-up of structure in a CDM cosmology. With no DF, galaxy mergers are strongly limited, only occurring from chance collisions. These mergers, in turn, are a key driver of the evolution of galaxies, providing fuel for star formation and disrupting galactic structures (Beckman et al., 2008; Khochfar & Silk, 2009; Khochfar & Silk, 2010; Robaina et al., 2010; Somerville & Davé, 2015). Gaseous DF is particularly pronounced at high redshifts, where halo gas mass fractions are highest (Daddi et al., 2010; Tacconi et al., 2010).

The structure of the wake behind a perturber can be separated into two distinct contributions, one from a collisionless medium, such as background stars or a region of dark matter, and one from a collisional medium, typically baryonic gas. The additional pressure forces present in collisional media result in significant differences in the retarding drag force, when compared to the collisionless case. This is most pronounced for Mach numbers close to $\mathcal{M} = 1$, but only when the scenario remains linear. As the scenario becomes increasingly non-linear, the gaseous DF drag force decreases (Kim & Kim, 2009). Here the linear regime refers to wakes where the over-density $\alpha = \rho/\bar{\rho} - 1$ is $\alpha \ll 1$, throughout the wake. The complex structure of the gravitationally induced wake has been studied extensively using both analytic (Just & Kegel, 1990; Ostriker, 1999; Namouni, 2010) and numerical techniques (Sánchez-Salcedo & Brandenburg, 1999; Sánchez-Salcedo & Brandenburg, 2001; Kim & Kim, 2007; Kim & Kim, 2009; Bernal & Sánchez-Salcedo, 2013).

The collisionless solution is found by calculating a sum of two body interactions, integrated over all time, between the perturber and particles that make up an infinite background medium (Chandrasekhar, 1943). The analytical solution to the collisional case can be obtained using linear perturbation theory (Just & Kegel, 1990; Ostriker, 1999). This solution applies in the linear regime, producing an approximation of the structure of the wake. The linear approximation is valid if the perturbing potential does not diverge, as this can be used to construct a scenario where no part of the wake will have over-densities greater than one. Within this linear approximation, the collisional drag force is significantly higher for perturbers moving with Mach numbers $\mathcal{M} = 0.7{-}2$. This dramatic difference is shown in Figure 2.1, (taken from Figure 3 of Ostriker, 1999), where the collisional

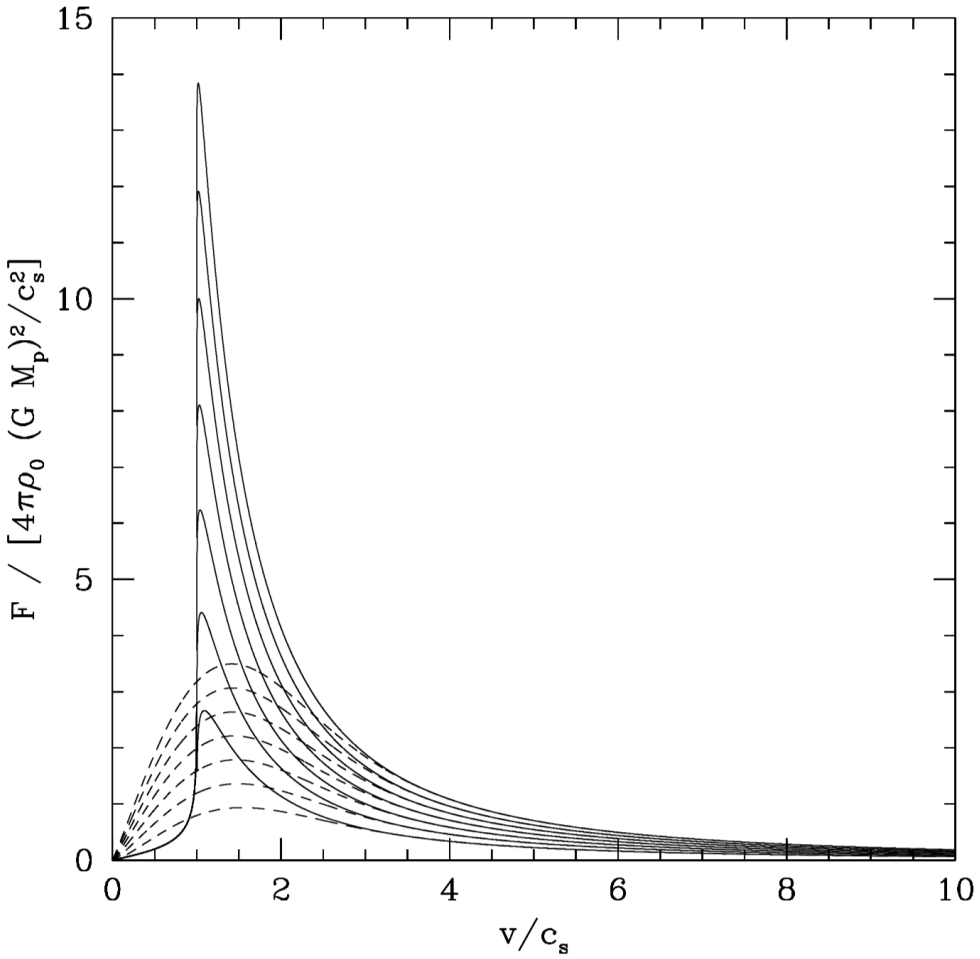

Figure 2.1 *Dynamical friction force from a gaseous medium (solid lines) for varied Mach number $\mathcal{M} = v/c_s$ (Ostriker, 1999). The equivalent forces from a collisionless medium are shown as dashed lines. The different lines show results for constant $\ln(c_s t/r_{\rm min})$. See Section 2.2 for more details on the origin of this relationship.*

results (solid lines) are compared to force from the wake in a collisionless medium.

This has been confirmed numerically for low mass, extended, perturbers (Sánchez-Salcedo & Brandenburg, 1999; Kim & Kim, 2009; Bernal & Sánchez-Salcedo, 2013). The analytic prescription can be extended numerically for more extreme mass perturbers, where large regions of the wake are no longer described by the linear wake. Kim & Kim, 2009, referred to as KK09 from here, run a set of 2D idealised setups with a moving Plummer potential embedded in an adiabatic gas. They assume cylindrical symmetry, running with (R, z) coordinates, with the perturber moving along the z-axis. The gas is modelled using a static, uniform grid of $3072 \times 12,288$ cells, requiring 5 cells per Plummer softening scale $r_{\rm s}$ to

converge with resolution, run for $t = 600t_c$. This timescale $t_c = c_s/r_s$ is the sound speed crossing time of the Plummer sphere's softening scale. They recover the Ostriker, 1999 result for perturbers in the linear regime, but find that the drag force decreases in the non-linear regime due to the development of a detached bow shock. The increased density ahead of the perturber partially counteracts the over-dense wake. At early times, a transient ring vortex also develops behind the shock, which moves downstream from the perturber. The authors present an empirical fit for the drag force from a non-linear wake that shows a continuous decrease as perturber mass increases, or softening scale decreases (see Section 2.2.4). The decrease in the force becomes dramatic in the highly non-linear regime, suggesting high mass compact objects, such as black holes, may not experience any gaseous dynamical friction. Bernal & Sánchez-Salcedo, 2013, BS13 from here on, run a similar numerical investigation, running with $\mathcal{M} = 1.5 - 4$. They use 5000×10000 cells, with 6 cells per r_s, testing both isothermal and adiabatic equations of state. They find that the drag force is independent of the equation of state, and that the KK09 solution underestimates the non-linear force at large times. A major difference between the results of these studies is in the lower bound of the force integration, with KK09 using a significantly lower value. I discuss this difference in more detail in Section 2.2.3.

More complex scenarios, including more physical processes, have also been studied, such as rigid perturbers that experience both gravitational drag, and the drag from physical collisions with the surrounding medium (Thun et al., 2016), as well as perturbers on orbital trajectories (Sánchez-Salcedo & Brandenburg, 2001; Kim & Kim, 2007), and the effect of radiative feedback from an accreting black hole on the drag force (Toyouchi et al., 2020).

The ability of modern cosmological simulation codes to accurately capture the effects of gaseous DF, in its full cosmological context, has not been studied in detail. Capturing the effects of DF in simulations requires modelling the hydrodynamic response, as well as the purely gravitational effects. The features of the hydrodynamic problem, such sharp density transitions in the supersonic case, make it difficult for traditional hydro solvers to model accurately (Tittley et al., 2001). The extended gaseous structure is highly unstructured, a challenge for Eulerian grid-based methods. Lagrangian particle-based methods, such as SPH, can better handle irregular density structures, but are less accurate at handling hydrodynamic instabilities (Agertz et al., 2007). These features are discussed in more detail in Section 2.2.2. In the last decade, a new hybrid class of methods have

been developed, utilizing moving unstructured mesh approaches, which attempt to merge the advantages of the two. They aim to provide the instability and shock capturing efficiency of the grid-based methods with the natural resolution adaptation of the particle methods. Within astrophysics, these methods are still relatively new, although a number of established codes have emerged (Springel, 2010; Duffell & MacFadyen, 2011; Hopkins, 2015; Duffell, 2016). They are ideal for studying the effects of DF, but little work has been done on this with these modern codes.

In this chapter, I will present results from a set of idealised numerical simulations I have run using the gravo-hydrodynamic code GIZMO. I summarise the derivation of the analytic solution for the gravitationally induced wake, with the net force on the perturber, in Section 2.2, and include a brief description of previous numerical results. In Section 2.3, I describe the setup of my idealised simulations, the parameter space that I explore, and how I produce the appropriate initial conditions. After this, in Section 2.4, I present the results for the numerical wake, comparing the drag force from the numerical results to those predicted by the analytic solution, for a range of Mach numbers and perturbers, showing a significant difference for Mach numbers close to unity. I discuss the implications of these results in Section 2.5, which includes comparison of our results to the conditions we would expect in typical cosmological simulations.

2.2 Analytic Solution

2.2.1 Collisionless Case

The dynamical friction force in a purely collisionless medium, such as stars or dark matter, was first estimated analytically by Chandrasekhar, 1943. This analytic solution is calculated for a set of two-body encounters between a massive perturber moving through a uniform background of less massive point mass particles. The trajectories of the particles are calculated for all time. Integrating along these trajectories provides the drag force on the perturber, which effectively transfer momentum and energy from the massive perturber to the background medium.

For a massive perturber of mass $M_{\rm p}$, moving at constant velocity V_0 through a uniform density ρ_0 medium, with no gravitational self interaction within the

medium, the dynamical friction force is (Ostriker, 1999)

$$F_{\rm DF} = -\frac{4\pi(GM_{\rm p})^2\rho_0}{V_0^2} I, \tag{2.1}$$

where, for a collisionless background medium (Binney & Tremaine, 1987),

$$I_{\rm coll} = \ln\left(\frac{r_{\rm max}}{r_{\rm min}}\right)\left({\rm erf}(X) - \frac{2X}{\sqrt{\pi}}e^{-X^2}\right). \tag{2.2}$$

Here, $X = V_0/(\sigma\sqrt{2})$, with σ as the velocity dispersion of the background particles, r_{min} is the physical extent of the perturbing object, and the maximum radius, r_{max}, is the size of the surrounding medium. The function erf(X) is the Gauss error function. This solution describes a perturber moving through a region of stars or dark matter, where there is little gas. A typical example would be a globular cluster, where only stars are present, with little to no dark matter or gas.

2.2.2 Collisional Case

The collisionless solution does not capture the behaviour of a collisional background medium, such as the gaseous circumgalactic medium (CGM). Pressure gradients, not present in a collisionless medium, introduce additional forces on the evolving gas. In the original approach for a collisionless background medium, Chandrasekhar, 1943 set the problem up as a sum of uncorrelated two body interactions. As no particle can 'see' any of the other background particles, this approach cannot capture these pressure forces.

Using linear perturbation theory, Ostriker, 1999 produced an analytic prediction for the DF force felt by a point mass perturber $M_{\rm p}$, moving at constant velocity V_0 in the z-direction through an infinite uniform density ρ_0 gaseous medium, where the gas has no self gravity. This medium has sound speed $c_{\rm s}$, so the perturber moves with Mach number $\mathcal{M} = V_0/c_{\rm s}$. Applying linear perturbation theory for an adiabatic gas, and an external gravitational potential $\Phi_{\rm ext}$, the Eulerian conservation equations that describe the perturbed density $\rho = \rho_0(1+\alpha)$ and velocity $v = c_{\rm s}\beta$ at any position $\mathbf{x}$ are (Ostriker, 1999)

$$\frac{1}{c_{\rm s}}\frac{\partial\alpha}{\partial t} + \nabla\cdot\beta = 0, \tag{2.3}$$

and

$$\frac{1}{c_s}\frac{\partial \beta}{\partial t} + \nabla \alpha = -\frac{1}{c_s^2}\nabla \Phi_{ext}. \tag{2.4}$$

These capture the conservation of mass and momentum. The α parameter represents the over-density produced in the gas. The divergence of Equation (2.4) is given by

$$\frac{1}{c_s}\frac{\partial}{\partial t}\nabla \cdot \beta + \nabla^2 \alpha = -\frac{1}{c_s^2}\nabla^2 \Phi, \tag{2.5}$$

which can be combined with the time derivative of Equation (2.3) to give

$$\nabla^2 \alpha - \frac{1}{c_s^2}\frac{\partial^2 \alpha}{\partial t^2} = -\frac{1}{c_s^2}\nabla^2 \Phi. \tag{2.6}$$

Using Poisson's equation, $\nabla^2 \Phi = 4\pi G\rho$, with gravitational constant G, one can substitute out the potential for the density of the perturber ρ_{ext}, which leaves

$$\nabla^2 \alpha - \frac{1}{c_s^2}\frac{\partial^2 \alpha}{\partial t^2} = -4\pi f(\mathbf{x}, t) \tag{2.7}$$

where $f(\mathbf{x}, t) = G\rho_{ext}/c_s^2$. This is solved using the retarded Greens function for a three dimensional wave, giving (Ostriker, 1999)

$$\alpha(\mathbf{x}, t) = \iint d^3\mathbf{x}' dt' \frac{\delta[t' - (t - |\mathbf{x} - \mathbf{x}'|/c_s)] f(\mathbf{v}', t')}{|\mathbf{x} - \mathbf{x}'|}. \tag{2.8}$$

Here, δ is the Dirac δ-function. This approach models the perturbation as the propagation of a sound wave in the adiabatic medium. The point perturber is moving at V_0 along the z-axis, and is at the origin at $t = 0$, so the function $f(\mathbf{x}, t)$ becomes

$$f(\mathbf{x}, t) = \frac{GM_p}{c_s^2}\delta(z - V_0 t)\delta(x)\delta(y)H(t), \tag{2.9}$$

where $H(t)$ defines the times when the perturber is acting on the gas. Three additional coordinates are defined as $s \equiv z - V_0 t$, $w \equiv z' - z$, and the cylindrical radius $R = (x^2 + y^2)^{1/2}$. Ostriker, 1999 shows that the integral becomes

$$\alpha(\mathbf{x}, t) = \frac{GM_p}{c_s^2}\int_{-\infty}^{\infty} dw \frac{\delta[w + s + \mathcal{M}(R^2 + w^2)^{1/2}] H((w + z)/V_0)}{(R^2 + w^2)^{1/2}}. \tag{2.10}$$

Expanding the δ-function about its roots

$$w + s + \mathcal{M}(R^2 + w^2)^{1/2} = 0, \tag{2.11}$$

to give

$$w_{\pm} = \frac{s \pm \mathcal{M}[s^2 + R^2(1 - \mathcal{M}^2)]^{1/2}}{\mathcal{M}^2 - 1}, \tag{2.12}$$

allows the integral to be solved. The component within the square root must be zero or positive for a solution to be defined. The solution is therefore divided into subsonic and supersonic cases, while $\mathcal{M} = 1$ scenarios are undefined. The negative root is undefined for subsonic cases, while both are defined for supersonic cases, as long as $s < 0$ and $|s|/R > (\mathcal{M}^2 - 1)^{1/2}$.

The solution to the integral is given by

$$\alpha = \frac{GM_{\rm p}c_{\rm s}^2}{[s^2 + R^2(1 - \mathcal{M}^2)]^{1/2}} \sum H\left(\frac{z + w_{\pm}}{V_0}\right), \tag{2.13}$$

with the H term defining the time over which the perturber is acting on the medium. To model a perturber that starts to perturb the medium at $t = 0$, a Heaviside step function is used, defined here as $H(t < 0) = 0$ and $H(t \geq 0) = 1$. The summation is introduced to handle both possible roots of the δ-function. In some scenarios both roots are non-zero, while in others, only one contributes. The summation will produce either 0,1, or 2. As mentioned above, the result is divided into subsonic and supersonic cases. For $\mathcal{M} < 1$ and $t > 0$, the solution is only defined for $R^2 + z^2 < (c_s t)^2$, which describes a sphere centered on the origin, where the perturber starts. The wave created by the perturber has not reached the region beyond this sphere, so the density remains unperturbed. For the above Heaviside function, only the positive root w_+ produces a non zero result, as the negative root gives $z + w_- < 0$. To show when H=0 for the negative root, one starts with $-z > w_-$, which expands to

$$-z > \frac{s - \mathcal{M}\left(s^2 + R^2\left(1 - \mathcal{M}^2\right)\right)^{\frac{1}{2}}}{\mathcal{M}^2 - 1}. \tag{2.14}$$

Multiplying through by the denominator of the left hand side (LHS), gathering all terms other than the square root on the right hand side (RHS), and multiplying

everything by -1, leaves

$$\mathcal{M}\left(s^2 + R^2\left(1 - \mathcal{M}^2\right)\right)^{\frac{1}{2}} > z(\mathcal{M}^2 - 1) + s. \tag{2.15}$$

One can then simply substitute in $s = z - V_0 t$, square both sides, and expand all terms. The results neatly cancel out most of the terms. The final substitution is to replace $V_0 = \mathcal{M} c_s$, which allows the Mach dependence to be removed. The above condition therefore reduces to $R^2 + z^2 < (c_s t)^2$, which means H=0 inside the spherical zone, which is the only part of the wake for the sub-sonic cases. Conversely, this also means that in the super-sonic cases, both negative and positive roots produce $H = 1$. The sum of the H factors for the sub-sonic case becomes $\sum H(t > 0) = 1$. For $\mathcal{M} > 1$, the same sphere exists, where only w_+ is defined, but there is an additional Mach cone structure, defined by the conditions provided by Equation (2.12). This again limits the wake to regions reached by the sound wave triggered by activating the perturber at $t = 0$. The cone region is therefore defined for $s/R < -(\mathcal{M} - 1)^{1/2}$. The left hand edge of the cone is set by the sphere. Within the cone region, both components of $w_\pm$ are defined, so $\sum H(t > 0) = 2$. The prediction for the over-density $\alpha(s, R, t)$ therefore becomes

$$\alpha = \frac{GM_\mathrm{p}/c_\mathrm{s}^2}{[s^2 + R^2(1 - \mathcal{M}^2)]^{1/2}} \times \begin{cases} 1 & \text{if} \quad R^2 + z^2 < (c_\mathrm{s} t)^2 \\ 2 & \text{if} \quad \mathcal{M} > 1, R^2 + z^2 > (c_\mathrm{s} t)^2, s/R < \\ & -(\mathcal{M}^2 - 1)^{1/2}, \text{ or } z > (c_\mathrm{s} t)^2 \\ 0 & \text{otherwise} \end{cases}. \tag{2.16}$$

Plots of this over-dense wake, in the subsonic case, are shown in Figure 2.2 for several Mach numbers. The plus symbols show the initial positions of the perturbers, which are moving in the positive z-direction. The cylindrical radius R and z-axis are scaled by ct, or the distance information has travelled in a static medium. For these subsonic cases, this distance from the perturber's initial position is also the maximum extent of the wake. The closed contours, those that do not intersect the edge of the wake, are symmetric about the current position of the perturber, which is at the position of the highest density. This symmetry means that those regions will not produce a net force on the perturber, since the force comes from the over-density, and these regions produce the same force in all directions. The open contours produce the entirety of the force. The fraction of the wake covered by these contours increases with Mach number, and so does the net force. The equivalent wakes for supersonic perturbers are shown in Figure 2.3. The similar spherical component, centered on the original position

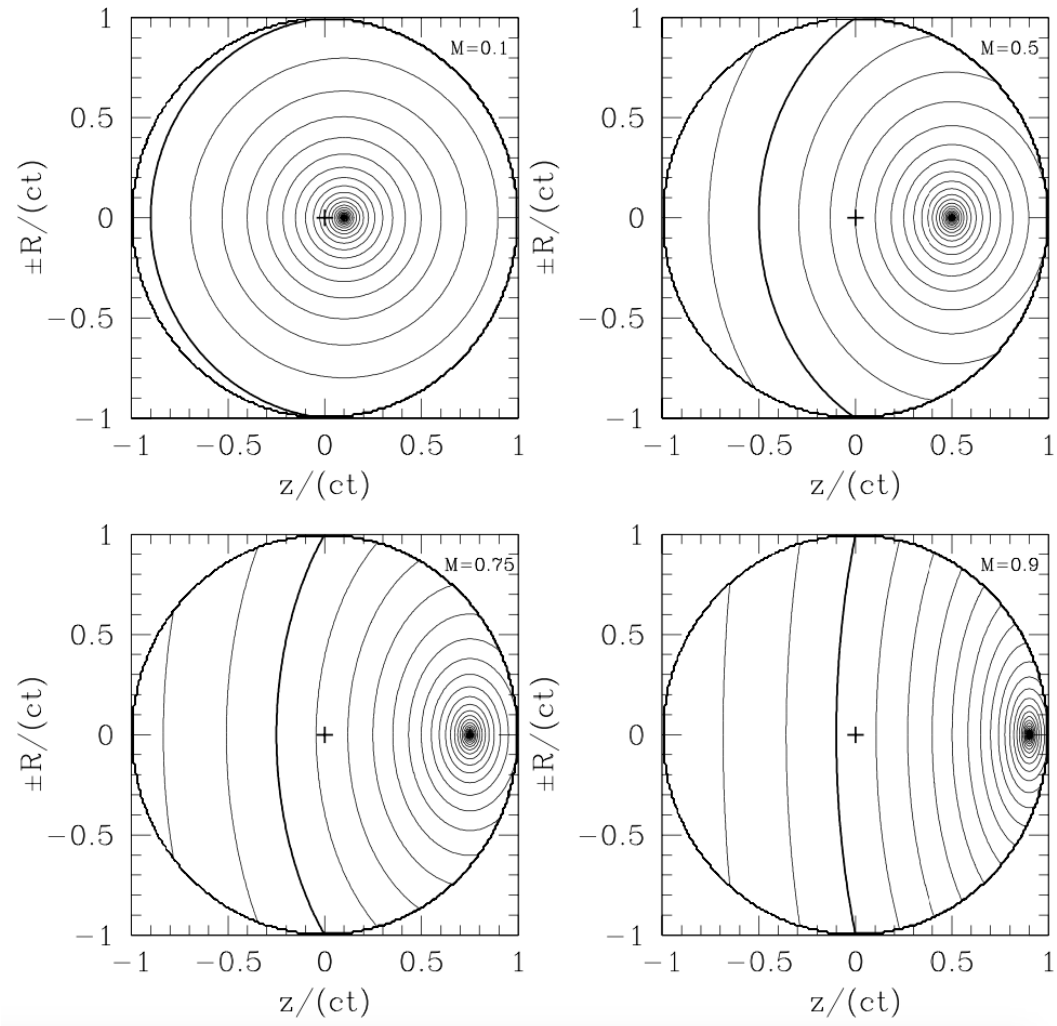

Figure 2.2 *The over-dense wake produced by linear perturbation theory for subsonic perturbers. The plus symbol indicates the initial position of the perturber, and the contours represent over-density α. The thick black line represents the edge of the perturbation. Only open contours contribute to the net force. (Taken from Figure 1 of Ostriker, 1999)*

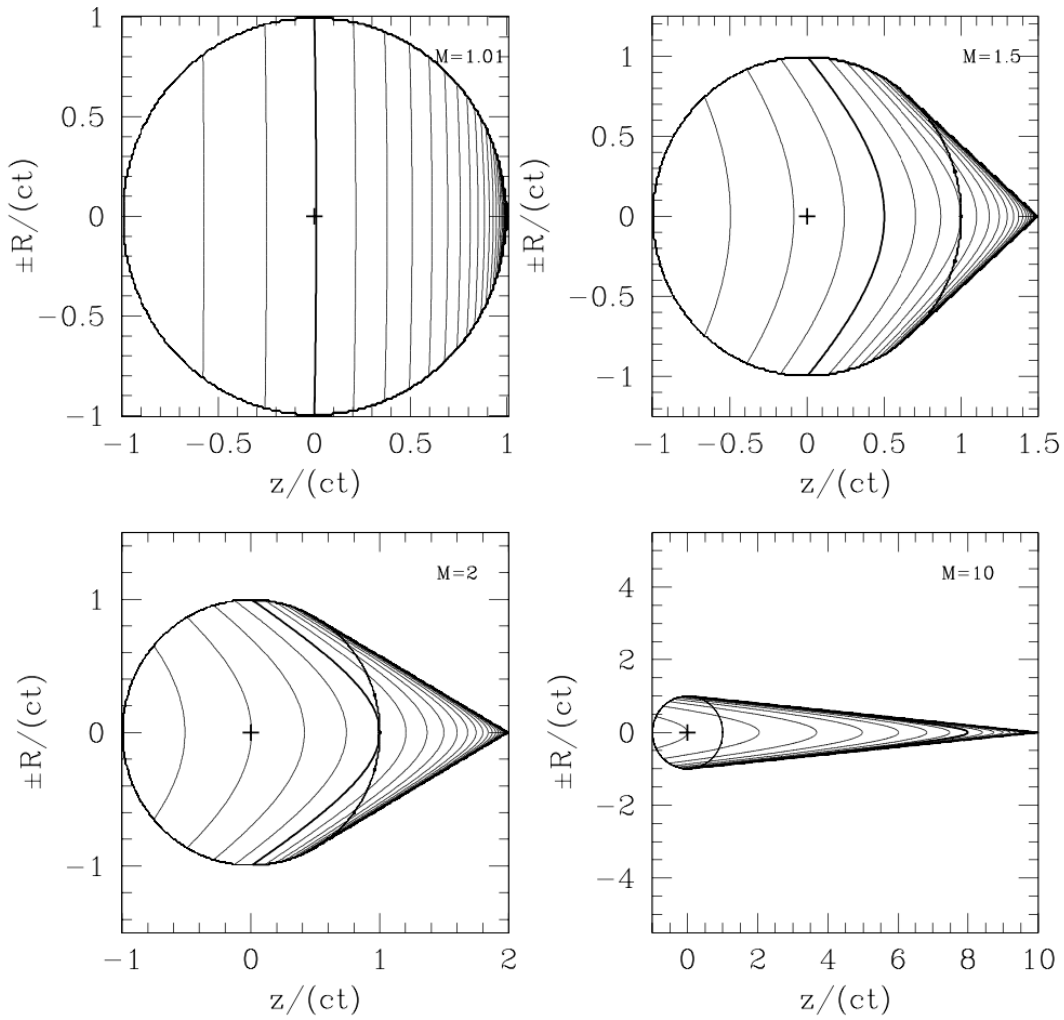

Figure 2.3 *The over-dense wake produced by linear perturbation theory for supersonic perturbers. The plus symbol indicates the initial position of the perturber, and the contours represent over-density α. The thick black line represents the edge of the perturbation. (Taken from Figure 2 of Ostriker, 1999)*

of the perturber as before, is clear. This region does not have the symmetric contours previously found in the subsonic case, as the perturber is always outside the sphere. The additional cone structure, mentioned previously, forms behind the perturber. The cone becomes more elongated as the Mach number increases. The front edge of the cone is a density discontinuity, with density decreasing backwards along the line of travel.

This distribution diverges as the spherical radius $r \to 0$. The assumptions of linearity used to derive this prediction mean that only parts of the wake where $\alpha \ll 1$ are valid. To produce an estimate of the gaseous DF drag force, a minimum radius $r_{\rm min}$ is assumed, and an integral performed over the over-density. This is done in the spherical polar coordinates, and results in

$$F_{\rm DF} = -F_0 I, \tag{2.17}$$

with

$$F_0 = \frac{4\pi (G M_p)^2 \rho_0}{V_0^2}, \tag{2.18}$$

where the I parameter represents the integral over the over-density, and depends on the Mach number of the case, with

$$I_{\rm sub} = \frac{1}{2} \ln \left(\frac{1+\mathcal{M}}{1-\mathcal{M}} \right) - \mathcal{M} \tag{2.19}$$

for the subsonic perturbers. The density profile is scale free, with no dependence on the lower limit of the integral. This scale free nature of the solution to the integral holds for $r_{\rm min} < (c_s - V_0)t$, which requires the wake is larger than the effective size of the perturber, and that the entire wake $z^2 + R^2 < (c_s t)^2$ beyond this inner edge is included in the analysis. This is effectively saying that one can take any section of the wake, within these limits, and it will give the same force, no matter the time that has passed. This is because the only contribution comes from open contours in the over-density (see Figure 2.2).

For supersonic perturbers, the I parameter is

$$I_{\rm super} = \frac{1}{2} \ln \left(1 - \frac{1}{\mathcal{M}^2} \right) + \ln \left(\frac{V_0 t}{r_{\rm min}} \right). \tag{2.20}$$

The time dependence here comes from the fact that the wake only grows downstream of the perturber, and does not fill the space around it, as is the

case for subsonic scenarios. There are no analogous closed contours around the perturbers current position, so all parts contribute a force that changes with time.

When this prediction is tested numerically, the requirement that $\alpha \ll 1$ can be parameterised using the global dimensionless parameter, A, with

$$A = \frac{GM_{\rm p}}{c_{\rm s}^2 r}. \tag{2.21}$$

This estimates the over-density at a radius r from the perturber by looking back along the path that the perturber has taken, achieved by taking $R = 0$ in Equation (2.16), which effectively leaves $r = s$. It is therefore required that $A \ll 1$ for the linear assumption to strictly hold. This parameter estimates the density perturbation at radius r. It effectively places a constraint on the physical scenarios that can be described by this prediction, since cases with very high mass perturbers, perturber with small physical extents, or very large sound speeds, will have large A values. This is discussed further in Section 2.2.4.

Any scenario can be completely described by two parameters: this A parameter, and the Mach number $\mathcal{M}$ of the perturber moving through the gaseous background medium. The Ostriker, 1999 prediction has been tested numerically, and was found to hold in the linear regime (Sánchez-Salcedo & Brandenburg, 1999; Kim & Kim, 2009; Bernal & Sánchez-Salcedo, 2013).

2.2.3 Choice of $r_{\rm min}$

The net drag force is computed by integrating over the over-dense wake. An equivalent integration can be made over the output from a numerical simulation to produce the numerical drag force. The choice of $r_{\rm min}$ for these calculations is not clear cut. In the analytic case it is constrained by the conditions described above, when only considering the linear regime.

For the numerical integration of the simulated wake, different authors have settled upon various strategies to set $r_{\rm min}$. Assuming a potential with gravitational softening r_s is used for the perturbing potential (see Section 2.3.1), the minimum radius can be parameterised based on this softening. A scale free time can also be defined based on this scale. The sound speed crossing time t_c of the softening scale allows for comparison between different setups. Sánchez-Salcedo & Brandenburg, 1999 empirically find that $r_{\rm min} = 2.25 r_{\rm s}$ accurately produces the predicted forces,

which they show is analogous to the minimum stellar radius found for a Plummer potential. BS13 estimate the value of $r_{\rm min}$ using the position of the maximum density in the wake produced by a Plummer sphere, as opposed to a point mass, finding a value consistent with the Sánchez-Salcedo & Brandenburg, 1999 result. They confirm this value empirically with their numerical results. KK09, on the other hand, develop a Mach dependent prescription, with $r_{\rm min} = 0.35\mathcal{M}^{0.6}r_{\rm s}$ providing the best fit for the force at $A = 0.01$, and argue that the difference between their result and previous ones is caused by differences in resolution and equation of state. BS13 address this suggestion, and find no difference in results between isothermal and adiabatic equations of state, nor with resolution. They find an equivalent Mach dependent fit with $r_{\rm min} = 1.5\mathcal{M}^{0.6}r_{\rm s}$, but conclude that it is only applicable at later times $t > 130t_c$. Both fits discussed by BS13 use a value significantly larger than that found by KK09, which has a dramatic effect on absolute value of the drag force from the wake. BS13 note that the KK09 fit can produce a force as much as a factor of two larger than their own. The choice of $r_{\rm min}$ is extremely important when considering the drag force in more complex scenarios, where the wake may be disrupted at large distances. The best fit value for $r_{\rm min}$ effectively shows us the radius from which the wake matches the analytic result, and so also shows us where differences in the wake may reside. It is not obvious how the minimum radius should be chosen to ensure a physically consistent result.

It should be noted that the softening scale $r_{\rm s}$ discussed here still refers to the scale length of the Plummer sphere used to represent the perturber, not some underlying gravitational softening of the simulation, which does not apply to this gas, since there is no self-gravity. I systematically investigate what radius can provide the equivalent fit for our numerical results, and discuss how the detailed structure of the numerical wake brings this about, in Section 2.4.2.

2.2.4 Non-Linear Regime

The linear regime does not adequately describe many important astrophysical systems, where $A \gg 1$, particularly compact perturbers such as proto-planets in proto-planetary disks (Muto et al., 2011, Bromley & Kenyon, 2016) or black holes (Beckmann et al., 2018; Dosopoulou & Antonini, 2017). For such perturbers, the condition that the over-density is small everywhere in the wake is not maintained. The perturbing potential creates regions where this assumption breaks down,

leading to the formation of structures not covered by the analytic prediction. A detached bow shock is found to develop upstream of the perturber, and a ring vortex is observed to detach and move downstream (Kim & Kim, 2009; Bernal & Sánchez-Salcedo, 2013). In isolation, the bow shock would actually accelerate the perturber, so when the bow shock structure is combined with the over-dense wake that sill develops, the net effect of the non-linear wake is a reduced drag force on the perturbing mass. The vortex structure KK09 find at early times will exacerbate this effect early on, as the low density in the vortex further reduces the drag force. The most dramatic differences are observed where the gaseous DF is predicted to diverge most from the collisionless case, with $\mathcal{M} = 1 - 2$. The key difference between the linear and non-linear cases comes in the development of this bow shock. It sits just upstream of the perturber, and disrupts the flow in the near wake. The presence of such structure will therefore be found in the inner most radii, close to the perturber. One therefore expects that the analytic wake will not be reproduced in the innermost regions for $A > 1$, where the non-linear structures will dominate. These features should not be present in cases with $\mathcal{M} \leq 1$, where the smoothed potential should produce an over-density $\alpha \ll 1$ at all positions (see Equation (2.16)).

Numerical experiments have been used to find empirical fits for the force from a perturber in the non-linear regime. KK09 produce a parameterisation of the reduced force in the non-linear regime using

$$F = F_{\rm lin} \left(\frac{\eta}{2}\right)^{-0.45}, \tag{2.22}$$

when $2 < \eta < 100$, where $F_{\rm lin}$ is the Ostriker linear force and

$$\eta = \frac{A}{\mathcal{M}^2 - 1}. \tag{2.23}$$

BS13, on the other hand, find that the time dependence in the supersonic solution has the non-linear force converging to the linear solution at large times. The non-linear structures, by definition, form close to the perturber where the gravitational field is strongest. The far-field part of the supersonic wake contributes the time dependence, and is undisturbed by the non-linear formations. As the wake grows continuously with time, the relative contribution to the net force from the undisturbed part of the wake increases, and the relative non-linear contributions shrink. KK09 present their results for large times ($t = 600 t_c$), so should also find this evolution. No analytic solution exits for the non-linear case, so it is not clear

exactly what solution should look like, in this case.

2.3 Numerical Method

In this section, I cover my numerical approximation of the idealised case, and the key parameters that describe the initial state. I run a set of idealised, gravo-hydrodynamic simulations using the multiphysics astrophysical code GIZMO (Springel, 2005; Hopkins, 2015). I cover the specific setup that I use, the details of the main simulations that I am showing results for, and the process of producing the initial conditions. These were run on medium to large scale high performance computing (HPC) clusters, requiring significant computing time.

2.3.1 Setup

The numerical setup is designed to closely replicate the assumptions made in the derivation of the analytic solution. The initial conditions have a uniform density ρ_0 adiabatic gas, moving with bulk velocity V_0, in a box with a fixed gravitational potential Φ from a massive perturber M_p. The bulk velocity has Mach number $\mathcal{M} = V_0/c_s$. The gas does not experience self gravity, and the boundaries of this box are periodic. The massive perturber has the form of a Plummer sphere (Plummer, 1911), with potential

$$\Phi(r) = -\frac{GM_\mathrm{p}}{\left(r^2 + r_\mathrm{s}^2\right)^{\frac{1}{2}}}, \tag{2.24}$$

with mass M_p and softening r_s. This softening is analogous to the extent of the perturbing object. The simulation time is characterised by sound speed crossing time of this length $t_c = r_\mathrm{s}/c_\mathrm{s}$.

To reproduce the initial uniform density, I generate glass-like initial conditions (ICs) using WVTICs (Donnert et al., 2017; Arth et al., 2019), which combines a weighted Voronoi mesh with a particle shuffling method to reproduce arbitrary density distributions with a set of SPH particles. The ICs created with this method were found to be superior to simple initial grids and random particle placements, minimising periodic oscillations in the net force on the perturber, and reducing initial variation in the local density distribution. This is discussed further in Section 2.3.3.

The idealised setup is effectively scale free, characterised by A (Equation (2.21)) and the Mach number $\mathcal{M}$. The A parameter sets the relationship between the perturber, from its mass and extent, and the medium, through the sound speed. The Lagrangian nature of the methods used here mean spatial resolution comparisons are not straight forward. The kernel lengths in a typical run were chosen to be of order $0.1r_s$, compared to 1 cell per r_s (Sánchez-Salcedo & Brandenburg, 1999), 5 cells per r_s (Kim & Kim, 2009) and 6 cells per r_s (Bernal & Sánchez-Salcedo, 2013) in previous 2D grid works. Our estimated spatial resolution is therefore on a similar level to these previous works. I use $r_{min} = r_s$ as the default choice, but I also explore radial break downs of the net force that show what r_{min} would produce a good fit, for different cases, in Section 2.4.3.

The relative motion is captured by the Mach number. I run setups for a range of Mach numbers, exploring the regime where collisional DF differs most from collisionless DF, from $\mathcal{M} = 0.7$ to $\mathcal{M} = 2.0$. A range of A values was also explored, from far inside the linear regime ($A = 0.01$) through the transition regime ($A = 1$) and beyond, into the highly non-linear regime ($A = 10$). A number of different box sizes and resolutions were used. The different setups are summarised in Table 2.1.

Each box was run until the wake reached close to the edge of the periodic box. If it were run longer, the edge of the over-dense wake would rap around. Larger boxes allow for longer runs, but at the cost of reduced mass resolution, critical to recovering the small over-densities in the wake structure. Our setup allows us to reach $t = 15t_c$ in the standard runs, and up to $t = 150t_c$ in the largest boxes, while still retaining acceptable mass resolution.

2.3.2 Solvers

I investigate the differences in the numerical results for the Lagrangian meshless finite mass (MFM) (Hopkins, 2015) and pressure-entropy smoothed particle hydrodynamics (P-SPH) (Hopkins, 2013) hydro-solvers. The MFM solver uses a state-of-the-art method that discretises the volume with a kernel function and solves the Riemann problem between the neighbouring particles, utilising a high order gradient estimator. The P-SPH solver is an extension to the standard SPH formulation (Gingold & Monaghan, 1977). It estimates pressure gradients using entropy rather than density, reducing inaccuracies at fluid interfaces. The

Table 2.1 *Details of select the simulation runs, listing the hydro solver, the box size L, the number of particles N, the Mach number $\mathcal{M}$, the A parameter, and the initial temperature T of the background gas.*

ID	Solver	L(kpc)	N	$\mathcal{M}$	A	T (K)
ML10N8MA07A01	MFM	10	512^3	0.7	0.1	T_0
ML10N8MA09A01	MFM	10	512^3	0.9	0.1	T_0
ML10N8MA101A01	MFM	10	512^3	1.01	0.1	T_0
ML10N8MA11A01	MFM	10	512^3	1.1	0.1	T_0
ML10N8MA13A01	MFM	10	512^3	1.3	0.1	T_0
ML10N8MA15A01	MFM	10	512^3	1.5	0.1	T_0
ML10N8MA2A01	MFM	10	512^3	2	0.1	T_0
ML10N8MA07A1	MFM	10	512^3	0.7	1	T_0
ML10N8MA09A1	MFM	10	512^3	0.9	1	T_0
ML10N8MA101A1	MFM	10	512^3	1.01	1	T_0
ML10N8MA11A1	MFM	10	512^3	1.1	1	T_0
ML10N8MA13A1	MFM	10	512^3	1.3	1	T_0
ML10N8MA15A1	MFM	10	512^3	1.5	1	T_0
ML10N8MA2A1	MFM	10	512^3	2	1	T_0
ML10N8MA13A01	MFM	10	512^3	1.3	0.01	T_0
ML10N8MA13A10	MFM	10	512^3	1.3	10	T_0
PL10N8MA13A01	PSPH	10	512^3	1.3	0.1	T_0
PL10N8MA13A1	PSPH	10	512^3	1.3	1	T_0
ML100N9MA13A01	MFM	100	1024^3	1.3	0.1	T_0
ML100N9MA13A1	MFM	100	1024^3	1.3	1	T_0
ML10N8MA13A01LT	MFM	10	512^3	1.3	0.1	$0.1T_0$
ML10N8MA13A1LT	MFM	10	512^3	1.3	1	$0.1T_0$

idealised DF problem has not been investigated in detail using these modern Lagrangian solvers, so our understanding of their detailed behaviour in this context is currently limited. DF provides a further test for the numerical accuracy of these schemes.

2.3.3 Initial Conditions

The highly idealised setup requires initial conditions (ICs) with a uniform density ρ_0 gas, moving with some bulk velocity V_0, at Mach $\mathcal{M} = v/c_s$. This setup needed to be replicated with a distribution of SPH particles, each with fixed mass. To reproduce the initial uniform density across the whole box, a number of methods were used.

Random Sampling

The most straight forward approach is to randomly sample N positions within the limits of the box, placing the fixed mass particles at these positions. This approach produces a box which, by definition, has an average density of the desired value. A first set of simulations were run using ICs created with this method. The random sampling inevitably produces local variation in the density structure, with some particle positions having very high initial densities and pressures, where particles have been randomly placed very close together. In Figure 2.4, I show the initial density PDFs for the different IC setups. The blue histogram shows the long tail of particles with high density for the random ICs discussed here. While these high density particles are relatively few in number, I decided they could be responsible for a mismatch between the analytic prediction and the numerical results (see Section 2.4). To combat this, I explored other possible initial particle distributions.

Grid

To avoid the large initial range in local densities, I also run a number of simulations using an alternative, grid based, initial setup. Although the density distribution in the random setup gradually relaxes toward the desired uniform background, it was possible this difference from the analytic problem is responsible for the discrepancies in the results (see Section 2.4). Allowing the ICs

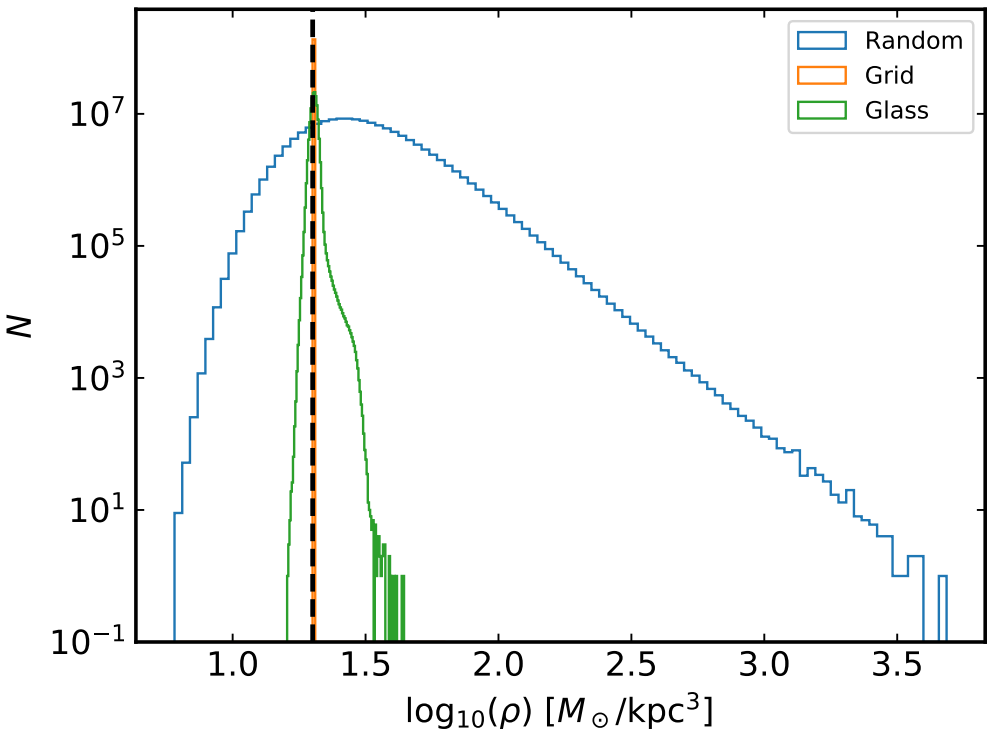

Figure 2.4 *Distribution of initial densities of all particles for the different IC setups. The densities at the location of each particle are calculated using the cubic weighting kernel used by the hydro solvers. The random case is shown in blue, the grid based ICs in orange, and the glass-like setup in green. The vertical dashed line shows the desired value for the uniform background density. Unsurprisingly, the random approach shows the largest variation in local density, while the grid perfectly reproduces the exact density. The glass ICs represent a compromise, as they have significantly less variation than the random case, without the periodic oscillation in the force found with the grid ICs.*

to relax before applying the perturber is a possibility, but this will be discussed later. To guarantee that the initial density was the same everywhere, I moved to a setup where the particles were placed on the vertices of a Cartesian grid. Since all particles now have identical distances to their neighbours, the density is the same at every particle position (orange histogram in Figure 2.4). This worked effectively for some scenarios, but it became clear that for cases with low perturber masses (for testing A numbers far into the linear regime), there was a periodic oscillation in the force that washes out the signal from the wake completely.

Any initial particle distribution will introduce an underlying variation in the net force exerted on the central massive perturber as it flows past. In the grid-based case, as the distribution moves with its initial velocity, one face of particles moves away, while the other moves towards the perturber. The distribution is asymmetric about the centre of the box, producing a spurious net force. This force will vary periodically as the faces of the initial grid move past the centre. In Figure 2.5, I show this periodic variation over a short time in blue, with the vertical black dashed lines indicating the times at which the distribution recreates its initial state. This force is normalised to a dimensionless form using Equation (2.18), dividing the force by $\mathcal{M}^2 F_0$. This is the scale free DF force, independent of the specific conditions of the scenario. In this case I have created the initial distribution to have the maximum variation at the start, as a test of this variation. The standard grid setup would start with the distribution symmetrically arranged about the centre, giving zero initial force. The oscillation of the force with time would be the same. The amplitude of this variation in the force is inversely proportional to the perturber mass. This relationship comes from the combination of Newtonian gravitational force between a particle and the perturber, proportional to the mass of perturber M_p, and the normalisation, which is proportional to M_p^2. Dividing the gravitational force by the normalisation leaves the dimensionless force dependent on M_p^{-1}. The variation in blue is for the $A = 0.01$ case, while the $A = 0.1$ case is shown in orange. The $A = 0.1$ case is obtained by increasing the perturber mass by an order of magnitude, for which the force oscillation will be an order of magnitude less. If one aimed to probe smaller A numbers, while avoiding the force variation increase, one could also increase the sound speed of the gas. The wake would expand faster in this case, and so to reach the same time without the wake wrapping around the periodic boundary conditions, the box would have to be made larger. This, in turn, would mean the particles would have greater mass, to achieve the uniform density with the same

number of particles, and so the force variation would once again increase. This variation will be discussed again later in this section.

A similar oscillation in the force is present with the random initial conditions, but the grid ICs produce the maximum possible variation for a given perturber mass - particle number setup. I found that the grid based initial conditions are viable for $A \geq 1$, but not for those with $A < 1$, where the perturber mass is lower, and the variation proportionally higher.

Glass

A compromise between achieving the perfect uniform density and avoiding the overwhelming periodic oscillation in the force is found by using glass-like ICs, generated with the WVTICs algorithm (Arth et al., 2019). This uses a combination of a weighed Voronoi tessellation (WVT) with a particle shuffling algorithm to reproduce any arbitrary density distribution with a set of fluid tracer particles. The WVT algorithm creates a low energy, relaxed, glass-like particle distribution, without explicitly identifying the exact form of the Voronoi tessellation. To achieve this, a standard SPH wrapper is used to calculate the densities and kernel lengths of each SPH particle. Other standard SPH quantities, such as internal energy, are not used in any part of the algorithm. The desired density $\rho^m(\vec{r})$, or model density, is used to calculate how far a given particle i should be moved $\Delta\vec{r}_i$, based on the SPH kernel function $W(|\vec{r}_{ij}|, h^m_{ij})$, the distance between the particles i and j, and an averaged weighted kernel length h^m_{ij}, such that

$$\Delta\vec{r}_i = h^m_{ij} \cdot W\left(|\vec{r}_{ij}|, h^m_{ij}\right) \cdot \frac{\vec{r}_{ij}}{|\vec{r}_{ij}|}. \tag{2.25}$$

The weighted smoothing length h^m_i for a given particle is an estimate of the smoothing length that should be present at that position, if the desired density distribution was in place. This is calculated using

$$h^m_i = \left(\frac{1}{\frac{4}{3}\pi} \frac{\rho^m_i}{\sum_j \rho^m_j}\right)^{1/3}, \tag{2.26}$$

and the averaged value, used in the calculation of the displacement $\Delta\vec{r}_i$ from particle j, is the arithmetic mean of the modified lengths h^m_i and h^m_j. This shifting of the particles will create the desired relaxed, low energy state. The shifting

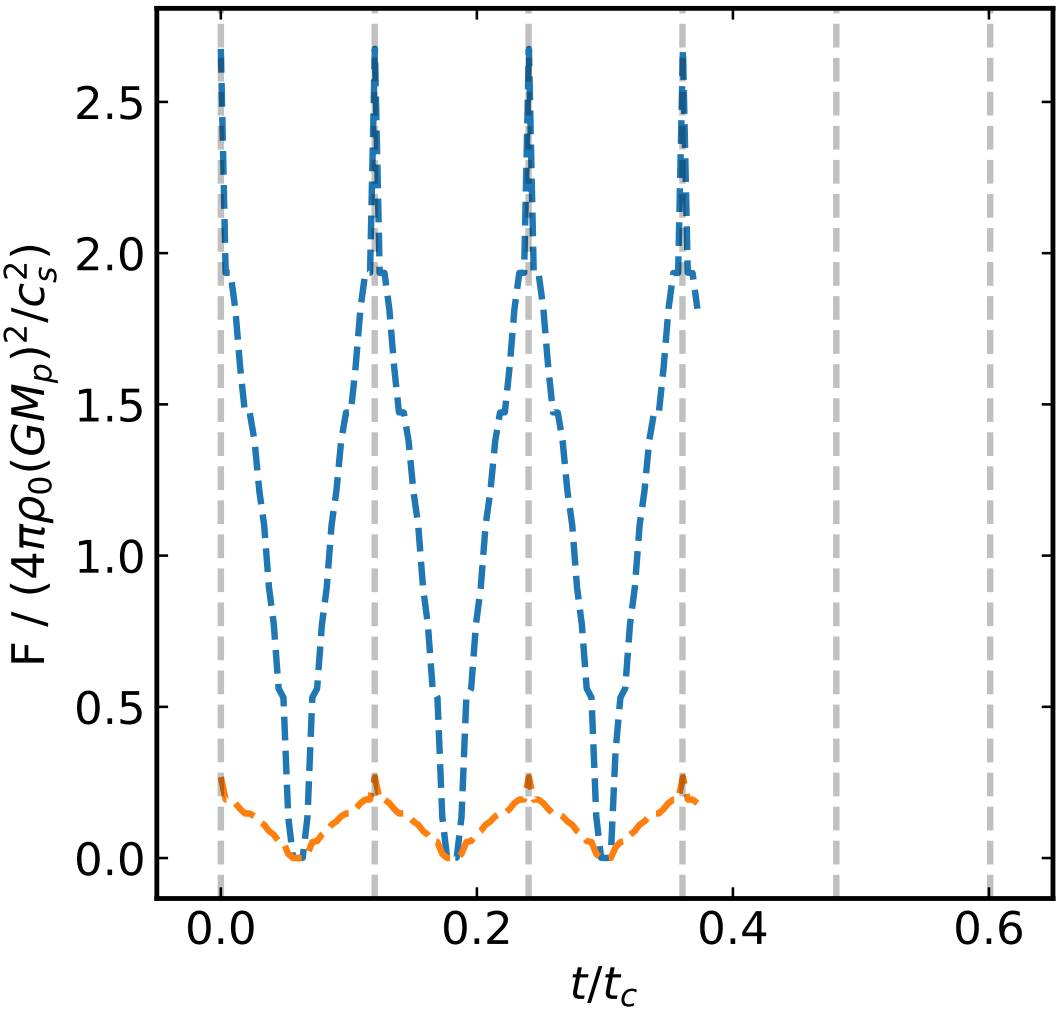

Figure 2.5 *Oscillation in the force from the grid based initial conditions on the* $A = 0.01$ *case perturber (blue), and the* $A = 0.1$ *perturber (orange). The grey dashed lines show the times when the distribution will be in the same position as the start. The evolution is calculated without any gravity from the perturber acting on the background gas, so all variation in the force is caused by the unperturbed bulk motion of the initial particle distribution moving past the position of the non-acting perturber.*

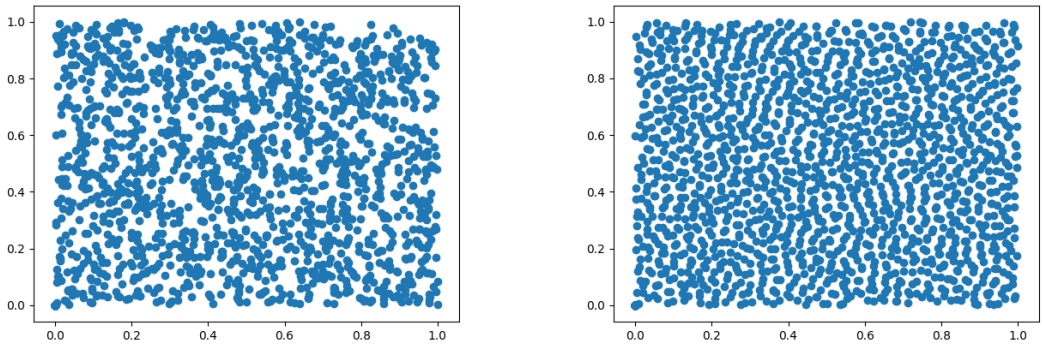

Figure 2.6 *Left: Initial distribution of particles with positions randomly drawn by the WVTICs algorithm. Right: Distribution of particles after 256 iterations of the WVTICs algorithm.*

process is repeated until most particles have moved a set fraction of the desired particle separation. The pushing of the particles acts only locally, as SPH forces are fundamentally local, so an additional redistribution step is used. This moves particles from regions where there are too many particles to parts where there are too few.

The whole process of relaxation and redistribution is repeated a set number of times, converging on an optimal solution. In this case 256 iterations were used. An example of the improved distribution is shown in Figure 2.6, with a 2D slice through the initial distribution of particles on the left, and the final distribution after 256 iterations of the WVTICs on the right. The improvement in uniformity is clear by eye, without the presence of any periodic grid-like structures. This approach is equivalent to allowing the random IC case to relax, before applying the gravitational perturber. Performing this action with GIZMO, instead, could produce a similar effect, but would be less computationally efficient, as GIZMO is not tailored towards producing specific distributions, and so does not include the redistribution step. This improvement is also clear in Figure 2.4, with the difference in distribution of densities between the purely random spatial distribution (blue), and the glass-like ICs (green). The glass ICs show a much smaller range of densities, minimising any spurious effects these may have on the flow.

Intrinsic Force Variation

The A parameter, which effectively sets the linearity of the numerical setup, can be varied by changing any of its dependent variables: the perturber mass, the sound speed of the background gas, and the gravitational softening scale of the perturber. Setups with the same A parameter constructed with different combinations of these parameters should produce the same results. However, there is an intrinsic error in the force produced by a given setup, effectively dictated by the inner radius, which forms the lower limit of the force integral, given in Equation (2.1), the mass of the perturber M_p, and the mass of the gas particles $M_{\rm part}$. If we consider a scenario where the perturber does not act on the medium, then as the initial distribution of particles moves past the perturber position, the configuration of finite masses will produce variation in the net force on the perturber. This force from the unperturbed medium should be zero, but the sampling of the medium by finite mass particles leads to some variation from this value. The lower limit of the integral sets the distance of the nearest particle included in the force summation. Small lower limits lead to large contributions from the particles very close to the perturber, where the small volumes are sampled by few particles. Large $r_{\rm min}$ leads to smaller variation. For a given background density and integral lower limit, the ratio of the particle mass to perturber mass $M_{\rm part}/M_{\rm p}$ has a significant role in determining the level of intrinsic variation. The larger this ratio is, the more severe the impact of the closest particles to the perturber.

The improvement in the net force on the perturber from the initial distribution of particles is shown in Figure 2.7 for the different setups. The normalised force is calculated by direct summation of the Newtonian gravitational force between the central mass and the gas particles, excluding particles within $r_{\rm min}$ of the centre. The normalisation is applied to the net force. This effectively shows the uncertainty in the force for different choices of initial condition setup. One hundred random shifts are drawn, and the force calculated for a range of inner radii $r_{\rm min}$. The dots show the mean of these forces for each setup, with the standard deviation marked by the error bars. This gives an estimate of the maximum underlying variation in the force that a given IC setup will produce. The glass-like ICs show a significant improvement over the random and grid ICs, demonstrating they will be able to probe the lowest A cases. These results are for the $A = 0.01$ case, but the same pattern is observed for other A values, simply shifted along the y-axis by a constant force.

Both the random and glass ICs show the net force on the perturber decreasing as the minimum radius of the force integral increases. The force offset, in these two cases, is dominated by the particle closest to the perturber, but outside this minimum radius. As the radius is increased, this closest particle will be further away. Gravitational force obeys an inverse square law, so the offset in the force will decrease with $1/r^2$. In the grid based IC case, the force does not decrease in this way. There is a constant offset to the net force. This is caused by the systematic asymmetry in the particle distribution. It is constant, because the asymmetry comes from the whole grid of particles. Simply removing the innermost particles does not change the fact that the asymmetry is still present at large radii.

If one considers the increase in the inner radius of the integral as removing a shell of particles from the integration, then the net force will remain the same if one removes the same number of particles in each direction, which is the case when the particle spacing is very small. This is the case here. Increasing $r_{\rm min}$, in the grid IC case, removes approximately the same number of particles from all directions, because of the small particle separation. These particles originally effectively cancelled one another out, so the removing them does not change the net force. In this way, it is clear that the net force on the perturber in the grid case in fact comes from the outermost part of the grid.

Figure 2.8 shows the force on the central massive perturber from the glass like initial conditions. The specific cases shown here are for the $M_{\rm p} = 2.5x10^3 M_{\odot}$ perturber and $N = 512^3$ particles, with L = 1, 10, 100 kpc. This corresponds to the $A = 0.1$ scenario. The ratios of particle mass to perturber mass are 6×10^{-11} (blue), 6×10^{-8} (orange), and 6×10^{-5} (green) respectively.

The numerical dimensionless force increases linearly with the ratio of particle mass to perturber mass, for a given background density. We can estimate the underlying uncertainty in the force from the output of a given setup. This intrinsic variation effectively limits the setups that will produce useful results, because the error can be too large to say anything meaningful about the force from the resultant wake. We must balance the need for a large box to reach large numbers of crossing times with the need to keep the number density high.

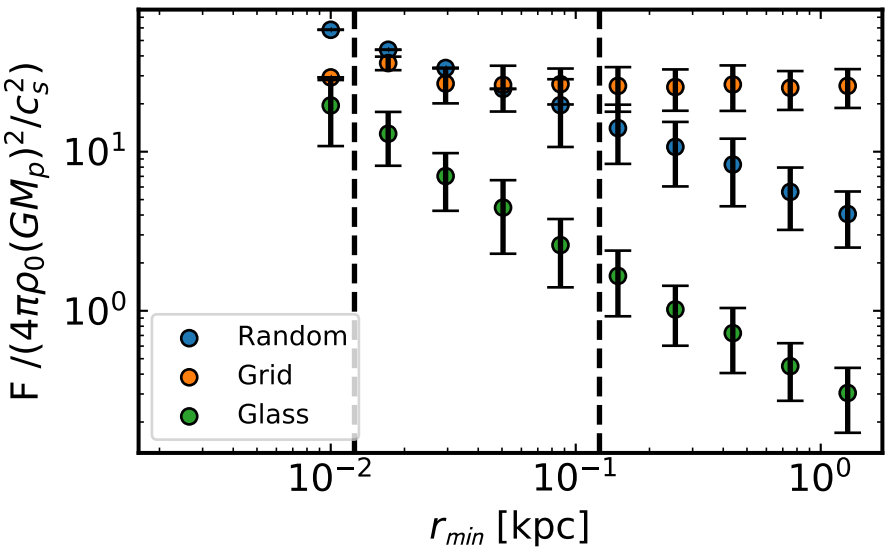

Figure 2.7 *Dimensionless force on the perturber from the random (blue), grid (orange), and glass (green) ICs. Setups are randomly repositioned about the perturber, effectively creating 50 different realisations of the same approach. The force is then calculated for different value of minimum radius* $r_{\rm min}$*. This shows the underlying variation in the force that would be expected from the discretisation of the background medium. The glass ICs show the smallest underlying oscillation, while the random ICs show significantly more. Both show a decrease in the force with increasing* $r_{\rm min}$*. The grid based ICs do not show this decrease.*

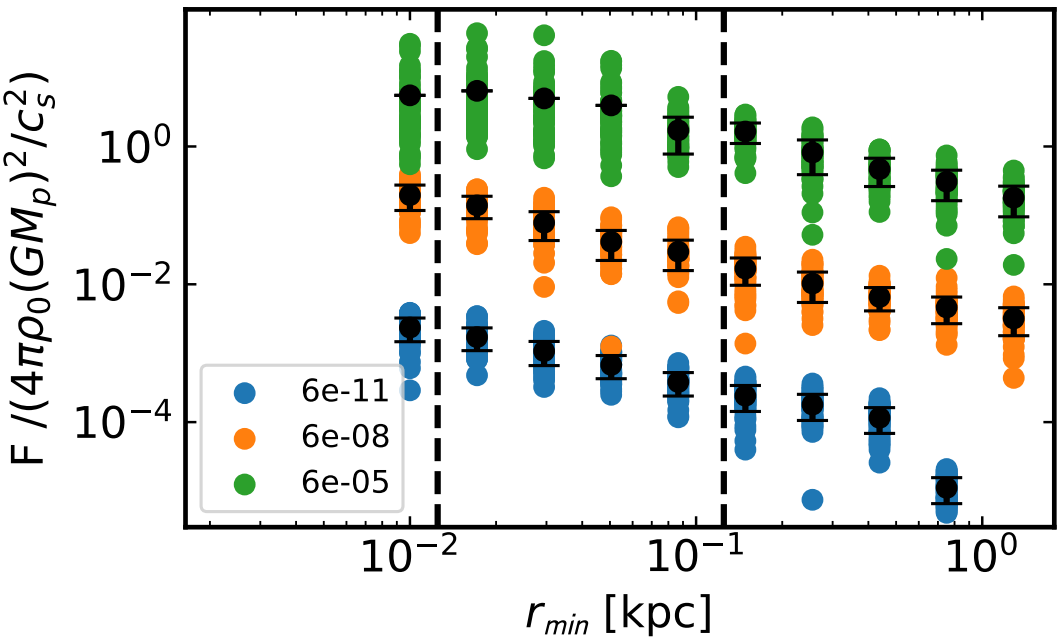

Figure 2.8 *Force on a* $M_{\rm p} = 2.5x10^3 M_\odot$ *perturber from the initial particle distribution for setups with mass ratios of* $M_{\rm particle}/M_{\rm perturber} = 6\times10^{-11}$ *(blue dots),* 6×10^{-8} *(orange dots),* 6×10^{-5} *(green dots). ICs are randomly shifted 50 times for each tested* $r_{\rm min}$*. The coloured dots show the net dimensionless force on the perturber at the centre of the box, excluding all particles within* $r_{\rm min}$ *of the perturber. The black dots show the mean force at that* $r_{\rm min}$*, with error bars showing the standard deviation in the forces. The variation rises significantly with particle-perturber mass ratio, and falls as* $r_{\rm min}$ *is increased. The variation shown here effectively limits the scenarios that can be run with these solvers.*

2.4 Results

In this section, I compare the numerical results from these idealised simulations to the analytic predictions for the force produced by this wake on the perturber. I show the time evolution of this force, as well as the force broken down into contributions from spherical shells, and compare the results from the two solvers. Results are from runs using the MFM solver, unless otherwise stated. The analytic solution strictly holds for $A \ll 1$, but previous works have shown the predicted force can be recovered for $A \leq 1$ (Kim & Kim, 2009). I therefore compare results for this regime to the analytic solution. I also include one result from a highly non-linear case, to test the behaviour beyond the linear cases.

To test the behaviour of the numerically modelled gas under different conditions, it is necessary to explore the scale free parameter space of A and Mach number. The setup is constrained by the periodic boundary conditions, setting an upper limit on how long the scenario can be run before the wake effects reach the edge of the box and wrap around. This time is defined by the edge length of the box and the initial velocity of the gaseous background. A range of Mach numbers were probed, exploring the regime around the strongest gaseous signal at $\mathcal{M} = 1$. The majority of the boxes are run for $t = 15t_c$, with the larger boxes pushed to $t = 150t_c$.

2.4.1 Force

The analytic prediction for the net force from the wake on the perturber is given by Equation (2.1). In the subsonic case, this solution to the integral holds for $r_{\min} < (c_s - V_0)t$, which requires the wake is larger than the effective size of the perturber. For supersonic cases, the solution is found by assuming $r_{\min} < (V_0 - c_s)t$, which means that the lower limit of the integral is inside the cone section of the wake. These limits only restrict the use of the analytic integral for the net force. The prediction for the over-density α is valid, even when these restrictions are not met. Therefore, a numerical integration of the analytic over-density α was used at times when these conditions were not satisfied. To achieve this, the wake is broken down into cone and sphere sections, and the solution calculated separately for each. This allows us to produce an accurate force for the analytic solution no matter the limits of the integration. The equivalent

force from the simulation outputs is calculated by direct force summation. The Newtonian gravitational force is calculated between each particle of mass $M_{\rm part}$, and the massive perturber $M_{\rm p}$. All particles within $r < r_{\rm min}$ of the perturber are excluded from the calculation.

The force is dependent on the Mach number and time. In Figure 2.9, I show the variation in the numerical and analytic forces with Mach number for two sets of runs at $t = 15t_c$, one with $A = 0.1$ and the other with $A = 1$. This is the largest time reached by the standard box. After this point, the wake will reach the edge of the simulation box and wrap around ahead of the perturber. I see that both $A = 0.1$ and $A = 1$ show numerical forces less than the analytic prediction for supersonic cases, with up to half the force missing close to $\mathcal{M} = 1$ for $A = 1$. The maximum missing force for $A = 0.1$ is also substantial, with up to a quarter of the predicted force not recovered in the numerical solution, and in the same Mach regime. The subsonic setups, on the other hand, show good agreement at $\mathcal{M} = 0.7$, with some divergence at $\mathcal{M} = 0.9$. The $A = 0.1$ case shows a smaller residual (Figure 2.9, lower panel), while the $A = 1$ case is worse at Mach numbers $0.7 < \mathcal{M} < 1.5$. The difference is most extreme in the supersonic regime close to $\mathcal{M} = 1$. This shows that the agreement gets worse as the setups become less linear.

The trend with linearity is also shown in Figure 2.10, where I show the evolution of the force with time, where $\mathcal{M} = 1.3$, for $A = 0.01$, $A = 0.1$, $A = 1$ and $A = 10$. The most non-linear case $A = 10$ shows extreme deviation from the linear prediction. This plot also demonstrates the increase in variation of the force as the ratio of particle mass to perturber mass increases (see Section 2.3.3). This effectively limits the A values that can be tested. The A parameter is reduced in these scenarios by reducing the mass of the perturber, so the perturber for $A = 0.01$ has a tenth the mass of the $A = 0.1$ perturber. The time evolution of the force in each case shows that the solution diverges from the analytic prediction at all times. I also see that the $A = 0.01$ and $A = 0.1$ cases follow the same trend. Decreasing the A number further, therefore, would not improve the match.

When comparing to the analytic prediction, the choice of $r_{\rm min}$ is limited by the requirement that the prediction only holds for wakes where the over-density is linear. I chose first to calculate results for $r_{\rm min} = r_{\rm s}$, where our choice of A means the wake should remain in the linear regime for all parts included in the integration ($A \leq 1$). It is clear that parts of this numerical wake are not well described by the analytic prediction. I find that the force from $r_{\rm min} = 4r_{\rm s}$ matches the analytic

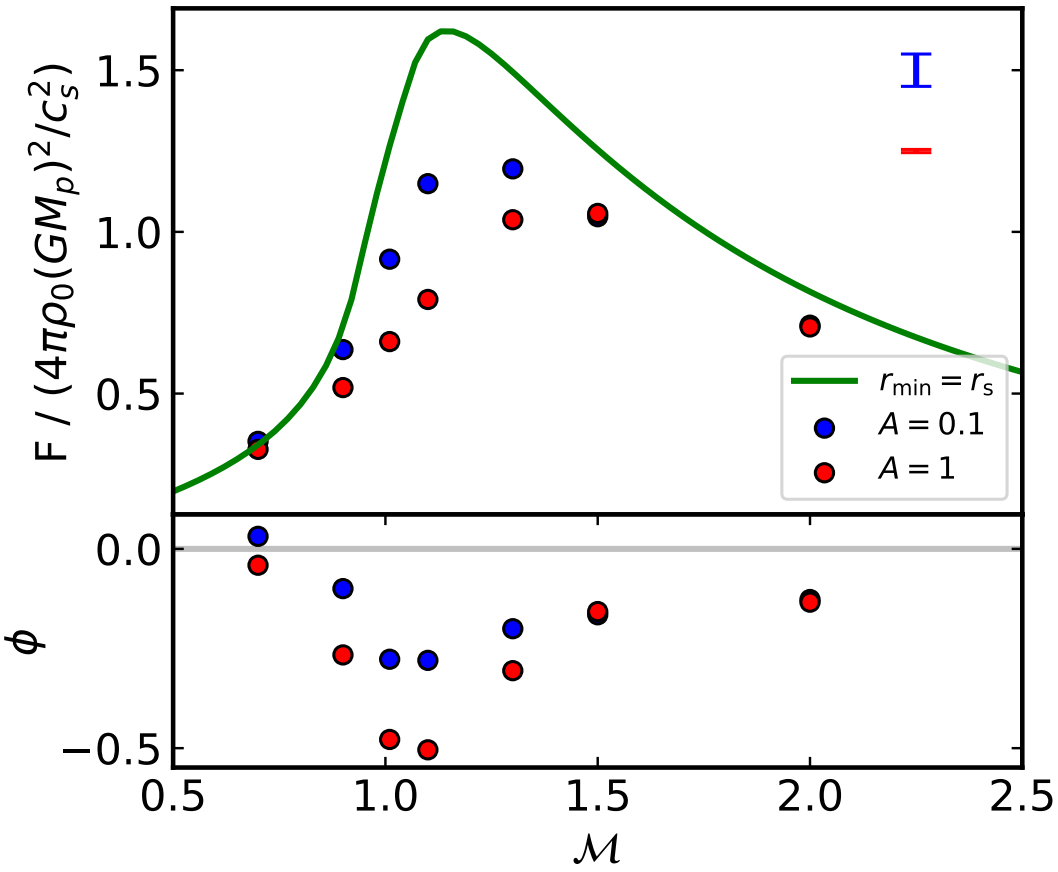

Figure 2.9 *Upper Panel: Dimensionless force from numerically induced wakes across a range of Mach numbers, at* $t = 15t_c$*. I show the numerical results (dots) and the corresponding analytic prediction (lines). The error bars show an estimate of the in intrinsic error in the force for each case. The force is well recovered in the subsonic regime, but diverges significantly for supersonic cases. The* $A = 0.1$ *setup provides a better match than the* $A = 1.0$ *case, providing systematically better matches to the predicted force. Lower Panel: Residual between the numerical and analytic results* $\phi = (F_{\rm num} - F_{\rm ana})/F_{\rm ana}$*.*

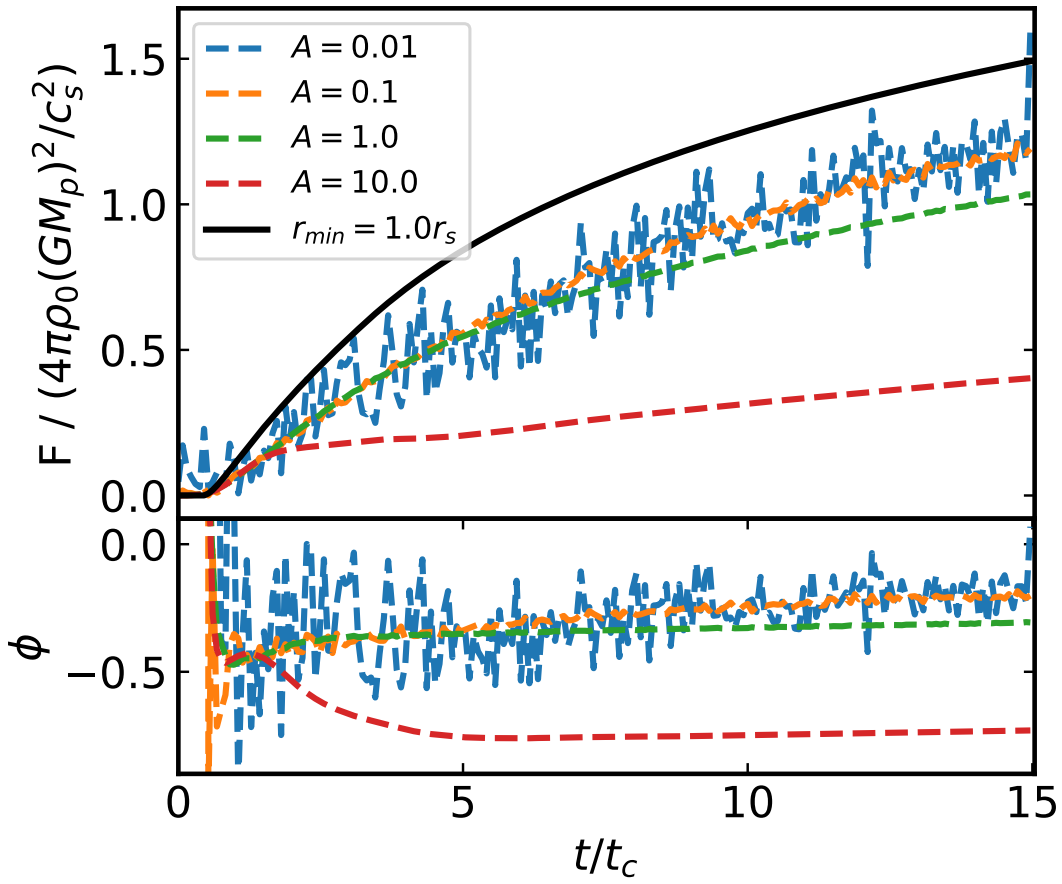

Figure 2.10 *Upper Panel: Time evolution of the dimensionless force for* $A = 0.01$, $A = 0.1$, $A = 1$ *and* $A = 10$. *The highly nonlinear case diverges early, while the transition case* ($A = 1$) *is a better match, but still differs from the strictly linear cases. Lower Panel: Residual between the numerical and analytic results* $\phi = (F_{\rm num} - F_{\rm ana})/F_{\rm ana}$.

prediction in the $A = 0.1$ case, and from $r_{\rm min} = 8r_{\rm s}$ with $A = 1$. In Figure 2.11, I show the analytic and numerical forces for $r_{\rm min} = 4r_{\rm s}$. The force in the $A = 0.1$ case shows good agreement at all Mach numbers for this lower limit, with the numerical force within 5% of the predicted value. The $A = 1$ results for this inner radius are also improved but still show disagreement of 10%. Figure 2.12 shows the equivalent forces for $r_{\rm min} = 8r_{\rm s}$, where both cases are now within 5%. It is not clear, from the force calculation alone, what is causing the mismatch in force within $4r_s$ ($A = 0.1$), or $8r_s$ ($A = 1$). There is also still some residual difference between the nuemrical and analytic results. In the next sub-section, I show that the far-field density beyond this radius matches the analytic prediction well, but the structure within this radius diverges significantly from the prediction.

2.4.2 Wake

To understand the differences between the analytic and numerical results, I directly compare the numerical density distribution to the analytic prediction for the form of the wake, given by α in Equation (2.16). The numerical over-density is found by binning the particles into (s, R) cylindrical bins, then dividing the density in that bin by the initial background density and subtracting one. Here s is the distance from the perturber along the direction of travel, and R is the cylindrical radius away from this axis. Along the s-axis, the negative s-direction is 'behind' perturber. In Figure 2.13, I show the numerical over-density in the DF wake at $t = 15t_c$. The upper part of each panel shows the numerical over-density $\alpha_{\rm num}$, with the analytic prediction $\alpha_{\rm ana}$ overlaid as white contours. The lower part shows the difference between the numerical and analytic over-densities $\phi = (\alpha_{\rm num} - \alpha_{\rm ana})$. The uniform density material ahead of the perturber has been excluded from the plot. The difference between the numerical and analytic wakes must produce the difference in forces. The broadest structure is recovered, in that there is an over-dense wake formed behind the perturber, which becomes more elongated as the Mach number increases. The lower parts of each panel show that the largest differences between the numerical and predicted wakes are found close to the perturber. This region is more clearly shown in a zoomed in view.

In Figure 2.14, I show a zoomed in view of the structure of the numerical over-density, with, as before, contours showing the analytic prediction (white dashed lines), for three Mach numbers, with both $A = 0.1$ (left column) and $A = 1$ (right column). The residual (here the difference between numerical and analytic

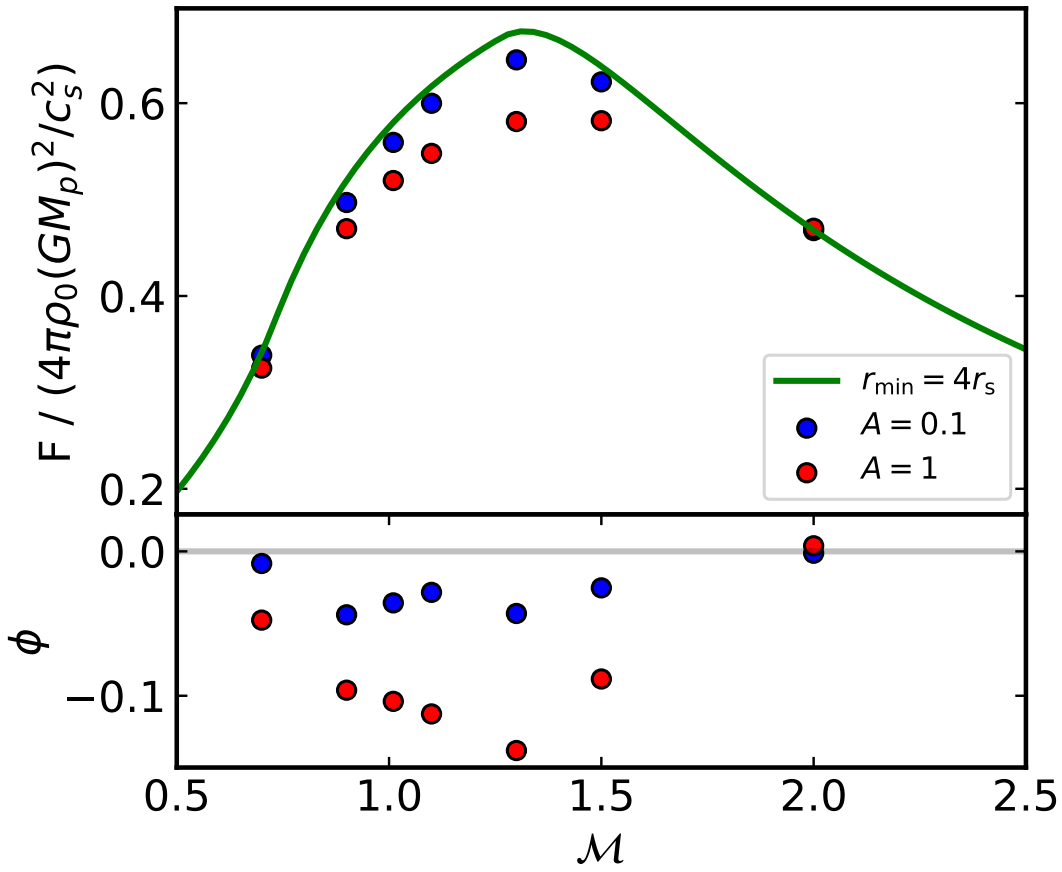

Figure 2.11 *Upper Panel: Dimensionless force from numerically induced wakes across a range of Mach numbers, at* $t = 15t_c$*. I show the numerical results (dots) and the corresponding analytic prediction (lines) for* $r_{\rm min} = 4r_{\rm s}$*. Error bars are not shown for these results, as the intrinsic errors are negligible compared to the absolute forces for this* $r_{\rm min}$*. The* $A = 0.1$ *results now show fairly good agreement at all mach numbers, with residuals at the few percent level. The* $A = 1$ *results still show significant divergence, of order 10%. Lower Panel: Residual between the numerical and analytic results* $\phi = (F_{\rm num} - F_{\rm ana})/F_{\rm ana}$*.*

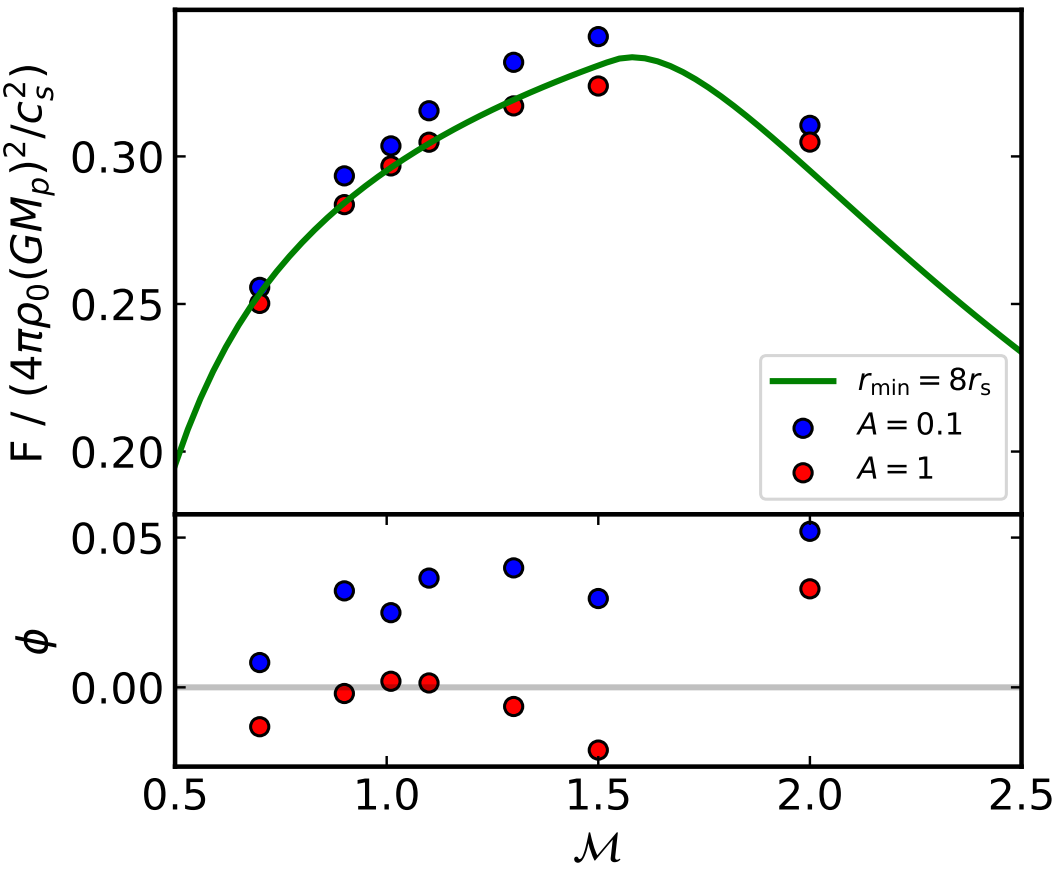

Figure 2.12 *Upper Panel: Dimensionless force from numerically induced wakes across a range of Mach numbers, at $t = 15t_c$. I show the numerical results (dots) and the corresponding analytic prediction (lines) for $r_{\rm min} = 8r_{\rm s}$. Once again the intrinsic error is negligible. Both $A = 0.1$ and $A = 1$ results show agreement within 5%, with some numerical results now above the analytic prediction. Lower Panel: Residual between the numerical and analytic results $\phi = (F_{\rm num} - F_{\rm ana})/F_{\rm ana}$.*

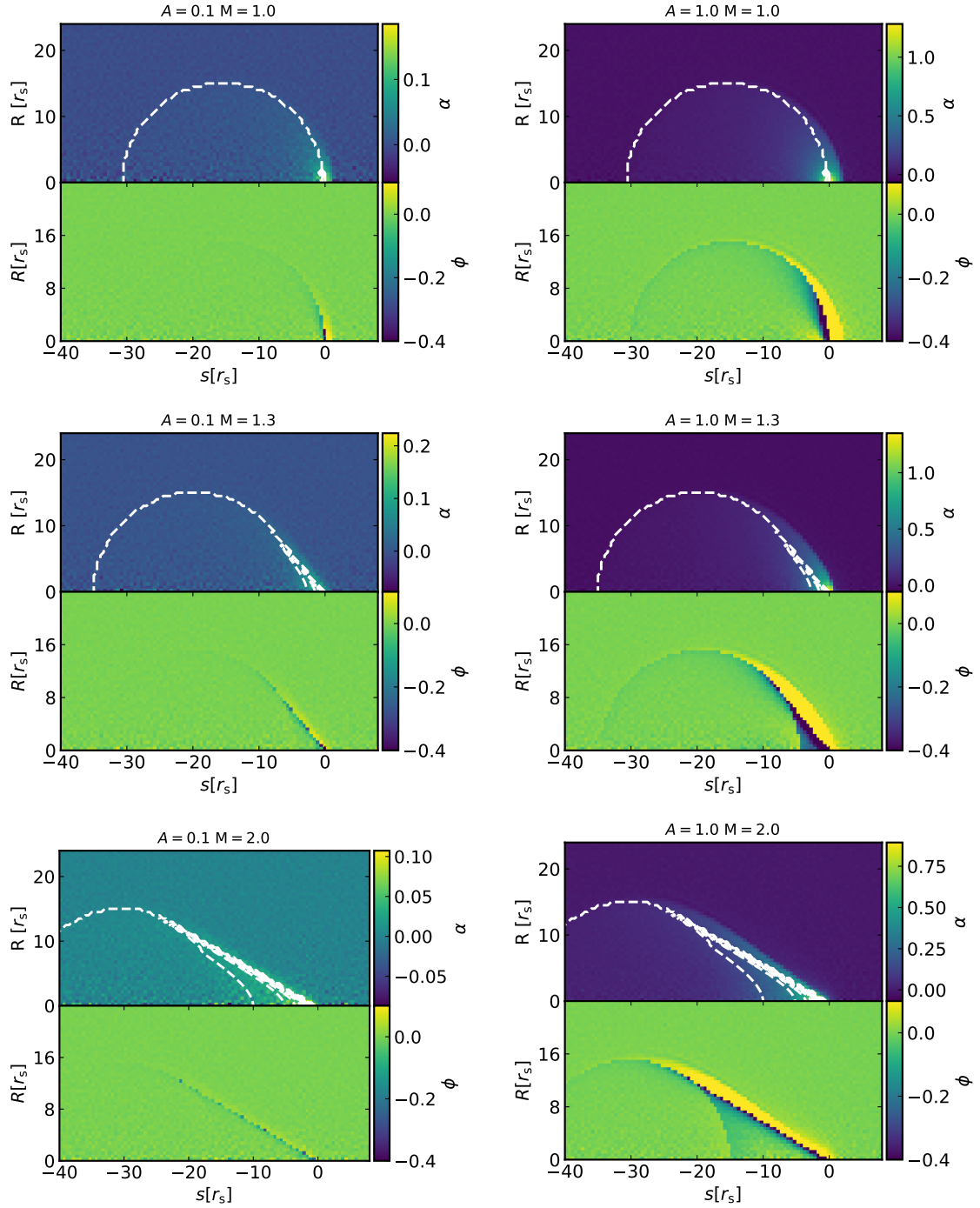

Figure 2.13 *Over-density α for $A = 0.1$ (left column), and $A = 1$ (right column), across $\mathcal{M} = 1.01, 1.3, 1.5$. The upper part of each panel showing the numerical over-density $\alpha_{\rm num}$ in the colour, and the analytic prediction for the over-density $\alpha_{\rm ana}$ as white dashed contours. The difference between these distributions $\phi = \alpha_{\rm num} - \alpha_{\rm ana}$ are shown in the lower part of each panel.*

results) is presented in the lower part of each panel, and shows us that the Mach cone structure is the main source of error. The sharpness of the density peak in the cone has been softened, with the density smeared out into a wider profile. The peak in the cone over-density is too small, and extends too far forward. This is reminiscent of the bow shock structure shown to form for non-linear cases (KK09), but now in the linear regime.

The bow shock is most pronounced for $\mathcal{M} = 1.01$, for both $A = 0.1$ and $A = 1$, and gradually diminishes as Mach number increases. The smearing out of the density profile across the cone front is present at all Mach numbers with $A = 1$, although it does seem to reduce in width. The $A = 0.1$ profile has the smearing out all but vanishing at $\mathcal{M} = 2$. The structure is generally well matched at large radii from the perturber, where the over-density is very small, but the cone is poorly recovered out to $s = 8r_{\rm s}$. For all cases, the less linear setup shows a higher value in its over-density, as expected for a higher mass perturber. The bow shock also extends further forward of the predicted structure for the less linear case.

In the $A = 1$, $\mathcal{M} = 1.3$ case (middle right), we see that the over-density in the part of the wake within the cone is lower than the predicted value, while the over-density on the spherical part of the wake is well recovered. No equivalent pattern is seen for $A = 0.1$. This shows again that it is not just the cone front that is not replicated, but also the profile behind the edge of the cone. At $A = 1$, $\mathcal{M} = 2$, the profile is evenly spread either side of the predicted front, while the over-density further within the cone is well recovered.

These observations fit with the net forces that I calculate for the different Mach numbers. The largest divergence in force is close to $\mathcal{M} = 1$, where I see the largest over-density ahead of the perturber and the poor recovery if the shock front itself, while at large Mach the force is a better match, with the small difference likely explained by the smearing out of the cone profile. In the intermediate case, we also see significant divergence, once again explained by both the bow shock and the smearing of the cone profile. The spherical parts of the wakes are in general well recovered, which fits with the subsonic cases giving a better match to the force.

We can break down the contribution to the force into spherical shells to better understand how the different parts of the structure are contributing to the force. This is simply done by performing the direct force summation for particles within a given radius range of the perturber. The results for the $A = 0.1$ scenario

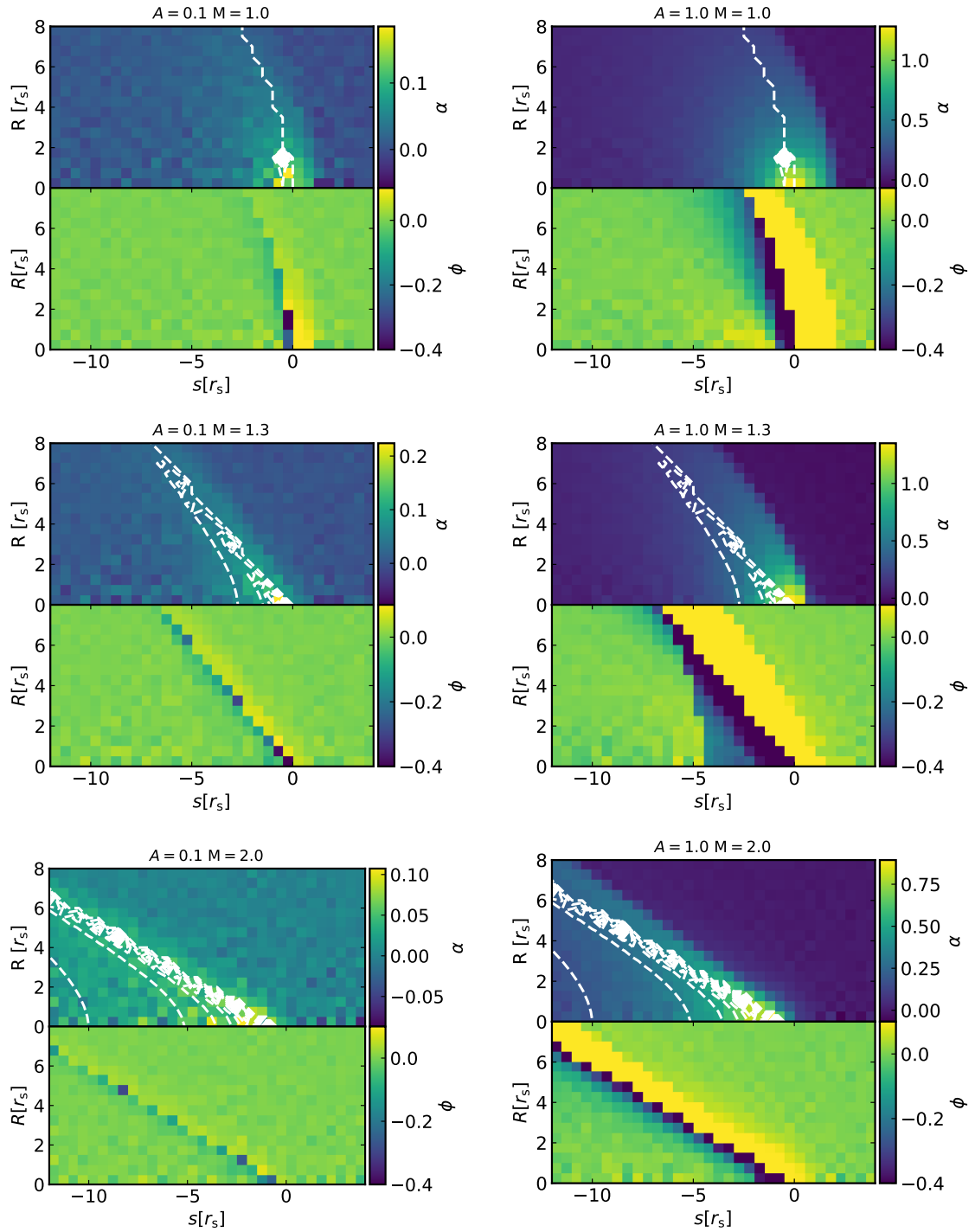

Figure 2.14 *Zoomed view of over-density for $A = 0.1$ (left column), and $A = 1$ (right column), across $\mathcal{M} = 1.01, 1.3, 1.5$. The upper part of each panel showing the numerical over-density $\alpha_{\rm num}$, with the analytic prediction for the over-density $\alpha_{\rm ana}$ as contours. We see the development of a bow shock like structure ahead of the perturber. This structure extends further forward in the $A = 1$ case, and is denser. In the $A = 0.1$ case, the effect is much smaller, though still present. The structure shrinks at higher mach numbers, with the biggest divergence close to $\mathcal{M} = 1$ in both cases. The difference between these distributions $\phi = \alpha_{\rm num} - \alpha_{\rm ana}$ are shown in the lower part of each panel.*

are shown in Figure 2.15 (top plot). This shows the numerical and analytic contributions to the force, as a fraction of the analytic force from the whole wake. We see that the force matches well at radii beyond $r_{=}4r_s$, but within this distance the force is poorly recovered at all times, with the majority of the difference coming from $r = r_s$ to $r = 2r_s$. This fits with what we see in the numerical wake, with the bow shock structure and shallower cone profile at the tip of the cone producing the mismatch, while as we move out in spherical shells, the contribution from the softened cone profile continues to have an effect, but this diminishes as more of the force comes from the well resolves interior of the cone.

The corresponding result for the $A = 1$ case (Figure 2.15, second plot) shows a similar pattern, although in this case the force is only a good match beyond $r = 8r_s$. The contribution to the difference increases as I move inward from this radius. The fraction missing from each radial bin is larger than its $A = 0.1$ counterpart, showing us that the wake diverges more across the whole range, instead of in a single region. Once again this fits with the numerical wake. The more extended bow structure has an effect at greater radii, since it extends further forward, and the interior of the cone is less well recovered, making the mismatch even larger. The far-field wake, at larger radii than before, is still well recovered, and I see this in the matching of the force for these larger radii.

In both cases, the force shows strong disagreement at very early times, but converges to a quasi-steady value by $t = 10t_c$. At the very early times, the whole force comes from the inner most bin, since the wake has not extended very far. As time increases, the total force mismatch will diminish, as the inner regions, that make up most of the difference, contribute less and less of the total force. A higher fraction of the force comes from the far-field wake at larger times, which is well matched, even at these early times. The mismatch in the inner regions does not improve, however, even if the total force match does get better.

The results presented here show that we have a deficit in the expected force, and that this comes from the innermost radii, close to the perturber. The deficit is also present from the very start, and does not converge on the predicted solution within large times. If this deficit were present in a wider physical context, the reduced force would act for a significant amount of time, and would not be disrupted by larger scale differences to the analytic setup. By this, we mean that even if the medium is not isotropic, the force from an induced wake from the local density will still be reduced, since the error comes from these inner parts which will not be changed by large scale variations in the background, or other effects that disrupt

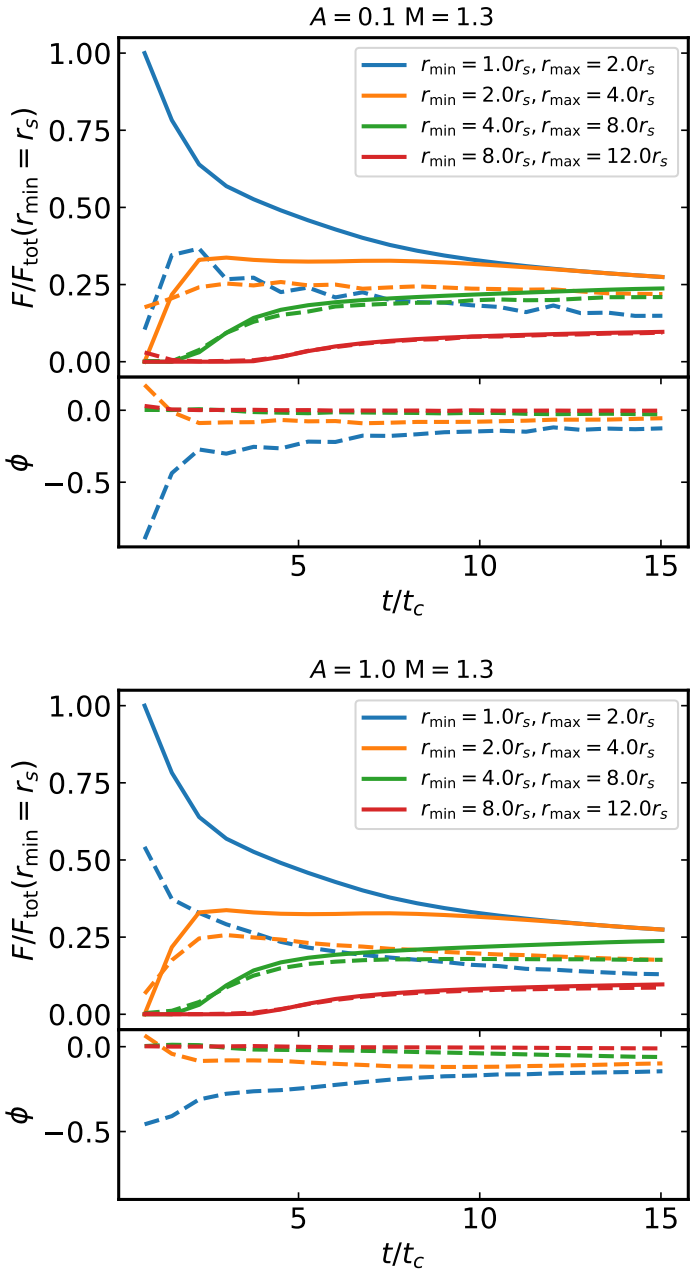

Figure 2.15 ***Top**: Upper Panel: Time evolution of the analytic (lines) and numerical (dashed lines) drag force contribution from radial bins for $A = 0.1$. The force is shown as a fraction of the total analytic force from the whole wake. The numerical and analytic forces match well outside $r = 4r_s$. The force deficit comes from inside this radius for this case. Lower Panel: Residual between numerical and analytic forces in that radial bin, as a fraction of the force from the whole wake at that time $\phi = (F_{num} - F_{ana})/F_{ana,tot}$. The residual has converged to a steady solution by $t = 15t_c$, showing the total force deficit of roughly 25% will remain for large times. **Bottom**: Upper Panel: Time evolution of drag force contribution from radial bins for $A = 1$. The numerical and analytic forces match well outside $r = 8r_s$, so a larger region is producing the force deficit, when compared to the $A = 0.1$ case.*

the extended parts of the wake.

2.4.3 Long Term Evolution

So far I have shown results for only the first $15t_c$ of the evolution of the gravitationally induced wake. Some of the previous works (Sánchez-Salcedo & Brandenburg, 1999; Kim & Kim, 2009), that have shown the analytic force can be reproduced in the linear regime, do so only for many hundreds of crossing times. At these times the wake has reached scales hundreds of times the gravitational softening scale of the perturber. The results from the larger box are shown in Figure 2.16. The intrinsic variation in the force is larger for these larger boxes, where $M_{\rm part}/M_{\rm p} = 8 \times 10^{-7}$ for $A = 0.1$ and $M_{\rm part}/M_{\rm p} = 8 \times 10^{-8}$ for $A = 1$. The increased variation makes drawing conclusions about this later time more difficult. The $A = 0.1$ case is close to being in approximate agreement, within its intrinsic variation, but still remains systematically offset below the analytic solution. There is still a deficit in the numerical force at these late times, when the wake is much larger, amounting to approximately $5\% - 10\%$ missing in the $A = 0.1$ case, and approximately 25% with $A = 1$. We do not see a significant change in the pattern observed for the results at $t = 15t_c$. While there is some suggestion that the difference in the $A = 0.1$ case may be reducing, it has not converged on the predicted force even at this late time.

2.4.4 Regions of the Wake

The results presented so far show the force calculated from either the whole wake, or radially selected parts of it. For the numerical results, this is done by calculating the force from all particles within the relevant radial limits. In an unperturbed medium, the particles ahead of the perturber cancel out the force from those behind. Once a wake starts to form, the force from the over-density creates an in-balance that produces the net drag force, anti-parallel to the direction of travel. To better understand the effect of the differences in the wake on the net force, I break the wake down into different regions. To cancel out the force from the unperturbed medium in these regions, I simply subtract out the force from particles in these regions in the initial conditions. I compare the evolution of the force from these regions to one another, to further identify what is causing the mismatch in the force.

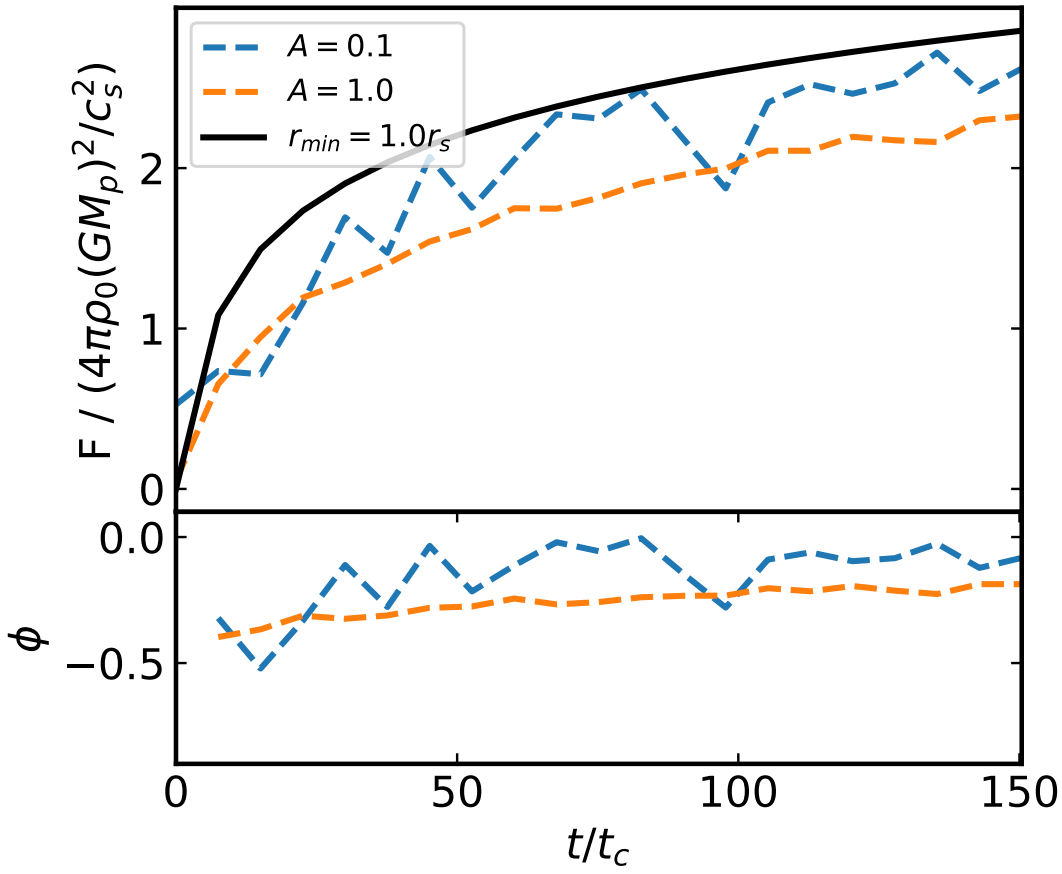

Figure 2.16 *Upper Panel: Time evolution of the dimensionless force for* $A = 0.1$ *and* $A = 1$ *in the larger box. The* $A = 0.1$ *case does not converge to the analytic solution in this longer time. Lower Panel: Residual between analytic and numerical results* $\phi = (F_{\rm num} - F_{\rm ana})/F_{\rm ana}$.

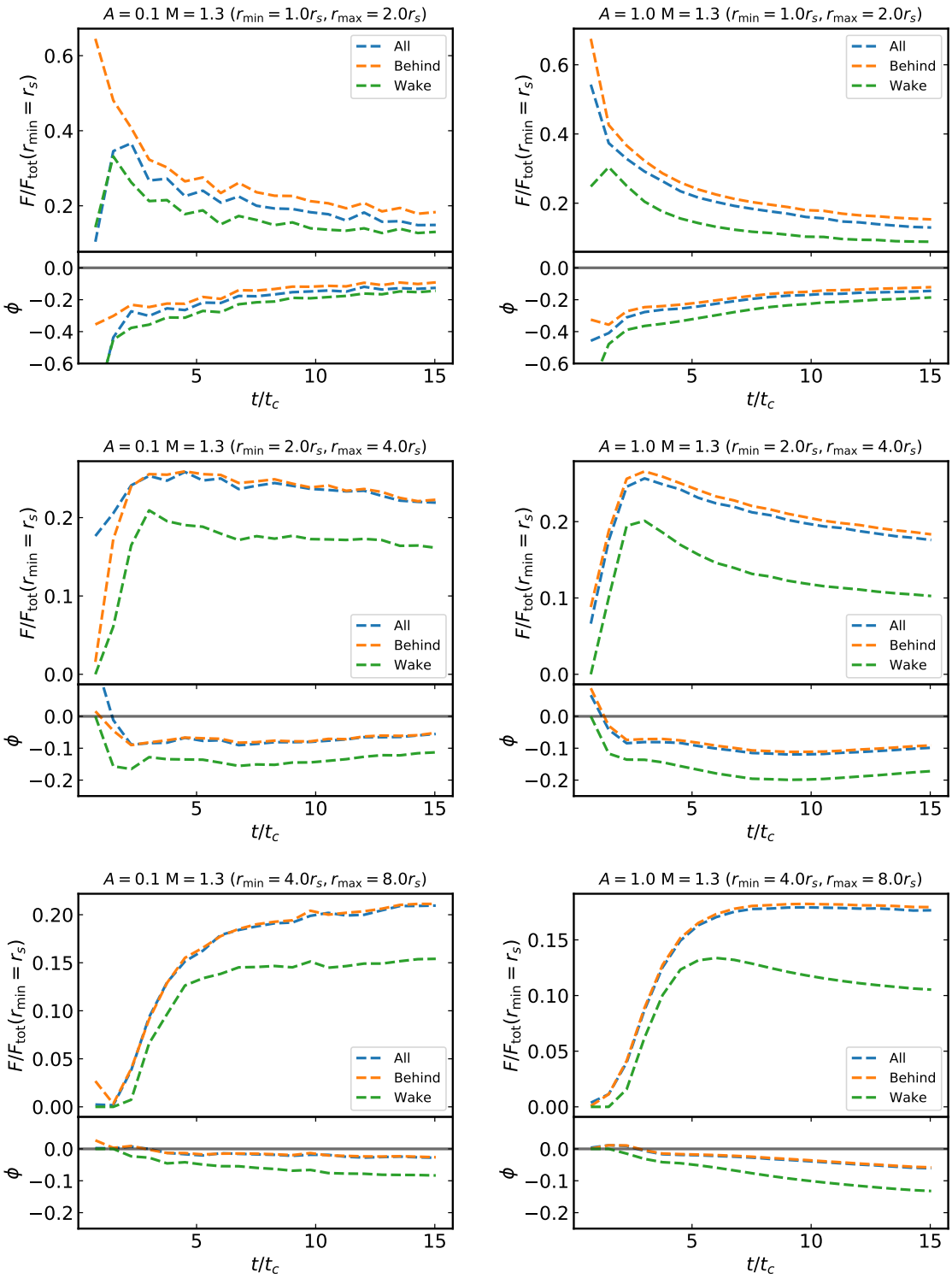

Figure 2.17 *Ratio of numerical and analytic forces from radial bins $r_{\rm min}$ to $r_{\rm max}$, calculated using all particles (blue), only particles behind the perturber (orange), and only particles in the region predicted by the analytic wake (green). The results are shown for three radial bins, with the evolution of both the $A = 0.1$ (left column) and $A = 1$ cases (right column). The lower part of each panel shows the residual between each numerical force and the analytic prediction $\phi = (F_{\rm num} - F_{\rm ana})/F_{\rm ana}$.*

In Figure 2.17, we show the force from all particles (blue), only particles behind the perturber (orange), and only particles within the zone predicted to be over-dense by the analytic solution. The left hand column shows results from the $A = 0.1$ cases, with $A = 1$ on the right. The top row show the force from $1r_s$ to $2r_s$, the middle row from $2r_s$ to $4r_s$, and the bottom row from $4r_s$ to $8r_s$. The lower part of each panel shows the residual between the appropriate force and the corresponding analytic prediction. We would expect that using only particles behind the perturber will give the largest force, since there will be no reduction in force from the over-densities ahead of the perturber, and all other particles will be included. The inner most radii, for both $A = 0.1$ and $A = 1$, show the expected pattern. The residual shows that while some of the missing force comes from the over-density ahead of the perturber, there is still force missing when only considering the region behind it.

Interestingly, the difference caused by particles ahead of the perturber is slightly larger in the $A = 0.1$ case, for this inner radial bin, which seems to be at odds with the smaller size of the forward over-density in this case. The difference is very small, and is explained by this comparison being performed with the force normalised to the total from the whole wake. The absolute forward force is larger in the $A = 1$ case, but is smaller as a fraction of the total force. As radial bin moves outwards, the force from behind the perturber matches that from the whole box. The mismatch continues to larger distances for the $A = 1$ run, which fits with the greater forward extent of the 'bow wave' in that case. By $r_{\rm min} = 4r_s$ to $r_{\rm min} = 8r_s$, the forces from these two regions match well in both cases. The force only from particles in the region predicted by the analytic solution is significantly lower than the force from both all particles, and all particles behind the perturber. This is not unexpected, given this region excludes a number of particles. The difference is present in both cases, and in all radial bins. It effectively shows, together with the 'behind' force, how much is contributed from the region behind the perturber, but ahead of the predicted cone edge. For the larger radial bins, the residual starts off close to zero, before dropping to a steady value. The initial match is simply because the force from these regions at early times should be zero, since the wake has not reached that position.

If the wake had the predicted structure, the force from the wake region would give the best match. In the inner regions, the difference in the force is created by the conditions described above. However, it should be noted that the structure of the wake in the outer regions is still not correctly recovered, since the force

from just the analytic wake region under-produces the expected force. There is enough mass in the whole wake at this distance, but the structure is not correct. These results conclusionsively show that the force deficit that we observe for both $A = 0.1$ and $A = 1$ is caused by a combination of the 'bow wave' structure and the smearing out of the density profile about the cone. Neither one alone is capable of explaining the reduced force that is observed, and while the force is recovered, beyond a certain radius, the structure still does not match.

2.4.5 Solver Comparison

It is clear from the above results that the gravitationally induced wake, produced by the MFM solver, does not match the analytic prediction. To investigate the cause of the mismatch, we can compare the force from wakes produced using both the MFM and PSPH solvers. In Figure 2.18 I show the results from runs with $A = 0.1$ (MFM in blue and PSPH in orange) and $A = 1$ (MFM in green and PSPH in red). Results between solvers are essentially identical for the same setups, with the PSPH results lying very slightly below the MFM results in both cases. While both methods sample the underlying density and temperature in a similar manner, they solve the equations of hydrodynamics in different ways. Whatever is causing the differences between these results and the analytic solution is present in both methods.

2.4.6 Varying Conditions

It is possible that the missing force result, that we have found above, is somehow caused by the specific physical and numerical conditions that we have used, despite the scale free nature of the problem. In order to check if this is the case, we have run a number of setups with different background temperatures and perturber masses, adjusted to keep the same $A = GM_p/c_s^2 r_s$ numbers. The variation in temperature manifests itself in the initial internal energy of the uniform gas, and so in the sound speed of this gas. Sound speed is proportional to the square root of the temperature, and so reducing the temperature by an order of magnitude changes the sound speed by $1/\sqrt{10}$. In Figure 2.19, I show the results from these runs, where the temperature has been reduced by an order of magnitude, and the sound speed is now $c_s = c_0/\sqrt{10}$. These are shown in comparison to the standard runs, where $c_s = c_0$. The perturber masses have

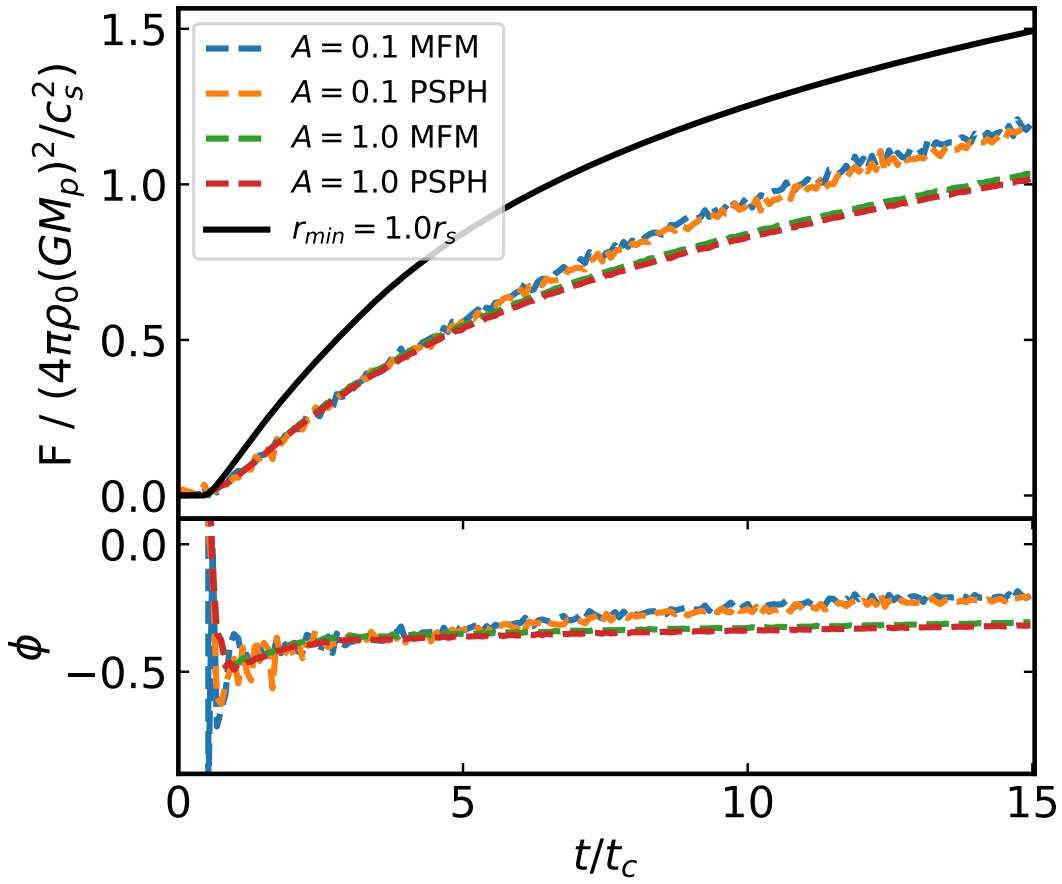

Figure 2.18 *Upper panel: Comparison of the time evolution of the dimensionless force from wakes produced with different hydrodynamics solvers, MFM and PSPH, for $A = 0.1$ and $A = 1$. Lower panel: Residual between numerical force and the analytic prediction $\phi = (F_{\rm num} - F_{\rm ana})/F_{\rm ana}$.*

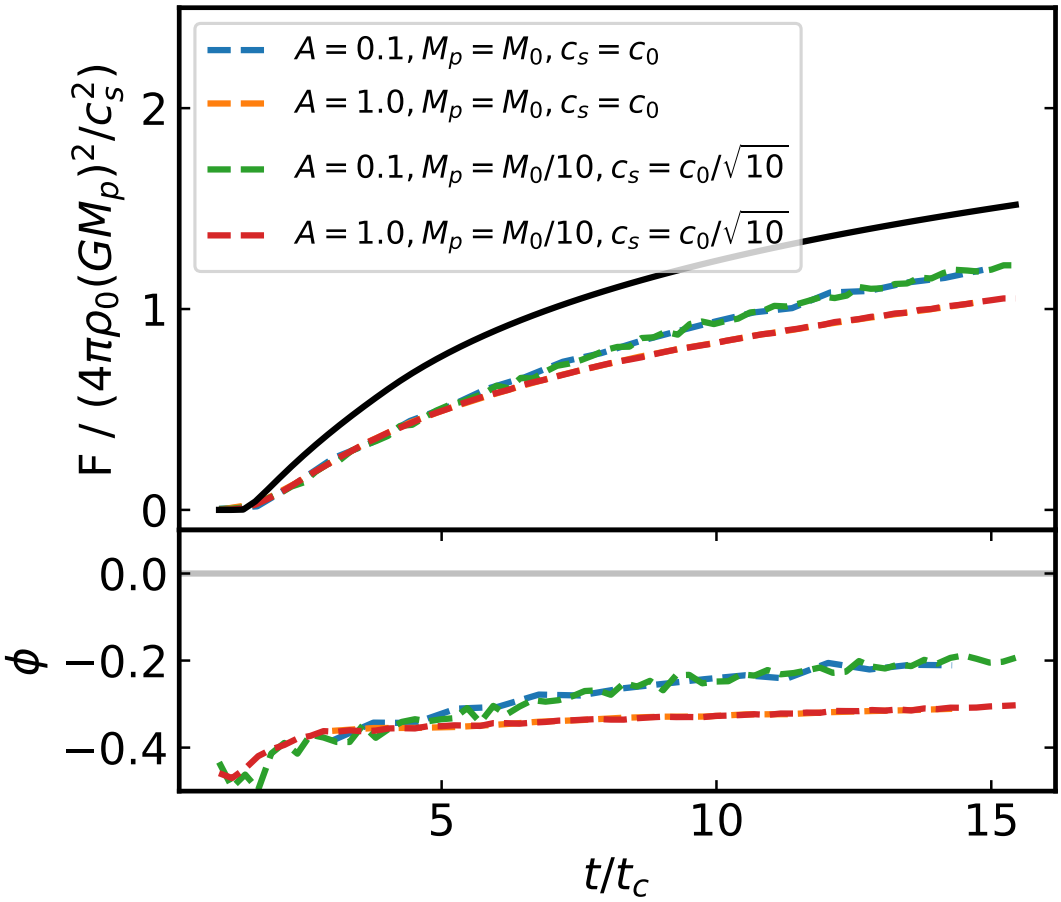

Figure 2.19 *Upper panel: Comparison of the time evolution of the dimensionless force from wakes produced with different sound speeds, for $A = 0.1$ and $A = 1$. Perturber masses are adjusted appropriately to keep the A value the same between cases with different sound speeds. Lower panel: Residual between numerical force and the analytic prediction $\phi = (F_{\rm num} - F_{\rm ana})/F_{\rm ana}$.*

been adjusted to maintain the same A value between the compared cases. I see that the evolution of the force is identical for the same A value, despite the order of magnitude change in the temperature/internal energy of the background gas. This is strong evidence that the specific conditions are not responsible for the difference between the analytic and numerical results, and that the scale free paradigm is a fair way to test the problem.

The other condition that I vary is the softening scale of perturbing potential r_s. Once again I adjust the perturber mass to keep the A parameter the same. By increasing the softening scale, but keeping the sound speed the same, I effectively reduce the number of crossing times that the simulation can be run for, before the wake wraps around through the periodic boundary. To compromise between

varying the softening scale by a significant amount, and keeping a reasonable number of crossing times, I have only doubled the softening scale, from r_s = 0.125kpc to 0.25kpc. The results are shown in Figure 2.20, where it is clear that the evolution is the same in both cases. The force is calculated for equivalent parts of the different wakes, using $r_{\rm min} = r_s$ for both. When the evolution is scaled by the different crossing times $t_c = r_s/c_s$, to best compare the evolution in the scale free parameter space, the analytic force is also the same in both cases. This makes sense since the dimensionless force shown here is independent of perturber mass, and depends only on the A parameter, the Mach number, and the proportion of the wake that is being integrated. When re-scaled by crossing time, which is obviously dependent on softening scale, it is clear how the two cases will produce the same dimensionless force. At any given number of crossing times, the wake will have extended the same number of softening scales from the perturber, and so will produce the same dimensionless force.

Varying either temperature/internal energy/sound speed, or softening scale, has no tangible effect on the force deficit between the numerical and analytic results. The above results are in complete agreement with our previous findings, showing the difference is not a fluke of the specific conditions or setup, but instead is a significant feature of the numerical methods.

2.5 Discussion

I have shown that the MFM Lagrangian hydro solver produces an over-dense wake from a massive perturber that does not match the wake predicted by linear perturbation theory. The numerical wake produces a force that is between 10% and 25% below that predicted by the analytic treatment. The difference is present across $\mathcal{M} \sim 1-2$ to at least $15t_c$, and tested to $150t_c$ for $\mathcal{M} = 1.3$. The largest difference is found close to $\mathcal{M} = 1$, with cases at $\mathcal{M} = 0.7$ and $\mathcal{M} = 2$ showing good agreement with the predicted force. The deficit is present well within the linear regime for which the linear prediction has been shown to hold well. We see that the wake in the innermost regions does not match the predicted over-dense structure. A bow-shock like structure builds up in front of, and to either side of, the perturber, while the sharp profile in the over-density of the Mach cone is softened. The front is smeared out and the peak in the over-density is lowered. This smearing extends along the cone front, but the largest impact on the force comes from differences in the innermost $4r_s$ (for $A = 0.1$) or $8r_s$ (for $A = 1$). The

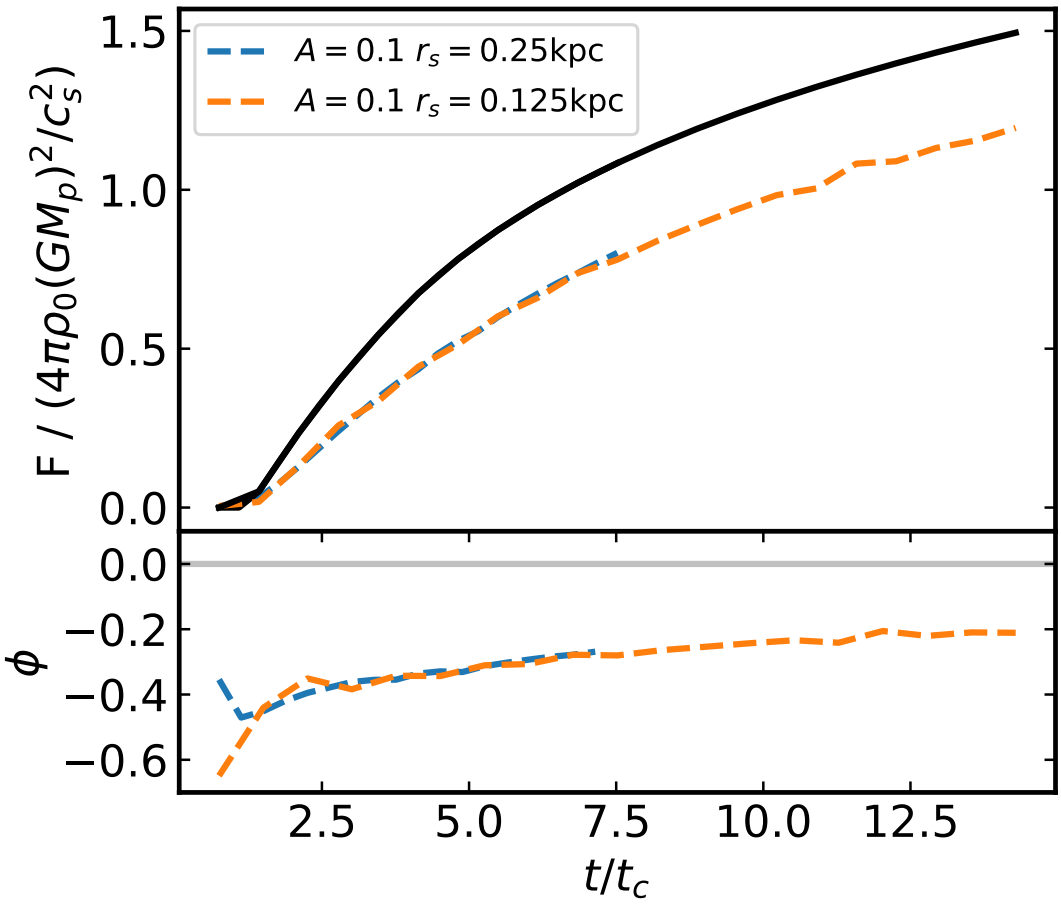

Figure 2.20 *Upper panel: Comparison of the time evolution of the dimensionless force from wakes produced with different softening scales r_s, for $A = 0.1$. Perturber masses are adjusted appropriately to keep the A value the same between the different cases. The force is calculated for $r_{\rm min} = r_s$, and are compared in terms of crossing times, which are different physical times. Lower panel: Residual between numerical force and the analytic prediction $\phi = (F_{\rm num} - F_{\rm ana})/F_{\rm ana}$.*

evolution of the force in the innermost regions shows extreme divergence early on, when the wake barely extend beyond the softening scale, but soon converges to a quasi-steady form. Spherical shells further from the perturber show a different trend, where the force matches at all times, but the structure of the wake at these large radii is still incorrect. The evolution of the force is identical for tests with the alternative PSPH solver.

There are a number of possible numerical causes for the mismatch between these results and the linear predictions. Since both the Lagrangian solvers used here are run using the same gravity backbone, it is possible the gravitational force felt by the gas particles is not replicating the assumptions made in the derivation of the liner solution. I have added the Plummer potential as a fixed external potential, and disabled the self gravity of the gas. The Plummer potential is used to soften the gravitational force felt by particles that pass very close to the origin of the potential, as the potential for a point mass diverges. This can create extreme accelerations, where particles are flung off at great speeds, when particles move unphysically close to the perturber. The analytic prediction assumes that the perturber is a point mass, but Ostriker states that the solution for an extended perturber (whose extent is $r_{\rm min}$) should be identical to their solution beyond $r_{\rm min}$. Previous numerical studies have also used Plummer potentials, and have recovered the analytic prediction. It is possible that the use of this gravitational softening means that particles that pass close to the perturber are feeling the incorrect force. Their trajectories, even once they have moved beyond the softening length, will not be completely physically correct. This does not explain the bow structure, however, which extend more than r_s ahead of the perturber, nor the lower density peak in the cone profile at a few r_s from the perturber, where the vast majority of the particles will not have passed close to the perturber.

The problem may lie with recovering the hydrodynamics itself, rather than the gravitational force. Both solvers used here are Lagrangian in nature, while the major numerical studies I refer to are performed using Eulerian grid based methods. Particle based methods have traditionally struggled to capture sharp changes in density, which I see in the density profile of the cone front. It is possible that the smearing of this profile leads to the development of the density ahead of the perturber, as the over-density spreads ahead of its predicted position. The combination of the smeared out profile and bow structure produce the difference in the net force on the perturber.

2.5.1 Implications for Cosmological Simulations

I have shown that there is a significant DF force deficit, across many crossing times, when idealised dynamical friction tests are run using Lagrangian hydrodynamics methods. This force deficit is caused by the over-dense wake being improperly formed. These Lagrangian solvers do not reproduce the sharp edge of Mach cone predicted by the analytic work, and these conditions continue for at least tens to hundreds of crossing times. Here I discuss the implications this has for DM substructure in cosmological simulations.

The typical conditions in which we find sub-halos in cosmological simulations can be mapped onto this idealised setup by estimating the appropriate Mach number and A parameter. I use halos and sub-halos from the IllustrisTNG-300 simulation box (Nelson et al., 2018) to assess the conditions in which simulated sub-halos are found. We select all host halos in the mass range $10^{11} M_\odot$ to $10^{15} M_\odot$. For each host in this range, we then assign a sound speed. This sound speed is calculated from the virial temperature of that host halo

$$c_s = \sqrt{\frac{\gamma k_B T_{\rm vir}}{\mu m_p}}, \tag{2.27}$$

where μ is the mean molecular weight of the gas, m_p is the mass of a proton. The virial temperature $T_{\rm vir}$ is found by assuming that the gas falling onto the virial radius is shock heated, and transforms its kinetic energy into thermal energy. If we assume this produces an isothermal sphere, the virial temperature is given by

$$T_{\rm vir} = \frac{1}{3} \frac{\mu m_p}{k_B} \frac{G M_{200}}{r_{200}}. \tag{2.28}$$

The virial mass is taken as the mass M_{200} enclosed within the radius r_{200}. This is the radius at which the average density of the enclosed material is two hundred times the critical density $\rho_{\rm crit}$. The sub-halos associated with these hosts are then binned by their mass and this sound speed. The most massive sub-halo in each host is excluded, as it is assumed to be the central object of that halo.

These sub-halos are taken as the perturbing massive objects, as they move through the extended gaseous medium of their host. The peculiar velocity of the sub-halo, relative to the peculiar velocity of the host halo, defines the velocity for the calculation of the sub-halo's Mach number. The radial extent of the sub-halo

is the equivalent to the softening scale of the Plummer potential. This is taken as the radius at which the density profile of the sub-halo produces the maximum circular velocity $V_{\rm max}$. The mass of the sub-halo, which is the equivalent quantity to the mass of the perturber, is simply the total baryonic and dark matter mass associated to that sub-halo by the halo identification algorithm.

The distribution of masses and sound speeds are shown in Figure 2.21. The top plot shows the conditional probability of a sound speed, given a sub-halo mass. The sound speed is only dependent on the mass of the host, which leads to the bands at the high sound speed/host mass end, where there are the fewest samples. We effectively see the negative slope in the mass function of the halo/sub-halo population, as there are many more sub-halos associated with low mass halos (low sound speeds), even though the most massive halos individually contain many more sub-halos than any given lower mass halo. The middle plot shows the mean Mach number for each pixel. The distribution is fairly uniform across both mass and sound speed. The mean Mach number is around $\mathcal{M} = 1 - 2$. This is also shown in the top panel of Figure 2.22 with the conditional probability of a Mach number, given a sub-halo mass. The bottom plot of Figure 2.21 shows the mean A parameter for each pixel. The distribution at the high sub-halo mass end is dominated by small numbers of very high A parameter values in each cell. The overall distribution of A parameters is shown in the middle panel of Figure 2.22. The distribution has been truncated at $A = 10$, but continues to values in the thousands. These extreme A numbers largely come from sub-halos with small numbers of particles, some of which have very small radii, which may be numerical artifacts of the halo finder. I show the conditional probability of finding an A number, given a sub-halo mass. The sub-halos below $10^{10} M_{\odot}$ in mass exist well within the linear regime ($A << 1$). Those between $10^{10} M_{\odot}$ and $10^{11} M_{\odot}$ show a wider range of masses, but still mostly reside in the linear or quasi linear regime ($A \leq 1$). Above this mass, the distribution spreads significantly, with a range of A numbers, a significant fraction in the non-linear regime ($A > 1$). The more massive sub-halos having larger A numbers is simply a factor of them having higher masses. The increase in size is not enough to counteract this increased mass, when it comes to their A parameter value.

The IllustrisTNG-300 sub-halos show that a large fraction of sub-halos in a typical state-of-the-art cosmological simulation exist within the linear DF regime, with $A < 1$, and with Mach numbers in the range $\mathcal{M} = 1 - 2$. This is the regime in which we have found a discrepancy between the analytic prediction, previously

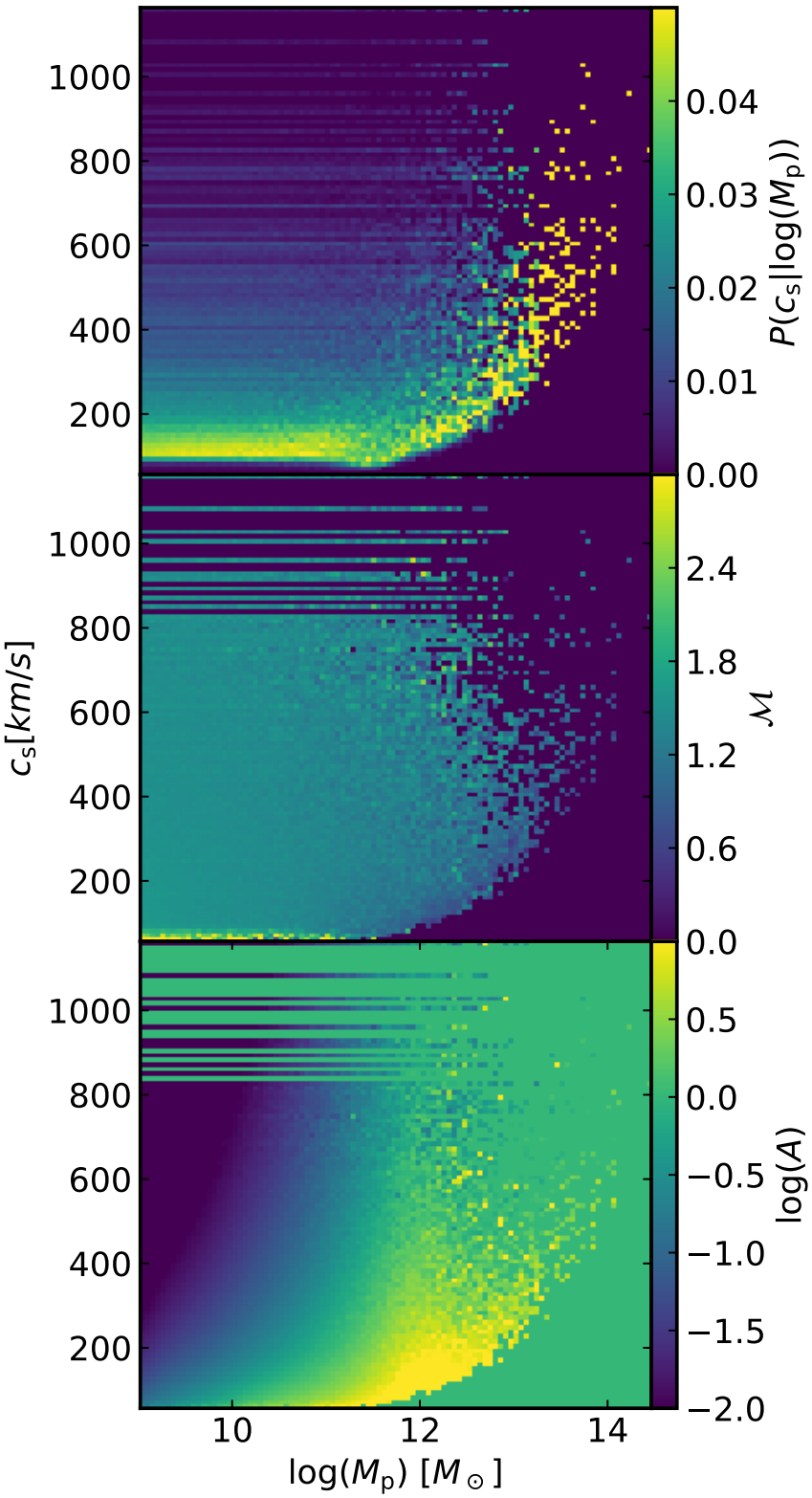

Figure 2.21 *Sub-halos from the IllustrisTNG-300 simulation box. All panels show the sub-halos with masses between* $M_\mathrm{p} = 10^{11} M_\odot$ *and* $10^{15} M_\odot$*, with sub-halo mass against sound speed. Top panel: Conditional probability of a sound speed, given a sub-halo mass. Middle panel: Average Mach number of the sub-halos in each* $(M_\mathrm{p}, c_\mathrm{s})$ *pixel. Bottom panel: Average A value.*

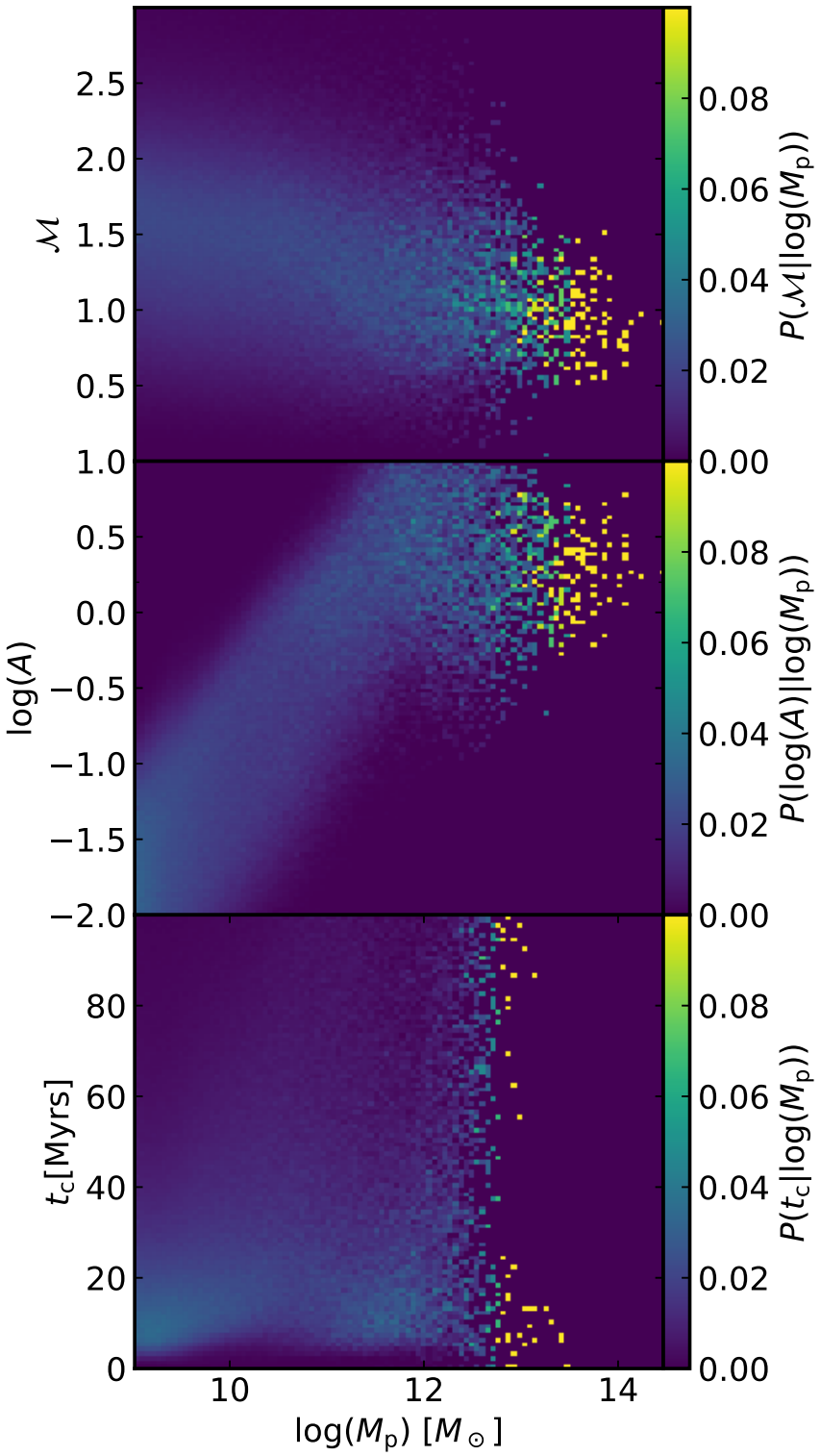

Figure 2.22 *Top panel: Conditional probability of finding a Mach number, given a sub-halo mass. Middle panel: Conditional probability of finding an A number, given a sub-halo mass. Bottom panel: Conditional probability of finding a crossing time, given a sub-halo mass.*

confirmed by high resolution Eulerian, grid based, simulations, and the numerical results from the Lagrangian solvers. The bottom panel of Figure 2.22 shows the distribution of sound speed crossing times for the radial extents of these sub-halos. We see that these values are of order $10 - 100$ Myrs. The period for which we have found significant divergence from the analytic solution is at a minimum $15t_{\rm c}$, with some results extending up to $150t_{\rm c}$. This corresponds to between hundreds of mega-years to several giga-years. Numerical results from Eulerian solvers require a time until the solution have converged to the predicted result (Kim & Kim, 2009), so this problem may exist across hydro methods. The reduced force for extended periods of time will lead to significantly slower infall of sub-halos into their host structure. The energy and angular momentum transferred from the perturbing sub-halo to the CGM will be equally reduced.

A number of factors make DF in a system with more complete physical processes diverge from the analytic solution. The gas will obviously experience self gravity, leading to the further growth of the over-dense wake. Radiative cooling mechanisms move the gas away from the adiabatic scenario considered in the analytic solution. Feedback mechanisms from star formation, such as supernova driven winds, will also have complex effects. These will all be present in the full cosmological context of cosmological simulations. The effect of these processes will be to significantly change the structure of the wake in complex ways. However, the fundamental mechanism of the problem will still be present, since any structure that physically should remain will likely under estimate the DF force in the same way as found in the simplified case. The numerical DF force will then still undershoot the physically correct DF force. I will run a set of simulations that probe the effects of self gravity and radiative cooling in the development of the wake.

The simulated medium through which the sub-halos move will be far from uniform. The CGM is highly variable, made up of gas with a huge range of sound speeds. The wake will not build up smoothly as shown in the idealised scenario, where it can grow for an unlimited time through an infinite uniform background medium. Instead, at any given time, the wake will be built from the medium that the perturber (sub-halo) is moving through at that time. In the context of results, however, this is not that significant, since the DF force deficit that I find comes from the inner most radii, and starts from the earliest time. The wake further from the perturber does not contribute to the missing force, so whether it is present, or disrupted by the more complex physical conditions of the

medium, the force deficit will still be present. Thus this result has implications far beyond the highly idealised scenario studied directly here, into the cases that include more varied and complex physical processes.

2.5.2 DF Hydro Test

It is clear that DF is a complex problem, and a significant challenge, for modern hydro solvers to treat. I have shown that state-of-art solvers can struggle to replicate the analytic solution. DF as a process is present in astrophysical systems across a host of mass and length scales, and is crucial to our understanding of the evolution of these systems. This makes DF an ideal candidate for inclusion as a standard test when developing numerical hydrodynamics methods and algorithms. It has a well defined solution for comparison, and includes both a shock with complex geometry and regions with simpler advective flows. The scale free nature of its formulation allows for straight forward comparison between numerical tests. I therefore suggest using DF as a new standard test, where the test case is set up with a linear A parameter, with Mach number in the range $1 < \mathcal{M} \leq 1.6$. The scale free nature allows any choice of density, temperature, sound speed, perturber mass, and so on, as long as it produces the desired A parameter and Mach number. This provides an effective diagnostic test of the hydro solver, that includes the effects of gravity, while maintaining a very well defined analytic solution.

2.6 Conclusion

I have run a suite of idealised dynamical friction tests, spanning a wide range of the A - Mach number parameter space. While the setup for the dynamical friction study presented here is idealised, it serves the purpose of testing the capability of modern hydro-dynamical solvers to recover long studied analytic solutions for a well defined test problem. The problem tests both hydrodynamics and gravity. These results show that the MFM and PSPH numerical wakes under-produce the net force on the perturbing object, with the difference coming from the structure of the wake close to the perturber. This difference does not decrease significantly over large times, and is present for two independent Lagrangian methods of modelling the hydrodynamics. This net reduction in the force will act for long

periods of time in cosmological simulations, with the potential for significantly underestimating the physical merger rates of dark matter substructure. It is clear that dynamical friction is a key process that must be captured accurately if we are to simulate this, and other, astrophysical processes. I suggest that the idealised dynamical friction 'wind tunnel' should be introduced as a standard test for gravo-hydrodynamics codes.

Chapter 3

The Residual Distribution Solver

3.1 Introduction

The vast majority of the observed baryonic matter in the Universe is in the form of baryonic gas. The behaviour of this gas can be modelled by solving the Euler equations for an inviscid fluid, which describe the conservation of mass, momentum and energy. These equations must be solved as a set of simultaneous partial differential equations (PDEs). For all but the simplest problems, this must be done numerically. The Navier-Stokes equations, which include the transformation between kinetic and internal energy via viscosity, can also be used. However the length scales over which this viscosity acts are much smaller than the resolution elements of galaxy formation simulations, allowing the simpler Euler equations to be sufficient. On a fundamental level, the equations must be discretised in some manner, to allow the numerical solution to be found. Typically this leads to a choice between discretising the problem in space, tracing the fluid evolution using a set of static cells, and discretising the problem by mass (Agertz et al., 2007). In the latter case, the gas is modelled as set of massive particles. Astrophysical simulations have been performed with a variety of both these approaches, broadly divided into Eulerian grid based methods (Teyssier, 2002; Bryan et al., 2014) and Lagrangian particle methods (Lucy, 1977; Gingold & Monaghan, 1977; Springel, 2005), alongside more recent hybrid moving mesh approaches that combine the Lagrangian nature of the particle methods with the various advantages of the Eulerian grids (Springel, 2010; Hopkins, 2015; Duffell, 2016). The implementation that I will discuss here is for an Eulerian method built

around an unstructured mesh, but the underlying method is naturally suited for future adaptation into such a moving mesh scheme. Solving for the evolution of this baryonic gas has been crucial in the development of our understanding of many astrophysical scenarios, from the creation of reionisation and the first galaxies (Feng et al., 2015; Ma et al., 2018), to the evolution of star forming regions (Clark et al., 2005), proto-planetary disks (Kuffmeier et al., 2017), and so on.

Some of the most successful astrophysical simulation codes solve the evolution of the baryonic gas using Eulerian grids, mentioned above and in Section 1.2.3. The majority of these divide the computational domain into a large number of identical cube shaped cells forming a structured mesh (see Section 3.3.1). Modern codes often take advantage of mesh refinement algorithms. These allow for some cells within the mesh to be subdivided into multiple smaller cells, usually dividing each edge length in half (Bryan et al., 2014). This can be used to refine the mesh in regions of increased density, which are usually of the most interest. This process is known as adaptive mesh refinement (AMR) (Berger & Oliger, 1984, Bryan et al., 2014). The fluid state is traced by these cells, with the evolution found by calculating the flux of gas between finite volume cells. This flux is found by solving the Riemann problem at the cell face (Fryxell et al., 2000; Stone et al., 2008), again described in more detail in Section 1.2.3. These approaches innately break the problem down into a set of one dimensional problems across cell faces, which inevitably ignores the information of flows in orthogonal dimensions. This is sometimes referred to as *dimensional splitting*. Flows can only travel across faces, which in structured meshes can lead to preferential flow directions. These can produce numerical artifacts, such as carbuncles, and can suppress flows in other directions. An example of the carbuncle effect can be seen with the Noh problem (see Section 3.5.2), produced using a Roe solver (Paardekooper, 2017). This is shown in Figure 3.1. The Roe solver, shown in the top row of results, shows spurious flows along the cardinal directions, where the radial flows are aligned exactly with the faces of the Cartesian grid. This results in aberrations in evolution of blast wave, which should be rotationally symmetric. The effect becomes more extreme as the resolution increases. The RD solver results, produced using a unstructured mesh of triangles, does not show these aberrations.

Since material cannot flow across the corner of cube cell, some methods have implemented corrections to attempt to account for these flows (LeVeque, 2002).

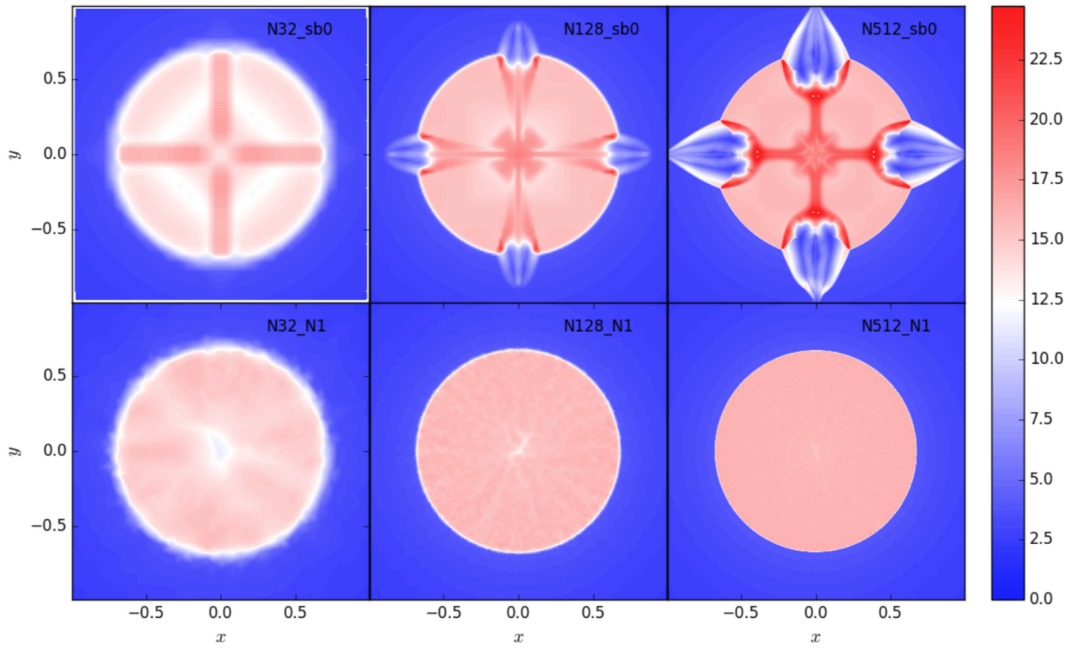

Figure 3.1 *Results for the Noh problem test, from the Roe solver (top row), and the RD solver (Bottom row), taken from Figure 21 of Paardekooper, 2017. The Roe solver results show the carbuncle numerical artifacts in directions where the flow is precisely aligned with the flow. The effect gets more extreme as the resolution increases. The equivalent RD results do not show this effect.*

Over the years, an alternative strategy has emerged, one in which dimensional splitting is not required. These solvers aim to produce truly multi-dimensional hydro-solvers that calculate the evolution of fluids across some mesh of volume elements by calculating the flow across elements in all dimensions at once (Abgrall & Roe, 2003; Abgrall, 2006; Deconinck & Ricchiuto, 2007).

3.2 Residual Distribution Theory

In this section, I will lay out the background and derivation of the residual distribution (RD) partial differential equation solver (Abgrall, 2006; Deconinck & Ricchiuto, 2007; Ricchiuto & Abgrall, 2010). I will cover the precursor to the method, the Roe solver (Roe, 1981; Stone et al., 2008), the important notation definitions, and the fundamental assumptions made in producing its form. The term residual distribution covers a whole family of solver, all built around the same idea (Ricchiuto & Abgrall, 2010). The various different approaches that have been developed within this method will also be discussed, including the choices I have made for the implementation that I am presenting here.

3.2.1 Roe Solver

In order to understand the derivation of the residual distribution method, it is useful to understand the work that came before their development. This can be done by considering the Roe Riemann solver (Roe, 1981; Stone et al., 2008; Paardekooper, 2017), which is formulated in a way that is directly analogous to the residual distribution approach, effectively making it a one dimensional predecessor of the residual distribution family. Riemann solvers produce solutions to the Riemann problem for sets of conservative equations. The problem consists of initial conditions with one discontinuity, and uniform states either side of this boundary. Roe laid out an approach by which a non-linear system of partial differential equations can be reformulated in a linear form, and how this can be used to solve the system. The method relies on the fact that the solution to any linear system of hyperbolic partial differential equations (PDEs), such as those given in Equations 3.1 and 3.2, can be written as the sum of waves (see Section 1.2.3). Between these waves there are intermediate states, with *wave strengths* that are the size of the discontinuity about each wave front. These waves are

the eigenvectors of the Jacobian matrix $\mathbf{A}$ in the set of PDEs, with the wave speeds given by the eigenvalues of this same matrix. Figure 3.2 shows a graphical representation of these fundamental waves. The initial states, either side of the boundary, are $\mathbf{Q}_{i-1}$ and $\mathbf{Q}_i$, with the boundary itself shown by the dashed line, and time is increasing in the y-direction. The position of the wave-fronts, from these states, are shown by the outermost solid black lines. They propagate outwards from boundary, with the remnant of the initial contact discontinuity maintained by the third, central, wave. Between the waves, the intermediate states are $\mathbf{Q}^*_{i-1}$ and $\mathbf{Q}^*_i$. The key contribution from Roe was the mechanism by which non-linear systems could be reformulated to use this property. This linearisation will be discussed later in Section 3.4. If one assumes for now that one has a hyperbolic conservation law, or system of laws, of the form

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial}{\partial x}\mathbf{F}(\mathbf{Q}) = 0, \tag{3.1}$$

where $\mathbf{Q}$ is the state, and $\mathbf{F}(\mathbf{Q})$ is a flux that is a function of the state, then it is possible to reformulate this with the Jacobian $\mathbf{A} \equiv \partial\mathbf{F}/\partial\mathbf{Q}$. The current equation is a non-linear PDE, since the flux term is not independent of the state. The new version with the substitution is now quasi-linear, as a suitably chosen Jacobian $\mathbf{A}$ will be only linearly dependent on the state. The quasi-linear form appears as

$$\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{Q}}{\partial x} = 0. \tag{3.2}$$

This new form can be said to be quasi-linear if the matrix Jacobian is at most linearly dependent on $\mathbf{Q}$. The matrix $\mathbf{A}$ holds all the necessary information to find the solution, as it contains the waves and wave speeds of the fundamental waves of the problem. The aim of the solver is to numerically find the solution to this system of equations $\mathbf{Q}(x, t)$, in both time and space.

To solve this numerically, one can divide the computational domain of size L into uniformly distributed N cells, with cell centres at positions $\mathbf{x} = (x_1, x_2, ..., x_N)$. The width of these cells is $\delta x = L/N$, with cell boundaries half way between each cell centre. The initial state $\mathbf{Q} = (\mathbf{Q}_1, \mathbf{Q}_2, ..., \mathbf{Q}_N)$ is known for each cell. At the boundary, between cells x_{i-1} and x_i, there is a Riemann problem, which is used to solve for the evolution of the state. These Riemann problems have well known exact solutions (Lax, 1957), which produce the flux at the cell boundary. A number of exact and approximate Riemann solvers have been developed to solve them (Godunov & Bohachevsky, 1959; Glimm, 1965; Harten, 1983; Leer,

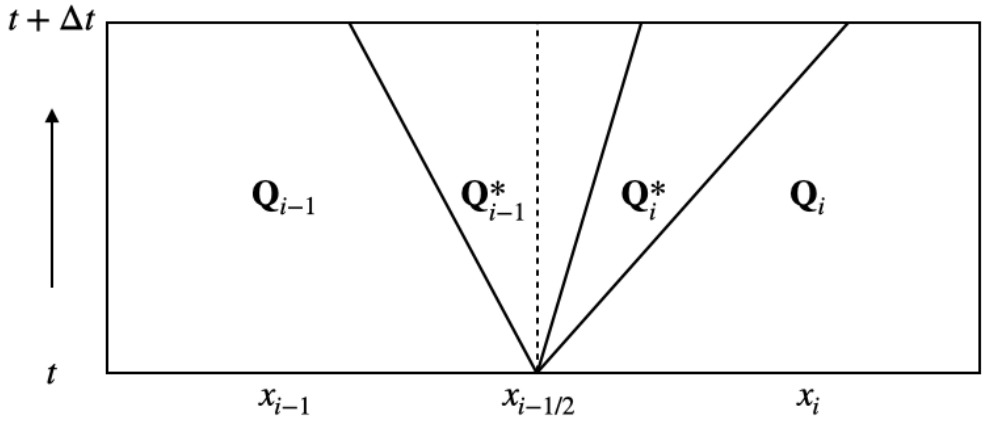

Figure 3.2 *The fundamental waves from a Riemann problem, between cells $i-1$ and i. The initial states of the discontinuity are $\mathbf{Q}_{i-1}$ and $\mathbf{Q_i}$, with intermediate states $\mathbf{Q}^*_{i-1}$ and $\mathbf{Q}^*_i$ between the waves. These waves are given by the eigenvectors of the Jacobian $\mathbf{A}$, and their speed by the corresponding eigenvectors.*

1984). The Roe solver is an example of an approximate Riemann solver. The approximation comes in the linearisation of the equations, for which the solver finds an exact solution (Roe, 1981). The set of Riemann problems within the whole domain can be solved separately, as long as the time step over which they are solved are small enough that the signal from one face has not interacted with another.

To find the solution to the discontinuity at the cell boundary, Roe uses an approximation of the Jacobian from the linearised form of the specific equations being solved. This approximates the Jacobian at the boundary between the two cells. The boundary Jacobian $\bar{\mathbf{A}}$ must satisfy a set of conditions (Roe, 1981). It is required that it be a linear mapping from $\mathbf{Q}$ to $\mathbf{F}$, that it converge on the exact Jacobian as the discontinuity decreases, that the eigenvectors of the boundary Jacobian matrix must be linearly independent, and that the net flux through the face can be written as

$$\bar{\mathbf{A}}(\mathbf{Q}_{i-1} - \mathbf{Q}_i) = \mathbf{F}_{i-1} - \mathbf{F}_i. \tag{3.3}$$

It is useful here to introduce the decomposition of the boundary Jacobian, which reformulates it as a function of it eigenvalues and eigenvectors

$$\bar{\mathbf{A}} = \mathbf{R}^{-1}\mathbf{\Lambda}\mathbf{R}, \tag{3.4}$$

where $\mathbf{R}$ is the matrix constructed by using the eigenvectors of $\bar{\mathbf{A}}$ as columns, and Λ is the a diagonal matrix of the corresponding eigenvalues. This is only possible if the matrix is diagonisable, which is an implicit condition on the matrix. To find the solution, the discontinuity that forms the Riemann problem at the cell boundary must be decomposed into the sum of contribution from its fundamental waves (Roe, 1981)

$$\mathbf{Q}_{i-1} - \mathbf{Q}_i = \sum_{p=1}^{q} \alpha_{i-1/2,p} \mathbf{e}_{i-1/2,p}, \tag{3.5}$$

where $\alpha_{\mathbf{i-1/2}}$ is the unknown wave strength of each wave. The eigenvectors are denoted by $\mathbf{e}_{i-1/2}$, where the p denotes the p^{th} eigenvector. The total discontinuity at the boundary is given by the sum of these wave strengths, which are individually found by projecting the original discontinuity onto the eigenvectors of the Jacobian, such that

$$\alpha_{i-1/2} = \mathbf{R}^{-1}_{i-1/2}(\mathbf{Q}_{i-1} - \mathbf{Q}_i), \tag{3.6}$$

where p represents the element of $\mathbf{Q}$, of which there are q. It is not necessary to calculate the intermediate states explicitly, since all the information that is required to get the solution to the problem is held in the wave strengths.

The above results combine to give all the information required to calculate the solution. Going back to the original set of equations that this approach is solving, it is simple to discretise the equations in time, using the Taylor expansion of the state. To first order accuracy, the state at time $t + \Delta t$ is given by the state at t, and the change in the state, with

$$\mathbf{Q}(x, t + \Delta t) = \mathbf{Q}(x, t) + \Delta t \frac{\partial \mathbf{Q}}{\partial t} + O(\Delta t^2). \tag{3.7}$$

Equation 3.1 can be substituted in for the temporal derivative, resulting in

$$\mathbf{Q}(x, t + \Delta t) = \mathbf{Q}(x, t) - \Delta t \frac{\partial \mathbf{F}}{\partial x} + O(\Delta t^2). \tag{3.8}$$

Removing higher order terms, and introducing the spatial discretisation imposed by the grid, the numerical update to the state at vertex i, from the boundary between $i - 1$ and i, and for time step number from n to $n + 1$, is given by

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n + \frac{\Delta t}{\Delta x}(\mathbf{F}_{i-1} - \mathbf{F}_i) = \mathbf{Q}_i^n + \frac{\Delta t}{\Delta x}\mathbf{F}_{i-1/2}, \tag{3.9}$$

which can be rewritten using the condition on the Jacobian and the formulation of the flux, given in Equations (3.3) and (3.5). Based on the boundary Jacobian, this becomes

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n - \frac{\Delta t}{\Delta x} \sum_{p=1}^{q} (\lambda^p)^+ \alpha^p \mathbf{e}^p, \tag{3.10}$$

where $(\lambda^p)^+$ is the positive eigenvalue only. Negative eigenvalues are replaced by zero. Since these eigenvalues represent the propagation speed of the wave from the discontinuity, excluding negative values effectively upwinds the solution. The equivalent update, to the state in the x_{i-1} cell, from this boundary, would use only the negative eigenvalues to achieve the same thing. For completeness, the total update to the state of cell i would be given by

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n + \frac{\Delta t}{\Delta x} \left(\mathbf{F}_{i-1/2} + \mathbf{F}_{i+1/2} \right) \tag{3.11}$$

which becomes

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n - \frac{\Delta t}{\Delta x} \left(\left[\sum_{p=1}^{q} (\lambda^p)^+ \alpha^p \mathbf{e}^p \right]_{i-1/2} + \left[\sum_{p=1}^{q} (\lambda^p)^- \alpha^p \mathbf{e}^p \right]_{i+1/2} \right). \tag{3.12}$$

The condition on the flux results in a conservative method, since the flow of the difference in the state $v(\mathbf{Q}_{i-1} - \mathbf{Q}_i)$, moving with velocity v, should be equal to the flux of material through the boundary $(\mathbf{F}_{i-1} - \mathbf{F}_i)$. This is only true for a linear set of equations. The construction of the Roe method centres around finding the suitable approximation of the Jacobian. This and the other conditions combine to guarantee that the method is conservative, can cope with shocks, and that it can smoothly recover the linearised version from the non-linear form. Identifying this matrix for an arbitrary set of equations is not necessarily trivial, since the flux condition given in Equation (3.3) is difficult to satisfy, but for our purposes the non-linear set of equations are the Euler equations, for which such a matrix has been found (Roe, 1981).

For the Euler equations, Roe produced a parameter vector that satisfies these requirements. For the standard fluid variables $\mathbf{Q} = (\rho, \rho v, \rho e)^T$, the Roe parameter vector is $\mathbf{Z} = (\sqrt{\rho}, \sqrt{\rho} v, \sqrt{\rho} H)^T$. The Eulerian flux vector has a quadratic dependence on these variables, which means that the Jacobian $\mathbf{A} = \partial \mathbf{F} / \partial \mathbf{Z}$ is now linearly dependent on the state variables. This partial differential produces a 3×3 square matrix, which satisfies the requirements to produce a conservative

scheme.

In summary, if the equation, or system of equations, that one is dealing with is linear, such that the highest order term, with respect to state $\mathbf{Q}$, has rank one, then it is possible to rewrite the boundary Riemann problem as a set of waves moving with velocities found from the *eigenvalues* λ of the Jacobian. These waves are moving discontinuities, with the difference in the state either side of the wave front found by projecting the initial difference in state at the boundary onto the *eigenvectors* $\mathbf{e}$ of the Jacobian.

1D Residual Distribution

It is possible to recast this method by redefining the net flux at a given boundary as the *residual* material that is left when the flow from one side is combined with material from the other. This residual ϕ^T is therefore defined as

$$\phi^T = \mathbf{F}_{i-1} - \mathbf{F}_i = \bar{\mathbf{A}}(\mathbf{Q}_{i-1} - \mathbf{Q}_i). \tag{3.13}$$

The residual is split between the cells either side of the boundary, in such a way that the distribution still sums to the original total. Similar to before, the split is achieved by using the positive and negative eigenvalues. Considering the matrix form of the discontinuity, given in Equation (3.5), one can see how this residual distribution comes about. The element residual is given by

$$\phi^T = \bar{\mathbf{A}}(\mathbf{Q}_{i-1} - \mathbf{Q}_i) = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^{-1}(\mathbf{Q}_{i-1} - \mathbf{Q}_i) = \sum_{p=1}^{q} \lambda^p \alpha^p \mathbf{e}^p. \tag{3.14}$$

As mentioned above, these are distributed by selecting only positive and negative eigenvalues respectively. Thus the residual distributed to cell i is given by

$$\phi_i = \phi_i^+ = \bar{\mathbf{A}}^+(\mathbf{Q}_{i-1} - \mathbf{Q}_i) = \bar{\mathbf{A}}^+\bar{\mathbf{A}}^{-1}\phi, \tag{3.15}$$

and similarly the residual sent to the $i-1$ cell is

$$\phi_{i-1} = \phi_{i-1}^- = \bar{\mathbf{A}}^-(\mathbf{Q}_{i-1} - \mathbf{Q}_i) = \bar{\mathbf{A}}^-\bar{\mathbf{A}}^{-1}\phi. \tag{3.16}$$

Together these two residuals sum to the total $\phi^T = \phi_i + \phi_{i-1}$, as required. In this formulation, originally known as *fluctuation splitting*, they are identical to the flux through the boundary of the cell. The important part of this solver, in

this setup, is its ability to be extended to more dimensions. To complete the formulation of this method that uses these residuals, the final form of the update to the state in cell i is given by the sum of residuals that are distributed to it from its boundaries. Explicitly, this is calculated using

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n - \frac{\Delta t}{\Delta x}\left(\phi_i^+ + \phi_i^-\right), \tag{3.17}$$

or in other words, the update comes from the summation of the residuals sent from the boundaries.

3.2.2 Residual Distribution in Higher Dimensions

Since the original formulation of this method, a significant amount of work has gone in to extending it to higher dimensions (Deconinck et al., 1993; Paillere et al., 1995; Abgrall & Roe, 2003; Abgrall & Marpeau, 2007; Ricchiuto & Abgrall, 2010). To achieve this, the key step is producing a consistent definition for the residual. In one dimension this is trivial, since the net flux through a cell boundary naturally fits with the idea of the residual flow, and the whole system is easily modelled using the set of waves described above. In the 2D case, it is not obvious how to define this variable. In this section, I will derive the basic form of the residual itself, in dimensions greater than one, and cover how this is transformed for use in the numerical method.

Notation

Before I go into the specifics of the derivation, there are a number of important terms to define with respect to the domain discretisation and geometry. The 2D space Ω is completely divided into a set of triangular elements T, with vertices (i, j, k), labeled counterclockwise. To continue the analogy to the 1D Roe method, in this discretisation the cells in the 1D case are now equivalent to the nodes of the triangulation, with the triangular elements taking the place of the cell boundaries when it comes to the calculation of the residuals. The significance of this transformation/comparison will be discussed in more detail later. The inner normals of the triangle edges are defined such that $\mathbf{n}_i$ is the inner normal to the edge between vertices j and k (see Figure 3.3). The labelling order of the vertices is only important in making sure the normals are orientated inwards.

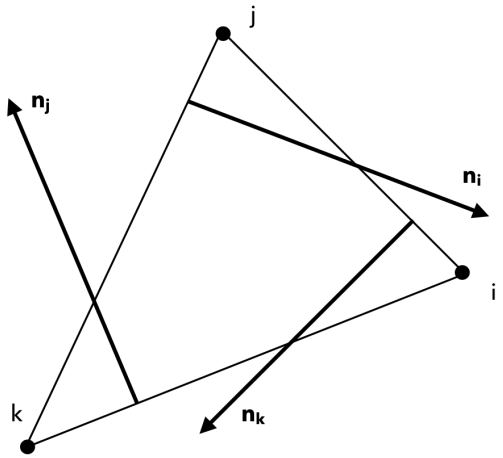

Figure 3.3 *Element vertices and associated normals*

These normals are found using

$$\mathbf{n}_i = (y_j - y_k)\hat{\mathbf{x}} - (x_j - x_k)\hat{\mathbf{y}}, \tag{3.18}$$

where $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are the x and y unit vectors respectively, with equivalent forms for the other vertices. It is also important to define the parameter $|S_i|$, representing the area of the dual cell of a vertex in a unstructured triangulation, shown in Figure 3.4, given by

$$|S_i| = \sum_{T|i\in T} \frac{1}{3}|T|, \tag{3.19}$$

for the dual cell of vertex i, summing over every triangle T with which i is associated. This dual cell is a Voronoi cell, part of the Voronoi tessellation that is the dual of the Delaunay triangulation. $|T|$ is the area of the triangle, given by

$$|T| = \frac{1}{2}|\mathbf{n}_i \times \mathbf{n}_j|, \tag{3.20}$$

where i and j are any two vertices of T. It is also important to define the specifics of the problem that is being solved. The system of partial differential equations depend on some set of continuous variables. For such a continuous variable $\theta(x, y, t)$, the equivalent discrete approximation is referred to as θ_h. The parameter h represents some characteristic length scale of an element, typically taken as the length of the longest edge, although this choice is somewhat free.

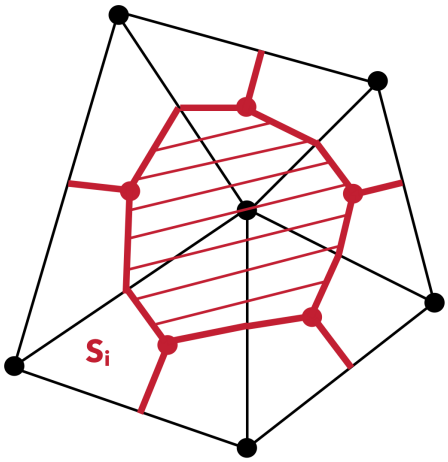

Figure 3.4 *Dual cell (red shaded area) of a vertex in an unstructured triangular mesh*

Residual - 1st Order in Time

As before, in Section 3.2.1, if one considers a set of linear partial differential equations, but now in more than one dimension, then the problem is formulated as

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathcal{F}(\mathbf{Q}) = \frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial}{\partial x}\mathbf{F}_x(\mathbf{Q}) + \frac{\partial}{\partial y}\mathbf{F}_y(\mathbf{Q}) = 0, \tag{3.21}$$

where $\mathcal{F}(\mathbf{Q}) = (\mathbf{F}_x(\mathbf{Q}), \mathbf{F}_y(\mathbf{Q}))$, or in its quasi-linear form as

$$\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{A_x}\frac{\partial \mathbf{Q}}{\partial x} + \mathbf{A_y}\frac{\partial \mathbf{Q}}{\partial y} = 0. \tag{3.22}$$

Here, as before, $\mathcal{A} = (\mathbf{A}_x, \mathbf{A}_y)$ represent the Jacobian matrices for the state variable $\mathbf{Q}$. The additional term obviously follows from the additional dimension. This form is quasi-linear because the Jacobian matrices are only *linearly* dependent on the PDE unknown $\mathbf{Q}$. This linear dependence is a condition placed on the Jacobian. I will only describe the 2D case here, but will expand on the extension to 3D in Section 4.3. The fundamental aim here, is to solve this system of equations without breaking the problem down with dimensional splitting.

The element residual, in this context, is now defined as the integral of the divergence of the numerical approximation of the flux (Deconinck et al., 1993; Ricchiuto & Abgrall, 2010). This follows naturally from the 1D definition, but

the integral is now over the triangular element, rather that across the boundary between two cells. This is given as

$$\phi^T(\mathbf{Q}_h) = \int_T \nabla \cdot \mathcal{F}_h(\mathbf{Q}_h) dxdy, \tag{3.23}$$

which, via the divergence theorem, is equivalent to

$$\phi^T(\mathbf{Q}_h) = \oint_{\delta T} \mathcal{F}_h(\mathbf{Q}_h) \cdot \mathbf{ndl}, \tag{3.24}$$

where $\mathbf{n}$ is the inward pointing normal to the triangle, with the integral performed around the edge of the triangle. This second form can aid in visualising what is being calculated.

The solution to this problem is assumed to be piece-wise linear across the triangular elements T of the triangulation $\mathcal{T}$ of the computational domain Ω, with the initial state known at the nodes, or vertices, of the triangulation. The approximate solution can be found by applying a P^1 Lagrange basis function ψ_i. This function is one at the node in question, and zero at every other node. In other words, for a given state variable, the solution is a flat plane that passes through the three vertices of the triangle. The analogous one dimensional equivalent would be a straight line between the two nodes of an element, which in this case would be a line. Using the notation described above, the approximation of the solution across the triangulation is given by

$$\mathbf{Q}_h = \sum_{i \in \mathcal{T}} \mathbf{Q}_i \psi_i, \tag{3.25}$$

or that the solution consists of the sum of the approximate solutions at every vertex i in the whole triangulation $\mathcal{T}$. These assumptions are basically just saying that the solution to the initial value problem is found over a triangulation of the computational domain, built around a set of vertices, where the solution is calculated. The solution is assumed to vary linearly between vertices. This allows one to formally define the discretised residual, or *local Galerkin residual*, for vertex i, as

$$\phi_i^G(\mathbf{Q}_h) = \int_T \psi_i \nabla \cdot \mathcal{F}_h(\mathbf{Q}_h) dxdy. \tag{3.26}$$

The residual for each triangular element is split between its vertices, with the sum of these residual from all the triangles of which a given node is a vertex, is

the update to the state at the position of that node.

In order to solve this numerically, a discreet mechanism of calculating this residual is required. First, the element residual can be rewritten as a function of the Jacobians in Equation (3.22), which becomes

$$\phi^T(\mathbf{Q}_h) = \int_T \mathbf{A}_x \frac{\partial \mathbf{Q}}{\partial x} + \mathbf{A}_y \frac{\partial \mathbf{Q}}{\partial y} dxdy. \tag{3.27}$$

A discreet method of finding the approximation of the change with space, and the Jacobian, are simply defined. Since we have broken the state up into a piece-wise linear reconstruction of the state, the spatial differential can be written in its discretised form as

$$\frac{\partial \mathbf{Q}}{\partial x} = \frac{1}{2|T|} \left(\sum_{i=1}^{3} \mathbf{Q}_i \mathbf{n}_i \right) \cdot \hat{\mathbf{x}}, \tag{3.28}$$

where the sum is over the three vertices of the element, $\mathbf{Q}_i$ is the state at each vertex, and $\mathbf{n}_i$ is the inward pointing normal to the edge opposite the vertex i. The unit vector in the x-direction is $\hat{\mathbf{x}}$. This amounts to finding the linear interpolation of the state in the element from the state at the three vertices. The y-direction equivalent is the same, but replacing all occurrences of x with y. These partial differentials are now constant across the element. The Jacobian is a little harder to define. Substituting in the above equation, the residual becomes

$$\phi^T = \frac{1}{2|T|} \left[\left(\sum_{i=1}^{3} \mathbf{Q}_i \mathbf{n}_i \right) \cdot \hat{\mathbf{x}} \int_T \mathbf{A}_x dxdy + \left(\sum_{i=1}^{3} \mathbf{Q}_i \mathbf{n}_i \right) \cdot \hat{\mathbf{y}} \int_T \mathbf{A}_y dxdy \right]. \tag{3.29}$$

Analogous to the Roe solver, it is possible to define an average Jacobian. In the 1D case, this was the average at the boundary, and in this 2D case it is the average over the element. This is therefore defined as the integral of $\mathbf{A}_x$ and $\mathbf{A}_y$ over the area of the element, divided by that area

$$\bar{\mathbf{A}}_x = \frac{1}{|T|} \int_T \mathbf{A}_x dxdy, \tag{3.30}$$

which leaves the residual as

$$\phi^T = \frac{1}{2} \left[\left(\sum_{i=1}^{3} \mathbf{Q}_i \mathbf{n}_i \right) \cdot \hat{\mathbf{x}} \bar{\mathbf{A}}_x + \left(\sum_{i=1}^{3} \mathbf{Q}_i \mathbf{n}_i \right) \cdot \hat{\mathbf{y}} \bar{\mathbf{A}}_y \right]. \tag{3.31}$$

The dot product of the normal with the unit vectors mean only that component

is included in each sum. Using this, combining the summations, and bringing the Jacobian inside the sum gives

$$\phi^T = \frac{1}{2}\sum_{i=1}^{3}\left(\mathbf{Q}_i\bar{\mathbf{A}}_x n_{x,i} + \mathbf{Q}_i\bar{\mathbf{A}}_y n_{y,i}\right). \tag{3.32}$$

Finally, it is possible to combine the Jacobians into a single term $\mathbf{K}_i = (\bar{\mathbf{A}}_x n_{x,i} + \bar{\mathbf{A}}_y n_{y,i})/2$, with the dependence on vertex i coming from the normal of the opposite edge. This simplifies the calculation of the element residual to the sum of the product of matrix $\mathbf{K}_i$ and state $\mathbf{Q}_i$

$$\phi^T = \sum_{i=1}^{3}\mathbf{K}_i\mathbf{Q}_i. \tag{3.33}$$

Now all that remains is a method to calculate the discreet form of the element Jacobian $\bar{\mathbf{A}}$. Since it has been required that the system of equations is linear, then the Jacobian $\mathbf{A}_x = \partial\mathbf{F}/\partial\mathbf{Q}$ will vary linearly. This means the element Jacobian can simply be computed as the Jacobian as a function of the average state of vertices of that element $\bar{\mathbf{A}}_x = \mathbf{A}_x\left(\bar{\mathbf{Q}}\right)$, where the average state is simply $\bar{\mathbf{Q}} = \frac{1}{3}\sum_{i=1}^{3}\mathbf{Q}_i$.

The combining of the Jacobian into $\mathbf{K}_i$ is the key step that makes this a truly multi-dimensional method. A similar method that considers the x and y Jacobians separately would effectively be splitting the problem by dimension, which is what this method avoids. While this is the cornerstone of how these methods work, a lot of the work that has gone into these methods has centered around how to use this residual to calculate the update to the state, and so evolve the solution.

I now have the discreet method for the calculation of the residual, for any triangular element. The key question now is how to distribute these to the vertices of the element. I will cover this in more detail shortly, but first let us assume we have some method for distributing the residual in a way that maintains conservation, and achieves upwinding. The residual is therefore split into three values, one sent to each vertex, such that $\phi^T = \phi_i + \phi_j + \phi_k$. Once again, the discreet update to the state is found using the discretised form of the Taylor expansion of the solution. To derive this form, the expansion starts off as before, with Equation (3.7), but with the additional dependence on y. The discreet

update formula becomes

$$\mathbf{Q}_i^n = \mathbf{Q}_i^{n+1} - \frac{\Delta t}{|S_i|} \sum_{T|i \in T} \phi_i \tag{3.34}$$

where the summation is over all triangles for which node i is a vertex. The area $|S_i|$ is the area associated with the updated vertex, defined by assigning one third of the area of each connected triangle. This area is also the area of the cell that is formed by the dual of the triangulation.

Distribution - 1st Order in Time

When designing the distribution schemes, which are the next integral part of the method, it is desirable that the scheme exhibit certain properties. Typically these include being conservative, obviously very important for our consideration of the fluid equations, and preserving linearity and positivity (Deconinck et al., 1993; Ricchiuto & Abgrall, 2010; Abgrall, 2012). A linearity preserving scheme will recover the exact solution for a linear set of equations, and a positive scheme will be total variation diminishing (TVD). A method is TVD if the sum of the differences between the numerical approximation of the solution, and the exact solution, decreases from one time step to the next. For some set of initial conditions, a scheme is positive if it does not introduce new maxima or minima to the solution. For this to be true, any variation from the exact solution that is introduced by the approximation must get smaller over time, and so the scheme is TVD. This positive characteristic is also known as being monotone. It is impossible to construct a linear scheme that is both positive and linearity preserving (Ricchiuto & Abgrall, 2010), so non-linear approaches must be formulated to achieve both desired properties at once. To be clear, a linear *scheme* mentioned here is one in which the solution can be expressed as a sum of the initial state, weighted by coefficients that do not depend on the state itself. This is not the same as the problem itself being linear. I will describe three examples of widely used schemes:

- LDA Scheme - Linear, low diffusion
- N Scheme - Linear, positive
- B Scheme - Non-linear, blending of the two other schemes

A widely used example of a relatively simple scheme is the low diffusion A (LDA) scheme (Struijs et al., 1991; Caraeni & Fuchs, 2002; Deconinck & Ricchiuto, 2007). It achieves second order accuracy in space, but in achieving this it sacrifices its total variation diminishing capability. This results in spurious oscillations in the presence of discontinuities, but as the name suggests, it is constructed to have low numerical diffusion, making it viable for smooth flows. The nodal residual, the part of the element residual sent to each vertex, is found using (Csik et al., 2002)

$$\phi_i^{LDA} = \beta_i \phi^T = \frac{\mathbf{K}_i^-}{\sum_i^3 \mathbf{K}_i^-} \phi^T. \tag{3.35}$$

To guarantee that this is conservative, it is only required that the distribution coefficients sum to unity, which ensures the distributed residual sum to element residual.

Another simple scheme is the N scheme, which is designed to be positivity preserving, so is TVD, and so does not experience the oscillations around discontinuities. The scheme is only first order accurate in space (Ricchiuto & Abgrall, 2010), so has significantly greater numerical diffusion, compared to the LDA scheme. Such a scheme is obviously most well suited to problems with shocks. In this case, the nodal residual is given by (Struijs et al., 1991)

$$\phi_i^N = \mathbf{K}_i^+ \left(\mathbf{Q}_i - \frac{\sum_{j=1}^3 \mathbf{K}_j^- \mathbf{Q}_j}{\sum_{j=1}^3 \mathbf{K}_j^-} \right). \tag{3.36}$$

Both the above schemes are linear, and so cannot be both positivity and linearity preserving. As mentioned above, this leads to schemes that, in general, have either strong numerical diffusion, or weak shock handling capabilities. A number of non-linear schemes have been developed (Csik et al., 2002; Abgrall & Roe, 2003; Dobes & Deconinck, 2008), which build on these two linear schemes by combining them. The result preserves the advantages of each scheme, and reduces the disadvantages, by introducing a blending coefficient that can be designed to detect when the conditions are best suited to each method. A number of possible blendings have been developed, but they are built around the same idea of constructing the distribution around

$$\phi_i^B = \Theta \phi_i^N + (I - \Theta) \phi_i^{LDA} \tag{3.37}$$

where I is the identity matrix, and Θ is the diagonal blending matrix. This matrix is constructed by setting

$$\Theta_{ii} = \frac{|\phi_i^T|}{\sum_{j=1}^{3} |\phi_{j,i}^N|}, \tag{3.38}$$

where the sum is over every vertex of element T, and the i index refers to the ith equation of the system. In this way, the change in each equation of the set is tested separately for the blending. The matrix effectively compares the full element residual to the sum of nodal residuals. If there is a rapid flow at the vertices, then there will likely be very positive and very negative nodal residuals. When summed in absolute form this will produce a larger number than the absolute value of the total, and so make the contribution from the N solver small, with its higher diffusion, and instead prioritise the result from the LDA solver.

Different applications of this blending matrix put different conditions on the matrix values. The Bmax and Bmin schemes (Csik et al., 2002; Paardekooper, 2017) simply replace every diagonal value of the blending matrix with either the maximum or minimum value of the blending matrix. Using Bmax will default towards the N scheme, while Bmin defaults to LDA. The so called Bx (Dobes & Deconinck, 2008) scheme replaces the diagonal values with ones calculated with a shock sensor. This sensor detects when there are two colliding flows, where a shock will develop. Where these flows are detected, the N scheme will be more heavily weighted. In all other conditions, the solution will use the LDA scheme.

Residual - 2nd Order in Time

The first order RD methods were largely developed to treat steady problems (i.e. ones where the solution converges on some steady state) (Paillere et al., 1995; Hubbard & Baines, 1997; Dobes & Deconinck, 2008). For these methods, having only first order accuracy in time is acceptable, but for problems with significant time variation, it is important to achieve second order accuracy in time. The order in time refers to the highest order of term included in the approximation of the solution. The solution can be written as a Taylor expansion

$$\mathbf{Q}(x, t + \Delta t) = \mathbf{Q}(x, t) + \Delta t \frac{\partial \mathbf{Q}}{\partial t} + \frac{1}{2}\Delta t^2 \frac{\partial^2 \mathbf{Q}}{\partial t^2} + O(\Delta t^3). \tag{3.39}$$

A method that is first order in time if it only uses the term linearly dependent on Δt. The second order schemes include the terms dependent on Δt^2. A number of systems that achieve this have been developed (Abgrall & Roe, 2003; Palma et al., 2005; Rossiello et al., 2009; Ricchiuto & Abgrall, 2010). This work includes extending the schemes described above to allow for second order temporal accuracy, and perform extensive studies of the various properties of the new system. Below I will summarise the extension to second order temporal accuracy, as well as the potential options that have been developed to implement this extension.

When dealing with a time dependent problem, there is clearly going to be a time dependence in the residual itself. In the first order formulation, the element residual was defined as the integral of the divergence of the flux over the element. To include the time dependence in the residual, it is necessary to define a new residual, the *total residual* Φ^T, which is the integral over the whole set of equations

$$\Phi^T(\mathbf{Q}_h) = \int_T \left[\frac{\partial \mathbf{Q}_h}{\partial t} + \nabla \cdot \mathcal{F}_h(\mathbf{Q}_h) \right] dxdy = \int_T \frac{\partial \mathbf{Q}_h}{\partial t} dxdy + \phi^T(\mathbf{Q}_h). \tag{3.40}$$

This residual now contains a way to take into account the change in the state over the time step. There is some inconsistency in the notation and naming conventions within the residual distribution field, but here I will use the above naming scheme, where the element residual ϕ^T is the area integral of the divergence, and the total residual Φ^T is the integral of the whole equation. The integral over the time derivative simply becomes the mean of the time derivatives of the solution at each node of the element multiplied by the area

$$\Phi^T = \sum_{j=1}^{3} \frac{|T|}{3} \frac{d\mathbf{Q}_j}{dt} + \phi^T. \tag{3.41}$$

This still contains the time derivative of the state, which is simply taken as the absolute change in the state at vertex j for time step $\Delta \mathbf{Q}/\Delta t$. The distribution of this new residual requires a way to distribute the time dependent part. This is achieved by applying a mass matrix $\mathbf{m}$ (Caraeni & Fuchs, 2002; Palma et al., 2005; Ricchiuto & Abgrall, 2010), which sets a fraction of the contribution from the temporal part of the total residual to be sent to each vertex. This is used to

find the nodal total residual with

$$\Phi_i^T = \sum_{j=1}^{3} m_{ij} \frac{d\mathbf{Q}_j}{dt} + \phi_i^T. \tag{3.42}$$

There are a number of choices for the mass matrix that offer different dissipative properties (Ricchiuto & Abgrall, 2010). Naturally these also depend on the chosen distribution scheme. The first, and simplest, of these is found by replacing the element residual in the first order method with the total residual (Caraeni & Fuchs, 2002). If we take the LDA scheme, the mass matrix then simply becomes the

$$m_{ij}^{F1} = \frac{|T|}{3}\beta_j, \tag{3.43}$$

where β_j is the LDA distribution matrix for the jth vertex of that element. This splits the total residual in exactly the same way as the first order LDA scheme, with the addition of the temporal part to the distributed residual. The second formulation of the mass matrix is derived by writing the discreet equations in a way that is analogous to a stabilised Galerkin finite element scheme (Cohen et al., 2001). These schemes apply a test function to the problem equations, where the test function is a polynomial of order equal to desired reconstruction order. The resultant conditions produced for the test function produce the solution to the set of equations. By applying the same idea to the total Galerkin residual (Ricchiuto & Abgrall, 2010), and using a test function dependent on β_j, the end result of this approach is to produce a mass matrix of the form

$$m_{ij}^{F2} = \frac{|T|}{36}(3\delta_{ij} + 12\beta_i - 1), \tag{3.44}$$

where the delta function δ_{ij} is zero except when $i = j$. Another option (Palma et al., 2005) is to assume that the mass matrix - time derivative term must be equal to the integral of the time derivative of the state, at a given vertex, over the fraction of the element associated with that vertex. This method results in two possibilities (Ricchiuto & Abgrall, 2010)

$$m_{ij}^{F3} = \frac{|T|}{3}\beta_i(\delta_{ij} + 1 - \beta_j), \tag{3.45}$$

and

$$m_{ij}^{F4} = \frac{|T|}{3}\beta_i(1 - \delta_{ij} + \beta_j). \tag{3.46}$$

These methods all produce consistent mass matrices, i.e. ones where the sum of the total nodal residuals is equal to the total residual. This is required to ensure conservation is maintained.

The RD method can now be recast as as the distribution of this new total residual. To achieve the second order accuracy in time, a Runge-Kutta time stepping scheme is applied. These methods function by constructing an intermediate state, and then finding the final state, for a given time step, as function of the original and intermediate states. Second (RK2), third (RK3), and fourth (RK4) order Runge-Kutta methods have been developed for the RD approach, but the additional computational costs of the RK3 and RK4 methods do not show a significant improvement in the accuracy of the numerical results (Ricchiuto & Abgrall, 2010), so I have only considered the RK2 approach here. For a generic problem of the form

$$\frac{dq}{dt} + e(q) = 0, \tag{3.47}$$

where $e(q)$ represents the evolution operator. For the time step n to $n+1$, the intermediate q^* and final q^{n+1} states are found by

$$\begin{aligned} &\frac{q^* - q^n}{\Delta t} + e^* = 0 \\ &\frac{q^{n+1} - q^*}{\Delta t} + \frac{1}{2}\left(e^{n+1} + e^*\right) = 0. \end{aligned}$$

For the RD problem, for the time step n to $n+1$, the intermediate state is constructed using the first order solver

$$\mathbf{Q}_i^* = \mathbf{Q}_i^n - \frac{\Delta t}{|S_i|} \sum_{T|i\in T} \phi_i \tag{3.48}$$

and the final state is found using the distribution of the total residual with

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^* - \frac{\Delta t}{|S_i|} \sum_{T|i\in T} \Phi_i \tag{3.49}$$

where the total residual is calculated based on both the initial and intermediate element residuals, in the standard RK2 form (Ricchiuto & Abgrall, 2010). The

second sub-step update becomes

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^* - \frac{\Delta t}{|S_i|} \sum_{T|i\in T} \left(\sum_{j=1}^{3} m_{ij} \frac{\mathbf{Q}_i^* - \mathbf{Q}_i^n}{\Delta t} + \frac{1}{2} \left(\phi_i(\mathbf{Q}_i^*) + \phi_i(\mathbf{Q}_i^n) \right) \right). \tag{3.50}$$

This provides all the information needed to construct a second order RD solver. the new total residual is only dependent on the initial state, the intermediate state, the time step, and the element residual for both the initial and intermediate states. As such there is no need to define specific second order forms for the different distribution schemes.

3.2.3 Choice of Method

In the previous section, I describe a number of different approaches that all come under the umbrella of RD methods. When implementing a RD hydro-solver, a number of choices have to be made. These can be made sequentially, based on the type of problem that is being attacked, and the computational resources that are available. The first choice is which distribution scheme to use. As discussed above, the various schemes have implicit advantages and disadvantages, with the N scheme best suited to problems with strong shocks, and the LDA better suited to problems with strong flows. If one of the blending schemes is chosen, then one has to decide on the blending mechanism, once again deciding which of the blended schemes should be favoured.

The second choice comes down to the desired temporal accuracy order. First order solvers were largely designed to solve problems that converge on steady solutions, which is the case for certain tests, but in astrophysical contexts such situations are rare. When choosing to use a second order solver, however, one must then decide which mass matrix formulation to use. This decision is less clear cut, as the stability of the various options are not well understood at this time.

3.3 Mesh

The approaches described above are built around an arbitrary set of static tracer positions $\mathbf{r}$, or vertices, which require a set of simplices that fill the

periodic domain, without gaps, and without overlapping edges/faces. These simplices effectively tell one the neighbours of any given vertex. Producing such a set of simplices has been studied in detail for a variety of uses, from astrophysical simulations such as these (Springel, 2010; Duffell & MacFadyen, 2011; Paardekooper, 2017), to graphical modelling and animation. In two dimensions, these simplices are triangles, hence the description as a triangulation. A given distribution of vertices can have a large number of possible meshes that fulfill the above criteria, but many of these will have undesirable characteristics.

3.3.1 Structured Mesh

For certain sets of vertices, it is possible to setup meshes, where the neighbours can be perfectly predicted by an arithmetic algorithm. An example of this would be a uniform grid of points in a Cartesian grid, with $N_x \times N_y$ vertices, across a domain of sides $L_x \times L_y$. With such a set, the neighbours of a given vertex can be found by simply moving by displacing by $\pm L_x/N_x$ in the x-direction, or $\pm L_y/N_y$ in the y-direction. This produces a mesh made of pairs of right angle triangles. Alternatively, a similar distribution, but with the vertices of the odd rows offset by half L_x/N_x in the x-direction. A set of interlocking equilateral triangles are created by these vertices. The left and middle panels of Figure 3.5 show these structured meshes. Equivalent triangulations can be built around circular coordinates. Structured meshes are simple to construct, but they severely limit the point distributions that can be used. A potential disadvantage of structured meshes comes from the inevitable alignment of edges. When used for fluid calculation, this can lead to preferential directions in the calculated flow. If the edges of cells are aligned, spurious structures can form, and flows not aligned with these directions can be suppressed.

3.3.2 Unstructured Mesh

If the vertex distribution has no simple pattern, such as those described above, then an unstructured mesh must be built instead. The ability to construct such a mesh allows one to use any arbitrary set of vertices, opening up the possibility to describe many complex geometries, and allow for resolution variation at any position. These unstructured meshes avoid the problems, mentioned previously, of preferential flow directions, as there is no pattern in the alignment of edges.

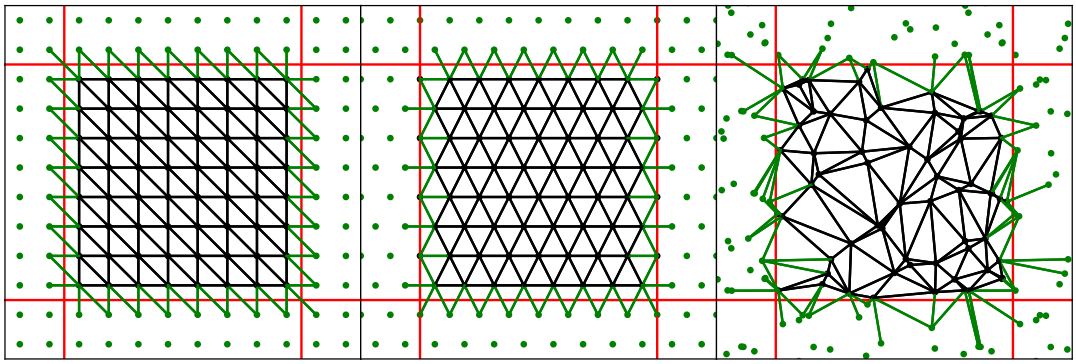

Figure 3.5 *Left panel: Structured triangular mesh based on vertices in a Cartesian grid. The periodic boundaries of the cells are indicated by the red lines, and green triangles show those identified as boundary triangles. Middle panel: Structured triangular mesh based on vertices in a Cartesian grid, but with odd rows offset by half the grid scale. Right panel: Unstructured Delaunay mesh built around an random distribution of vertices*

Since there are often a large number of possible meshes that cover the domain, it is useful to define conditions that prioritise features which, in the case of a hydro-solver, improve the accuracy of the method. One important feature to consider is the elongation of the triangles in a mesh. Triangles that are extremely stretched end up linking vertices that are relatively far apart, potentially leading to spurious local flows (Springel, 2010). This can be minimised by maximising the *minimum opening angle* of the triangles in the mesh. Highly elongated triangles have one large angle and two very small angles, so by maximising the minimum angle of a given triangle, one can minimise elongation. The optimal solution to satisfy this effort is obviously created by equilateral triangles, but for an arbitrary distribution of vertices, equilateral triangles will not be possible in the vast majority of cases.

It is worth, at this point, briefly discussing periodic boundaries. A mesh can simply be built around the vertices in an infinite domain, with no reference to any boundaries beyond those dictated by the vertices. However, many test problems, as well as astrophysical scenarios, utilise periodic boundary conditions, where the edges of the domain wrap around from one side to the edge on the opposite side of the domain. This can prevent spurious results propagating from either an open or closed boundary at the edge of the domain. To achieve this, the typical approach is to add a set of ghost vertices beyond the boundary of the domain. These replicate the positions of vertices on the other side of the domain, shifted by the edge length of the domain, providing the correct geometry for calculating the flow through the boundary. For the hydro-solvers, any update that is passed

to these vertices is copied to the real vertex that the ghost represents, completing the loop and allowing material to flow out one edge and in the other, as if there was an identical copy of the domain on either side. These ghost particles are constructed in all directions from the domain, effectively leading to nine copies of the original mesh, with the original in the middle, and eight surrounding it forming a 3×3 square. Examples of these ghost particles are shown in the panels of Figure 3.5 as green vertices, where the boundaries are shown in red.

3.3.3 Delaunay Mesh Construction

In this project, the discretisation of the gas will be done using a Delaunay triangulation, built around a set of tracer vertices. This triangulation, by definition, maximises the minimum opening angle. There are several well documented methods of constructing the Delaunay triangulation, and here I will describe several of the most well used. These methods, and the corresponding numerical algorithms, have been extensively tested (Springel, 2010; Cheng et al., 2016; Duffell, 2016). It is important to understand the mathematical underpinnings of what a triangulation is, and how it is chosen and built. The rigorous definition of a two dimensional triangulation of a point set is given in the following definition (Cheng et al., 2016)

Definition 1 *Let S be a finite set of points in a plane. A triangulation of S is a simplicial complex $\mathcal{J}$ such that S is the set of vertices in $\mathcal{J}$, and the union of all simplices in $\mathcal{J}$ is the convex hull of S.*

This leads us to three important terms, *convex hull*, *simplex*, and *simplicial complex*, which are defined as follows

Definition 2 *(Convex Hull) A convex combination of points in X is a point that can be written as an affine combination with all weights non-negative. The convex hull of X is the set of all convex combinations of points in X. (An affine combination of points in X is a point p that can be written $p = \sum_{i=1}^{k} w_i x_i$ for a set of scalar weights that sum to 1.)*

Definition 3 *(Simplex) A k-simplex τ is the convex hull of a set of $k+1$ affinely independent points (i.e. if none of the points are affine combinations of the*

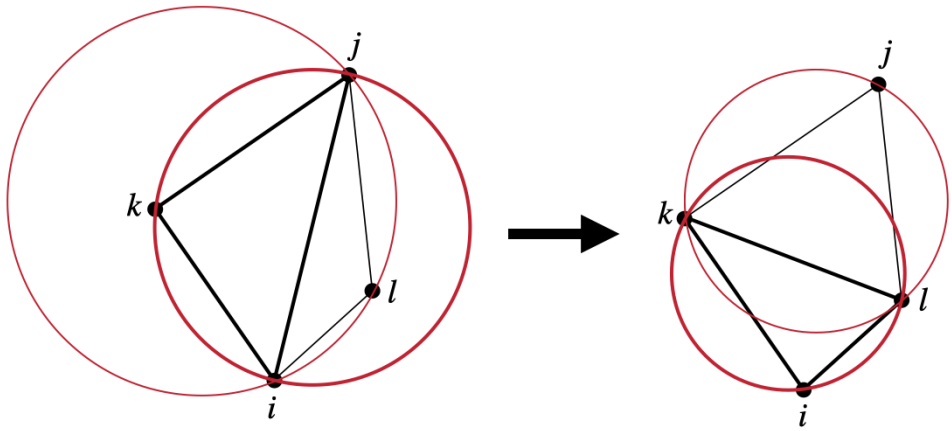

Figure 3.6 *Edge flip process. Moving edge ij to lk changes the two non-empty circumdisks of triangles ijk and ilj into two triangles (ilk and ljk) with empty circumdisks.*

others). Specifically, a 0-simplex is a vertex, a 1-simplex is an edge, a 2-simplex is a triangle, and a 3-simplex is a tetrahedron.

Definition 4 *(Simplicial Complex) A simplicial complex $\mathcal{J}$, also known as a triangulation, is a set containing a finite number of simplices that satisfy the following restrictions*

- *$\mathcal{J}$ contains every simplex in $\mathcal{J}$*
- *For any two simplices $\sigma, \tau \in \mathcal{J}$, their intersection $\sigma \cap \tau$ is either empty or a face of both σ and τ*

These four definitions give us the basic language behind formally discretising a space, about a set of points, into a set of cells. In other words, the simplex is the 'simplest' shape in a given number of dimensions, one in which all vertices are connected by edges to all other vertices, and a simplicial complex is a set of these simplices where no edges cross, and the elements completely fill the domain. Another key concept is the idea of a *circumdisk*, or *circumshpere*, defined here for d-dimensions.

Definition 5 *(Circumball) Let τ be a simplex embedded in $\mathbb{R}^d$. A circumball, or circumscribing ball, of τ is a d-ball whose boundary passes through every vertex of τ. Its boundary, a (d-1)-sphere, is called a circumsphere, or circumscribing sphere, of τ.*

In the left hand side of Figure 3.6 shows an example of a circumdisk for triangle ijk as the thick red line. We now have the important properties required to define our tessellation. This is known as the Delaunay Triangulation, where the important property is whether it can be defined as Delaunay.

Definition 6 *(Delaunay property) In the context of a finite point set S is characterized by the empty circumdisk property: no point in S lies in the interior of any triangle's open circumscribing sphere (see definition 5).*

An example of a non-empty circumdisk is shown on the left hand side of Figure 3.6. The red circles show the circumdisks for ijk and ilj, both of which enclose vertices that are not part of that triangle. On the right hand side, I show two examples of triangles with empty circumdisks. The circumdisks of triangles ilk and ljk do not contain any external vertices, and so these triangles satisfy the Delaunay condition. The Delaunay triangulation will exist uniquely for any set of points, except in a small set of scenarios where the open circumdisk is not empty because of an additional vertex just on the edge of the disk (Cheng et al., 2016). In these cases the Delaunay condition can be relaxes to only require the closed circumdisk to be empty. An example of this is in the case of the Cartesian grid of points. The open circumdisk for three of the four vertices of a square will always enclose the fourth vertex of the square, as the edge will just touch it. Changing to the closed circumdisk excludes interloper that are right on the edge, and so the condition is satisfied. For such structured sets of points, the Delaunay triangulation is not unique. In these cases, algorithms that construct the mesh will be set to always pick one option, as the specific choice between possibilities is not important. Otherwise the algorithm could get stuck in an endless loop.

There are several characteristics of the Delaunay triangulation that make it appealing for fluid discretisation. The Delaunay condition of requiring an empty circumdisk innately maximises the minimum opening angle, and also minimizes *largest circumdisk*. This has a similar effect on minimizing distortion, but specifically for cells with obtuse angles. Such angles may still exist in an arbitrary

point distribution. To further reduce distortion, methods for shifting the vertices of the tessellation, without reducing the physical accuracy of the simulation, have been proposed (Springel, 2010). Such an approach attempts to regularise the distribution of vertices, shifting the vertices in ways that would make distribution more uniform.

In this way, the Delaunay triangulation is formally defined, allowing the development of algorithms that can construct meshes with this property. There are a variety of methods and approaches that can be used. The most fundamental of these are laid out below. I also briefly describes the corresponding dual tessellation that is commonly used in hydrodynamics solver on unstructured meshes.

Flip Algorithm

This method is very simple to implement, and is more of a tool used by other methods than a method in its own right. A random triangulation is applied to the set of points. Each edge is flipped until the associated simplices are Delaunay (i.e. their open circumspheres are empty of other vertices). This process is shown in Figure 3.6, where flipping one edge (ij to lk) converts two non-Delaunay triangles into Delaunay triangles. While simple, this method can be inefficient if the initial triangulation is far removed from the Delaunay triangulation (Cheng et al., 2016). However, the fundamental technique of flipping edges can be used in other methods, and this method is considered robust, scaling with $O(N^2)$. It can be shown (Cheng et al., 2016), that this method will always recover the Delaunay triangulation, eventually, no matter the starting tessellation.

Gift Wrapping

The Gift Wrapping method builds a Delaunay triangulation around a predetermined Delaunay triangulation. The new simplices *crystallize* around the known triangulation through a gift wrapping step (Cheng et al., 2016). The step takes an edge, creates a shrinking circumsphere for the $d-1$ vertices of the edge that keeps shrinking until it contains no other vertices. The last vertex to lie within the disk becomes the additional vertex of the new simplex. If it never contains another vertex then the edge is part of the boundary of the domain.

Since this has to be done with every edge in turn it can be slow. In three dimensions, the worst case scenario is scaling with N-points as N^3 (Cheng et al., 2016). The main use of this method is often to fill a cavity that has been cleared, for instance when some set of points has moved, making the original triangulation non-Delaunay.

Divide and Conquer

This algorithm takes the whole set of particles, then divides them into two halves, with a line down the middle. The Delaunay triangulation of each half is then calculated, after which the two halves are stitched back together, merging the triangulations into a single structure. This method works extremely well in two dimensions, and has been demonstrated to have a $O(NlogN)$ dependence, in the worst case scenario (Cheng et al., 2016). It is not fast when extended into three dimensions.

Incremental Insertion

The Incremental Insertion method adds vertices one by one. The vertex to be inserted can be chosen entirely randomly, but a weight is often applied to insert vertices close to the existing edges. The Delaunay nature is then restored to the triangulation, before another point is inserted. After point insertion, all parts of the triangulation that the new point disrupts must be removed, creating a cavity within the triangulation. This cavity is then filled with the new triangulation about the new vertex. The worst case scenario for this method is scaling with N^2, and in practice these methods produce the fastest 3D Delaunay construction algorithms (Cheng et al., 2016).

Voronoi Tessellation

There is a second tessellation that can be built from the initial Delaunay triangulation. This is the Voronoi tessellation. Simply put, it is built by bisecting all the edges of the Delaunay triangulation with a new set of edges. These then form the edges of the new cells, with the old vertices forming a central point of each new cell. A visualization of a Delaunay triangulation, and its corresponding Voronoi tessellation is shown in Figure 3.7. Constructing such a tessellation

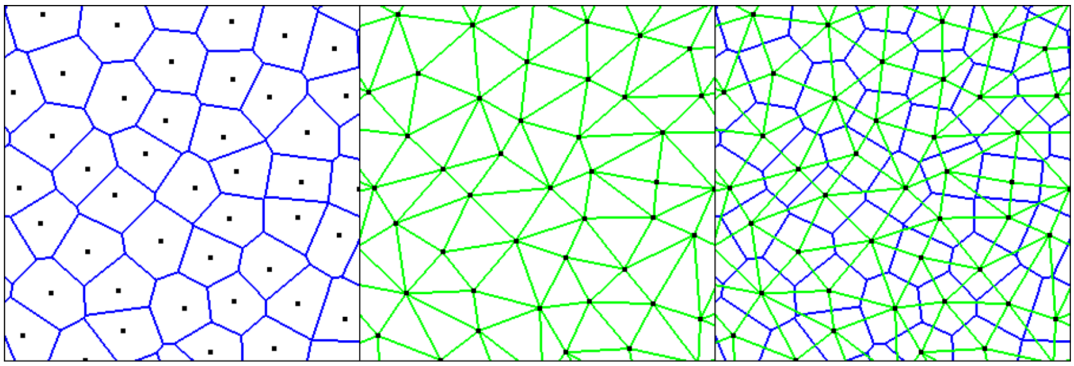

Figure 3.7 *Delaunay triangulation (center) and corresponding Voronoi tessellation (left) of a set of points, with an one overlaid on the other (right) (Duffell, 2016)*

from the Delaunay counterpart will scale linearly, since it is a relatively simple arithmetic operation, and increasing the number of vertices does not, on average, increase the number of neighbours that to which a vertex is connected.

It is the Voronoi tessellation that is currently used by most hydrodynamics schemes that use unstructured meshes, often as part of a moving mesh (Springel, 2010; Hopkins, 2015; Duffell, 2016). For these schemes, the shape of the Voronoi cells, with their many faces, is ideal for modeling complex hydrodynamical flows. In those methods, the fluid equations are effectively solved through the faces of the cells. For our purposes however, with the residual distribution method, we only require the construction of the initial Delaunay mesh. This is because the residual distribution method does not require the faces of the cells to calculate flux, but instead only requires the vertices of the tessellation, which means the method requires one less step. The Voronoi tessellation is built on top of the Delaunay triangulation, so avoiding having to build it will inevitably save time, and so increase computational efficiency.

CGAL

In practice, the various construction mechanisms are often used in combination to produce the most efficient algorithm. A number of extensive libraries have been developed to produce periodic Delaunay triangulations in both 2D and 3D. I have integrated triangulation construction from the QHULL (http://www.qhull.org) and CGAL (https://www.cgal.org/) libraries, with the focus on CGAL, due to its inclusion of 3D periodic triangulations, which will be required in Chapter 4.

3.3.4 Moving Meshes

As mentioned before, some hydro-solvers are built into so-called 'moving mesh' systems. In this approach, the underlying grid can move with the flow of the fluid. If the mesh is allowed to move freely, it must by definition be unstructured, as any initial structure cannot inhibit the motion of the cells. In this context, a method that is explicitly built for unstructured meshes, such as the RD family of solvers, is naturally suited for adaptation onto a moving mesh. This provides a significant possibility for future development of these solvers.

3.4 Euler Equations and Residuals

The specific set of partial differential equations we will be solving are the 2D Euler equations, which model the behavior of inviscid, compressible fluids. For this situation, the components of Equation (3.21) are given by

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho E \end{pmatrix}, \quad \mathbf{F}_x(\mathbf{Q}) = \begin{pmatrix} \rho v_x \\ \rho v_x^2 + P \\ \rho v_x v_y \\ \rho v_x H \end{pmatrix}, \quad \mathbf{F}_y(\mathbf{Q}) = \begin{pmatrix} \rho v_y \\ \rho v_x v_y \\ \rho v_y^2 + P \\ \rho v_y H \end{pmatrix}, \tag{3.51}$$

where the pressure P is defined by the chosen equation of state. These equation describe the conservation of mass momentum and energy. The polytropic ideal gas case is often used, where

$$P = \rho(\gamma - 1)\left(E - \frac{\mathbf{v} \cdot \mathbf{v}}{2}\right). \tag{3.52}$$

The other variable are defined as normal, where ρ is mass density, $\mathbf{v} = (v_x, v_y)$ are the velocity components, and E is the specific energy have, with enthalpy H given by

$$H = E + \frac{P}{\rho}. \tag{3.53}$$

The adiabatic speed of sound c_s is defined as

$$c_s = \sqrt{\frac{\gamma P}{\rho}}. \tag{3.54}$$

By definition these equations assume the fluid has no viscosity. As mentioned before, this assumption is applicable to many astrophysical scenarios, where viscosity appears to be negligible. Equivalent RD methods have been developed for the viscous Navier-Stokes equations (Abgrall & Santis, 2015).

Linearisation

The RD method is only applicable to linear sets of PDEs. The Euler equations are non-linear, as the Flux term is dependent on the state vector. The equations must be recast in a quasi-linear form. In order to produce the desired linearisation, a new parameter vector is defined, such that the Jacobian is only linearly dependent on the unknown of the PDE. The Roe parameter vector $\mathbf{Z}$ is suitable for this purpose, defined for the two dimensional case as

$$\mathbf{Z} = \begin{pmatrix} \sqrt{\rho} \\ \sqrt{\rho} v_x \\ \sqrt{\rho} v_y \\ \sqrt{\rho} H \end{pmatrix}. \tag{3.55}$$

Starting with the original inviscid 2D Euler equations, written now as a summation across dimensions, the initial equations look like

$$\frac{\partial \mathbf{Q}}{\partial t} + \sum_{j=1}^{2} \frac{\partial \mathbf{F}_j}{\partial x_j} = 0, \tag{3.56}$$

where $\mathbf{x} = (x, y)$ in 2D. This can be converted into a PDE of $\mathbf{Z}$ by introducing the partial derivatives with respect to the new vector, $\partial \mathbf{Q}/\partial \mathbf{Z}$ and $\partial \mathbf{F}_j/\partial \mathbf{Z}$, using the chain rule. The Euler equations now appear as

$$\frac{\partial \mathbf{Q}}{\partial \mathbf{Z}} \frac{\partial \mathbf{Z}}{\partial t} + \sum_{j=1}^{2} \frac{\partial \mathbf{F}_j}{\partial \mathbf{Z}} \frac{\partial \mathbf{Z}}{\partial x_j} = 0. \tag{3.57}$$

The state vector can be rewritten in terms of the new parameter vector as

$$\mathbf{Q} = \begin{pmatrix} Z_1^2 \\ Z_1 Z_2 \\ Z_1 Z_3 \\ \frac{Z_1 Z_4}{\gamma} + \frac{\gamma - 1}{2\gamma}(Z_2^2 + Z_3^2) \end{pmatrix}, \tag{3.58}$$

with the flux vectors as

$$\mathbf{F}_x = \begin{pmatrix} Z_1 Z_2 \\ \frac{\gamma-1}{\gamma} Z_1 Z_4 + \frac{\gamma+1}{2\gamma} Z_2^2 - \frac{\gamma-1}{2\gamma} Z_3^2 \\ Z_2 Z_3 \\ Z_2 Z_4 \end{pmatrix}, \tag{3.59}$$

and

$$\mathbf{F}_y = \begin{pmatrix} Z_1 Z_3 \\ Z_2 Z_3 \\ \frac{\gamma-1}{\gamma} Z_1 Z_4 + \frac{\gamma+1}{2\gamma} Z_3^2 - \frac{\gamma-1}{2\gamma} Z_2^2 \\ Z_3 Z_4 \end{pmatrix}. \tag{3.60}$$

From these forms, it is clear that the state and flux vectors depend quadratically on the Roe vector. The partial derivatives of these vectors, with respect to the Roe parameter vector, are therefore linearly dependent on the new parameters. This means that $\mathbf{Z}$ can be used to linearise the Euler equations using the form given in equation (3.57).

With the Roe parameter vector, the Jacobian in Equation (3.57) satisfies the requirements given in Section 3.2.1 for the Roe solver. The Jacobian in this case is the derivative of the flux with respect to $\mathbf{Z}$. The Jacobian at the boundary $\bar{\mathbf{A}}$, or in the element in the 2D case, is simply the Jacobian of the mean state of the vertices

$$\bar{\mathbf{A}}_x = \mathbf{A}_x(\bar{\mathbf{Z}}) = \mathbf{A}_x \left(\frac{\mathbf{Z}_1 + \mathbf{Z}_2 + \mathbf{Z}_3}{3} \right). \tag{3.61}$$

The element residual for this set of PDEs is now defined with respect to the new unknown $\mathbf{Z}$, such that Equation (3.32) is equivalent to

$$\phi^T = \frac{1}{2} \sum_{i=1}^{3} \mathbf{Z}_i \frac{\partial}{\partial \mathbf{Z}} \mathcal{F}(\bar{\mathbf{Z}}_i) \cdot \mathbf{n}_i. \tag{3.62}$$

However, in this formulation, the residual will calculate the update to the Roe parameter vector, rather than fluid state vector $\mathbf{Q}$. In order to make use of this Roe parameter vector to update the fluid state, the fluid state must be

reintroduced into the residual

$$\phi^T = \frac{1}{2}\sum_{i=1}^{3} \mathbf{Z}_i \frac{\partial}{\partial \mathbf{Z}} \mathcal{F}(\bar{\mathbf{Z}}) \frac{\partial}{\partial \mathbf{Q}} \mathbf{Z}(\bar{\mathbf{Z}}) \cdot \mathbf{n}_i \frac{\partial}{\partial \mathbf{Z}} \mathbf{Q}(\bar{\mathbf{Z}}). \tag{3.63}$$

These two equations are equal, but the second form allows us to write the residual for the original Euler equations, but calculated as a function of the mean state $\bar{\mathbf{Z}}$, rather than the fluid state. The residual is therefore given by

$$\phi^T = \sum_{j=1}^{3} \mathbf{K}_i(\bar{\mathbf{Z}}) \hat{\mathbf{Q}}_i(\bar{\mathbf{Z}}), \tag{3.64}$$

which is directly comparable to the generic discreet form of the element residual from Equation (3.33). The variables $\mathbf{K}_i$ and $\hat{\mathbf{Q}}_i$ of this specific discreet form are

$$\hat{\mathbf{Q}}_i(\bar{\mathbf{Z}}) = \frac{\partial}{\partial \mathbf{Z}} \mathbf{Q}(\bar{\mathbf{Z}}) \mathbf{Z}_i = \begin{pmatrix} 2\bar{Z}_1 Z_1 \\ \bar{Z}_2 Z_1 + \bar{Z}_1 Z_2 \\ \bar{Z}_3 Z_1 + \bar{Z}_1 Z_3 \\ \frac{1}{\gamma}\left(\bar{Z}_4 Z_1 + \gamma_1 \bar{Z}_2 Z_2 + \gamma_1 \bar{Z}_3 Z_3 + \bar{Z}_1 Z_4\right) \end{pmatrix}, \tag{3.65}$$

and

$$\mathbf{K}_i(\bar{\mathbf{Z}}) = \frac{1}{2}\frac{\partial}{\partial \mathbf{Z}} \mathcal{F}(\bar{\mathbf{Z}}) \frac{\partial}{\partial \mathbf{Q}} \mathbf{Z}(\bar{\mathbf{Z}}) \cdot \mathbf{n}_i = \frac{1}{2}\frac{\partial}{\partial \mathbf{Q}} \mathcal{F}(\bar{\mathbf{Z}}) \cdot \mathbf{n}_i = \frac{1}{2}\mathcal{A}(\bar{\mathbf{Z}}) \cdot \mathbf{n}_i. \tag{3.66}$$

This $\mathbf{K}_i$ matrix is sometimes referred to as the inflow matrix, as it can be used to encode the nature of the flow at each vertex since it projects the Jacobian onto the normal of the face opposite that vertex. The $\mathbf{K}_i$ matrix is now defined as the average of Jacobian matrices of the original form of the Euler equations, projected onto the edge normals of the element, where

$$\mathcal{A}(\bar{\mathbf{Z}}) = \left(\bar{\mathbf{A}}_x(\bar{\mathbf{Z}}), \bar{\mathbf{A}}_y(\bar{\mathbf{Z}})\right) = \left(\frac{\partial}{\partial \mathbf{Q}} \mathbf{F}_x(\bar{\mathbf{Z}}), \frac{\partial}{\partial \mathbf{Q}} \mathbf{F}_y(\bar{\mathbf{Z}})\right). \tag{3.67}$$

This may seem like we have come full circle, as the Jacobian is back to being defined as the differential with respect to the fluid state, but it is important to note that it is being calculated for the average Roe parameter, which produces a subtly different result to using the average fluid state. The introduction of $\hat{\mathbf{Q}}_i$, which has the same units as the fluid state but clearly differs from it in detail, and the fact that the Jacobian is evaluated at the average Roe state, together encode the effect of the linearisation.

To summarise, the RD method is only applicable to linear sets of PDEs, so a suitable linearisation of the Euler equations is required. Roe produced such a linearisation, initially for the Roe solver, but it is usable in this context as well. To calculate the residual for use in the update of the fluid state, the K matrix is based on the Jacobians evaluated at the average Roe parameter for the spatial element. This is combined with a variable that is analogous to the state variable, but defined with the chosen linearisation. These produce a consistent definition of the residual for that can be used to update the fluid state, without losing the effects of the linearisation and invalidating the scheme.

3.4.1 K-Matrix

The above definition of the inflow matrix gives all the physical information that is needed to calculate it for use in the solver, but the exact form must still be derived. In this case I do so for the 2D Euler equations. The partial differential of a four element vector by another four element vector produces a square matrix, where the elements are found individually by $\mathbf{A}_{jk} = \partial \mathbf{F}_j / \partial \mathbf{Q}_k$. The simplest way to perform this calculation is to first substitute in the Roe parameters to both flux and state. These substituted forms are given in Equations (3.58), (3.59), and (3.60). The inflow matrix is the arithmetic average of the projections of the mean Jacobian matrices onto the component, in that dimension, of the opposite edge normal. This is written as

$$\mathbf{K}_i = (\mathbf{A}_x n_{x,i} + \mathbf{A}_y n_{y,i})/2. \tag{3.68}$$

The evaluation of the derivative terms in each element of the matrix is straight forward. For example, the first row of the $\mathbf{A}_x$ matrix is found using

$$\begin{aligned}
A_{x,11} &= \frac{\partial F_{x,1}}{\partial Q_1} = \frac{\partial}{\partial (Z_1^2)} Z_1 Z_2 = 0, \\
A_{x,12} &= \frac{\partial F_{x,1}}{\partial Q_2} = \frac{\partial}{\partial (Z_1 Z_2)} Z_1 Z_2 = 1, \\
A_{x,13} &= \frac{\partial F_{x,1}}{\partial Q_3} = \frac{\partial}{\partial (Z_1 Z_3)} Z_1 Z_2 = 0, \\
A_{x,14} &= \frac{\partial F_{x,1}}{\partial Q_5} = 0.
\end{aligned}$$

The final form of the matrix is

$$\mathbf{K}_i = \frac{1}{2}\begin{pmatrix} 0 & n_x & n_y & 0 \\ \alpha n_x - v_x\omega & \omega - \gamma_2 v_x n_x & v_x n_y - \gamma_1 v_y n_x & \gamma_1 n_x \\ \alpha n_y - v_y\omega & v_y n_x - \gamma_1 v_x n_y & \omega - \gamma_2 v_y n_y & \gamma_1 n_y \\ (\alpha - H)\omega & H n_x - \gamma_1 v_x\omega & H n_y - \gamma_1 v_y\omega & \gamma\omega \end{pmatrix}, \tag{3.69}$$

where $\alpha = \gamma_1(v_x^2 + v_y^2)/2$ and $\omega = v_x n_x + v_y n_y$. The RD methods described above require the construction of $\mathbf{K}^+$ and $\mathbf{K}^-$, the matrices built from only the positive and negative eigenvalues respectively. To calculate these additional matrices, I need the decomposition of the inflow matrix, such that $\mathbf{K}_i = \mathbf{R}^{-1}\mathbf{\Lambda}\mathbf{R}$, where $\mathbf{\Lambda}$ is the diagonal matrix composed of the eigenvalues of $\mathbf{K}_i$. The right hand matrix $\mathbf{R}$ is made up of columns consisting of the eigenvectors of the inflow matrix. Solving the product of these matrices gives the inflow matrix in the form that I need. For the Euler equations, the eigenvalues are given by $\lambda_1 = \omega + c$, $\lambda_2 = \omega - c$, and $\lambda_3 = \lambda_4 = \omega$, while the K-matrix becomes (Paardekooper, 2017)

$$\begin{aligned}
K_{11} &= \frac{\alpha_c}{c}\lambda_{123} - \frac{\omega}{c}\lambda_{12} + \lambda_3, \\
K_{12} &= -\frac{\gamma_1 v_{xc}}{c}\lambda_{123} + \frac{n_x}{c}\lambda_{12}, \\
K_{13} &= -\frac{\gamma_1 v_{yc}}{c}\lambda_{123} + \frac{n_y}{c}\lambda_{12}, \\
K_{14} &= \frac{\gamma_1}{c^2}\lambda_{123},
\end{aligned}$$

$$\begin{aligned}
K_{21} &= (\alpha_c v_{xc} - \omega n_x)\lambda_{123} + (\alpha_c n_x - v_{xc}\omega)\lambda_{12}, \\
K_{22} &= (n_x^2 - \gamma_1 v_{xc}^2)\lambda_{123} - \gamma_2 v_{xc} n_x \lambda_{12} + \lambda_3, \\
K_{23} &= (n_x n_y - \gamma_1 v_{xc} v_{yc})\lambda_{123} + (v_{xc} n_y - \gamma_1 v_{yc} n_x)\lambda_{12}, \\
K_{24} &= \frac{\gamma_1 v_{xc}}{c}\lambda_{123} + \frac{\gamma_1 n_x}{c}\lambda_{12},
\end{aligned}$$

$$
\begin{aligned}
K_{31} &= (\alpha_c v_{yc} - \omega n_y)\lambda_{123} + (\alpha_c n_y - v_{yc}\omega)\lambda_{12}, \\
K_{32} &= (n_x n_y - \gamma_1 v_{xc} v_{yc})\lambda_{123} + (v_{yc} n_x - \gamma_1 v_{xc} n_y)\lambda_{12}, \\
K_{33} &= (n_y^2 - \gamma_1 v_{yc}^2)\lambda_{123} - \gamma_2 v_{yc} n_y \lambda_{12} + \lambda_3, \\
K_{34} &= \frac{\gamma_1 v_{yc}}{c}\lambda_{123} + \frac{\gamma_1 n_y}{c}\lambda_{12},
\end{aligned}
$$

$$
\begin{aligned}
K_{41} &= (\alpha_c H_c - \omega^2)\lambda_{123} + \omega(\alpha_c - H_c)\lambda_{12}, \\
K_{42} &= (\omega n_x - v_x - \alpha_c v_{xc})\lambda_{123} + (H_c n_x - \gamma_1 v_{xc}\omega)\lambda 12, \\
K_{43} &= (\omega n_y - v_y - \alpha_c v_{yc})\lambda_{123} + (H_c n_y - \gamma_1 v_{yc}\omega)\lambda 12, \\
K_{44} &= \frac{\gamma_1 H_c}{c}\lambda_{123} + \frac{\gamma_1 \omega}{c}\lambda_{12} + \lambda_3,
\end{aligned}
$$

where the terms with the form $X_c = X/c$. The factor of a half has been left out from each element for simplicity of notation. The new terms correspond to $\lambda_{123} = (\lambda_1 + \lambda_2 - 2\lambda_3)/2$ and $\lambda_{12} = (\lambda_1 - \lambda_2)/2$. The elements reduce to the original form of $\mathbf{K}_i$ when the eigenvalue are plugged in. For a given fluid state, substituting in only those eigenvalues that are either positive or negative for that state will produce $\mathbf{K}_i^+$ or $\mathbf{K}_i^-$. This specific form of the inflow matrix is only valid for the Euler equations. Other sets of equations, such as the viscous Navier-Stokes equations, produce a different matrix (Villedieu et al., 2011; Abgrall & Santis, 2015).

3.4.2 Time Step

All the ingredients required to calculate the residual, and distribute it in a conservative manner, have now been defined, as has the dual cell area $|S_i|$. To calculate the update using Equation (3.34), a mechanism to calculate a suitable time step is required. As has been discussed previously, the time step choice is not arbitrary. The CFL condition, that the numerical domain of dependence should enclose the physical domain of dependence, fundamentally limits the time steps that will produce a physically accurate result. Such a condition can be achieved

(Ricchiuto & Abgrall, 2010) by requiring that the time step Δt is limited by

$$\Delta t \leq \min_{i \in \mathcal{T}} \frac{2|S_i|}{\sum_{T|i \in T} l^T_{\max} \lambda^T_{\max}}, \tag{3.70}$$

where $l^T_{\max}$ is the longest edge of triangular element T, and $\lambda^T_{\max}$ is a measure of the maximum speed at which information can move across the element. This is done by setting

$$\lambda^T_{\max} = \max_{j \in T}(|\mathbf{v}_j| + c_j). \tag{3.71}$$

This is the maximum of the combination of the fluid speed and sound speed at the vertices of the triangle, which is equivalent to the maximum signal speed in that element. Together, the product of this length and signal speed, multiplied by a factor of a half, produce an estimate of the area per time of an imaginary triangle swept out by the material in this element. Summing up the contributions from all the element associated with a vertex i, and dividing the actual area associated with that vertex by this value, produces an estimate of the time it will take a signal to propagate across the dual cell. Keeping the time step below the minimum such value required by any vertex in the mesh $\mathcal{T}$ produces a limit within which the CFL condition will always be met. In practice, some fraction of this value will be usually be used, as an additional guarantee that the condition is not breached. This fraction typically varies between 0.1 and 0.5, depending on the complexity of the problem.

3.4.3 Summary of Equations

In the previous sections I have covered the theoretical background to the RD solvers development and extension. I have also described the precise formulation required to construct an RD solver for the 2D Euler equations. I will now briefly summarise the most important results, namely the final forms of the equations required to implement the solver in practice. The update to the fluid state $\mathbf{Q}_i$ at vertex i , from time step n to $n+1$, is given by

$$\mathbf{Q}^*_i = \mathbf{Q}^n_i - \frac{\Delta t}{|S_i|} \sum_{T|i \in T} \phi_i, \tag{3.72}$$

and

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^* - \frac{\Delta t}{|S_i|} \sum_{T|i\in T} \Phi_i, \tag{3.73}$$

where the time step is given by Equation 3.70, and the dual area is

$$|S_i| = \sum_{T|i\in T} \frac{1}{3}|T|. \tag{3.74}$$

The nodal residual is calculated from the element residual, based on the chosen scheme, and the element residual

$$\phi^T = \sum_{j=1}^{3} \mathbf{K}_i \hat{\mathbf{Q}}_i. \tag{3.75}$$

The exact form of $\mathbf{K}_i$ is given in Section 3.4.1. The equivalent total residual is calculated using the element residual

$$\Phi^T = \sum_{j\in T} m_{ij} \frac{\mathbf{Q}_j^* - \mathbf{Q}_j^n}{\Delta t} + \frac{1}{2}\left(\phi_i(\mathbf{Q}^*) + \phi_i(\mathbf{Q}^n)\right). \tag{3.76}$$

The mass matrix form varies with the scheme, as does the distribution itself. The linearised state $\hat{\mathbf{Q}}_i$ is calculated with

$$\hat{\mathbf{Q}}_i = \begin{pmatrix} 2\bar{Z}_1 Z_1 \\ \bar{Z}_2 Z_1 + \bar{Z}_1 Z_2 \\ \bar{Z}_3 Z_1 + \bar{Z}_1 Z_3 \\ \frac{1}{\gamma}\left(\bar{Z}_4 Z_1 + \gamma_1 \bar{Z}_2 Z_2 + \gamma_1 \bar{Z}_3 Z_3 + \bar{Z}_1 Z_4\right) \end{pmatrix}. \tag{3.77}$$

Together these equations describe all the key variables and functions needed to construct the 2D RD hydro-solver. Combining these with the distribution schemes described in the previous section, and making the choice of temporal order, one can produce a fully functioning RD solver.

3.5 Hydrodynamics Tests

I run a number of standard hydrodynamic tests to assess the abilities and characteristics of my implementation of the 2D residual distribution hydro-solver.

Many of these have well understood analytic solution to which the results can be compared, but others, often those with complex multi-dimensional flows, do not have explicitly defined solutions. Instead the tests generally look for the formation of expected structures. In these tests, I will be comparing results from the LDA, N, and B schemes, as well as the Roe solver. The comparisons will include the methods that are either first or second order accurate in time. When I refer to the first order LDA scheme (LDA1) the order refers only to the temporal accuracy.

3.5.1 1D Tests

There are a number of simple tests that measure the effectiveness of capturing fundamental flow, from simple advection of material to shocks fronts (discontinuities in density and pressure). I run these initial tests in pseudo one dimension, where the second dimension is still included in the calculation, but there is no variation in the fluid conditions in that direction.

Gaussian Pulse

I first test the capabilities of the code to handle simple advection by modelling the propagation of a Gaussian density pulse, in one dimension. The initial conditions consist of a uniform density background with $\rho = 10$, with an additional Gaussian density pulse centered at $x = 0.3$. The initial density distribution is

$$\rho(x) = \rho_p e^{-\frac{s^2}{w^2}} + \rho_0(1 - e^{-\frac{s^2}{w^2}}); \tag{3.78}$$

where $s = x_c - x$ is the distance from the centre of the Gaussian distribution, and w the width of the pulse. In this case, the peak density is set to $\rho_p = 50$. The x-velocity is the same at every position, with $v_x = 1$. In this case, the Gaussian distribution should simply move with the bulk velocity of the initial flow, maintaining its shape exactly. However, as shown in Figure 3.8, the Gaussian profile widens as it moves across the grid, with the peak density in the pulse gradually decreasing. The evolution is identical for all but the second order LDA scheme, where the effect is even more pronounced. The position of the maximum is in the expected place, and mass, energy and momentum are conserved to machine precision. These plots only show a column of vertices through the centre of the mesh, but there is no variation in the orthogonal direction, so this evolution

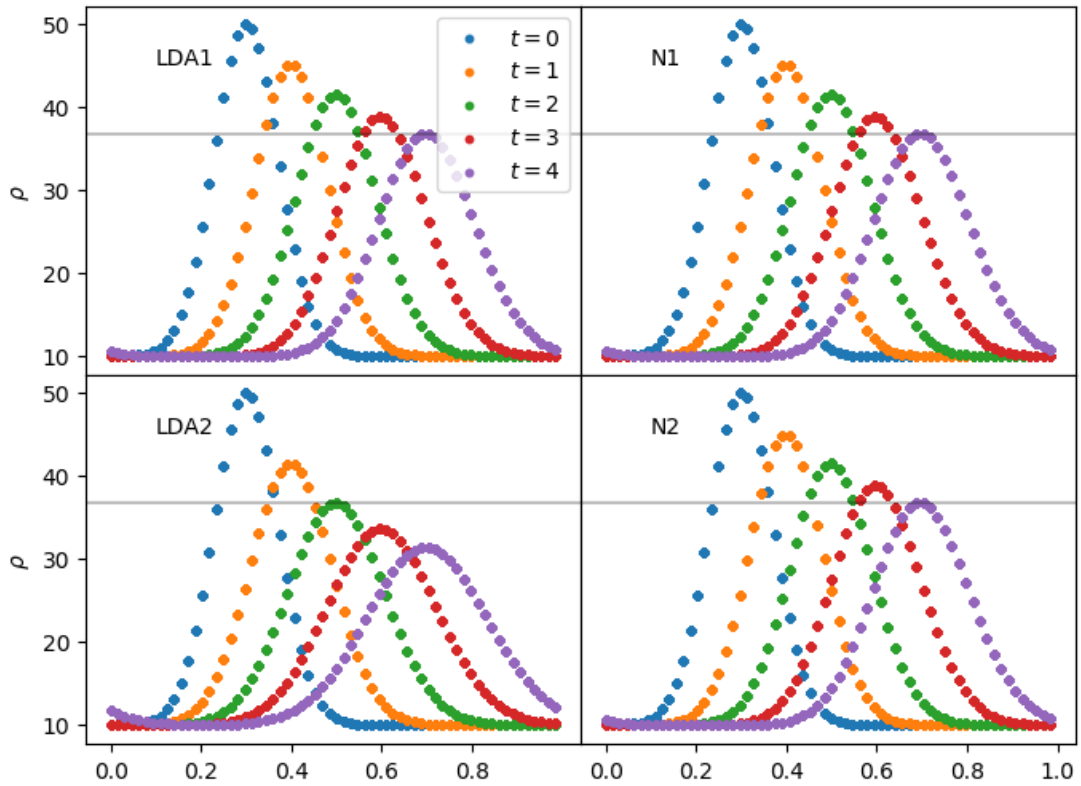

Figure 3.8 *Advection of a Gaussian density pulse moving in the positive x-direction for the first (top row) and second (second row) order LDA (left column) and N-scheme (right column) solvers. There is significant numerical diffusion, which is largely constant across the different methods, with the exception of the second order LDA solver.*

is found at all y-positions. The slumping effect is caused by numerical dissipation, which can be pictured in the following manner. One can take a uniform density box, with one cell that has a larger density, with the whole box moving at some velocity v, and consider only the first cell at the background level ahead of the high density cell. The CFL limitation on the time step size means that, over this time step, only a fraction of the high density cell will move to this new cell. The single high density cell is now spread across two cells. In this next time step more material will move from the original cell, possibly all the remaining over-density, but some of the over-density in the new cell will also move onwards. The increased diffusion in the LDA2 scheme is likely caused by the choice of mass matrix, since the currently used one having the highest diffusivity, but this requires further study to determine. The second order N scheme does not suffer from this additional numerical dissipation because the mass matrix falls out of the formulation.

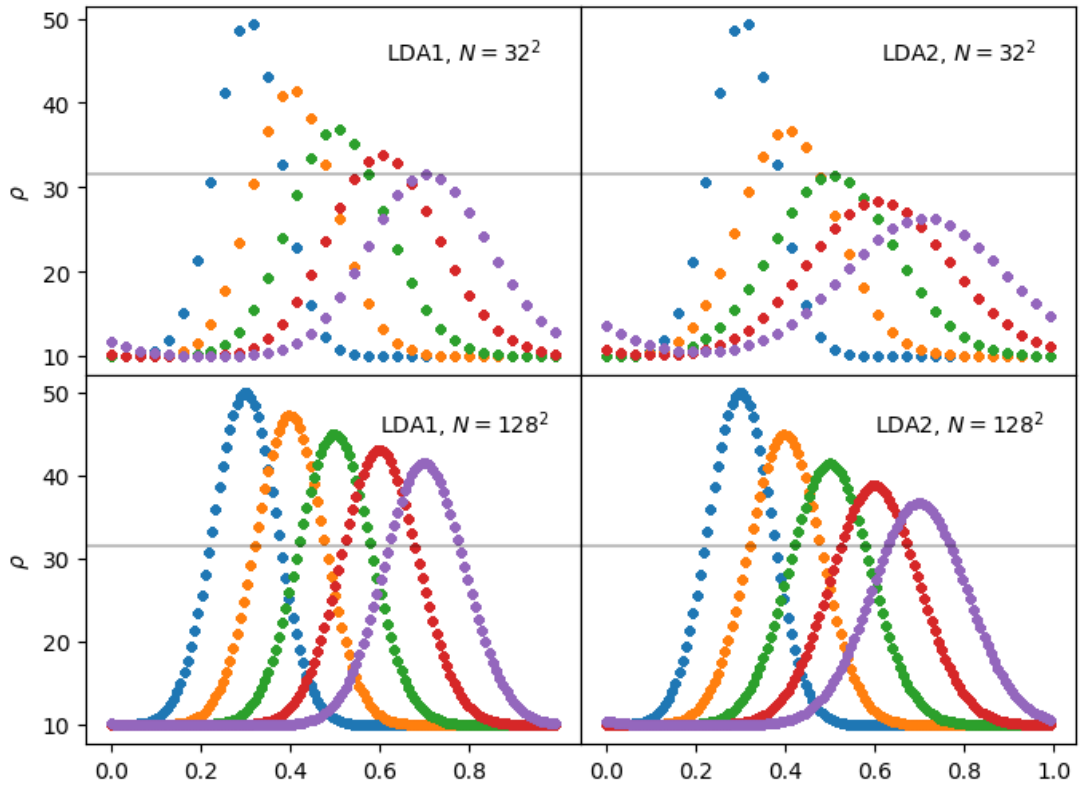

Figure 3.9 *Comparison of the advection of a Gaussian density pulse at different resolutions for $N = 32$ (top row) and $N = 128$ cells (bottom row). The left column shows results for the LDA1 solver, and the right for LDA2. The higher resolution shows significantly less numerical diffusion.*

Over time this effect compounds, leading to the slumping smoothing out of the density profile that is seen in these results. It is fundamentally caused by the discretisation of space, and is most pronounced for profiles that change a lot over a small number of cells. For this reason, greater resolution, achieved with more cells in the same physical space, reduces the dissipation of the profile. The same improvement has previously been observed for other solvers (Robertson et al., 2010). In Figure 3.9, I demonstrate this by comparing results for $N = 32^2$ (top row) and $N = 128^2$ (bottom row) cells, for both the LDA1 (left column) and LDA2 (right column) solvers. The resolution given here describe both the x and y directions, but the ICs only vary in the x-direction, so the effective resolutions along a given direction are $N = 32$ and $N = 128$. Both the higher resolution cases show significantly less numerical diffusion than their low resolution counterparts.

Sod Shock Tube

Shocks are a common feature of many astrophysical systems, found in cosmic filaments, gas falling into dark matter halos, the formation of stars, and the supernovae at the end of stellar lifetimes. The Sod shock tube (Sod, 1978) sets up a simple 1D shock, with a well defined solution to the evolution of the density, velocity, and pressure.

The initial conditions consist of two regions, each that fill half of the box. The velocity is zero everywhere. The left hand side of the tube has density $\rho_L = 1$ and pressure $p_L = 1$, with the right hand side of the tube at $\rho_R = 0.125$ and $p_R = 0.1$. These are run using adiabatic constant of $\gamma = 5/3$, and CFL coefficient of 0.4, meaning the time step is two fifths of the value strictly required by the time step condition. The adiabatic gas constant is the ratio of the specific heat capacity at constant pressure C_P, of the gas, to the specific heat capacity at constant volume C_V, or $\gamma = C_P/C_V$. Figure 3.10 shows the results from both the first order LDA and N schemes, compared to the exact solution for the Sod shock tube. I have also included results from the 1D Roe solver. The LDA results are shown as dots, with $N = 64$ vertices in the x-direction in blue, and $N = 128$ in red. The N-scheme results are represented by crosses, in green ($N = 64$) and cyan ($N = 128$). The black solid lines give the exact solution, while the dashed lines show the results for the first order Roe solver, for $N = 64$ and $N = 128$ respectively. Below each panel, I show the difference between the numerical and analytic solutions. The results from the Roe solver largely match the results from the RD solvers, with slightly less softening of the sharp profiles. This difference is caused by the Roe solver being applied to truly 1D grid, whereas the RD solvers are run on a pseudo 1D mesh, where there is no variation in the y-direction. There is a small amount of numerical dissipation from material flowing in the y-direction, even though the resultant variation in that dimension is zero.

There is clear evidence of spurious oscillations at $x = 0.5$ for the LDA1 solver, as predicted, but it decreases with better spatial resolution. The N-scheme does not show these structures, as expected, but the profiles at the transitions between solution phases do show smoothing. This is present in both LDA and N-scheme solvers, as well as in the Roe solver solution. The smoothing is improved in all cases by the increase in resolution, and is caused by the numerical diffusion discussed in with the Gaussian pulse test. Further increasing the resolution could improve the sharpness of these profiles. The systematic differences discussed here

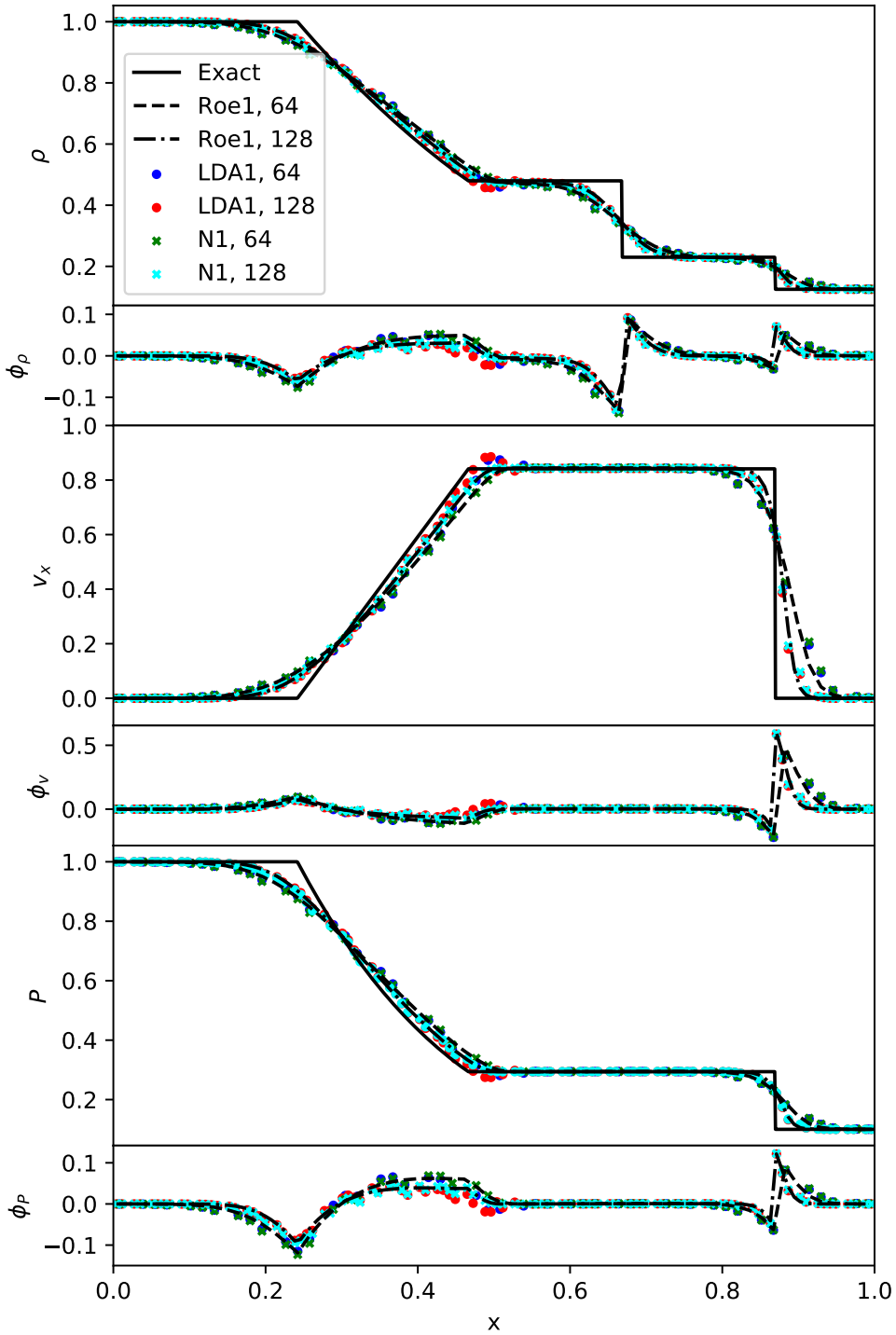

Figure 3.10 *Sod shock tube for the first order LDA and N schemes (in pseudo 1D), and the Roe solver (in true 1D). $N = 64$ and $N = 128$ vertices in the x-direction. Dots show LDA results (blue for $N = 64$ and red for $N = 128$), and crosses N-scheme results (green $N = 64$ and cyan $N = 128$). Solid black line is the exact solution. From top to bottom, major panels show density, x-velocity and pressure, with the minor panels showing the difference between the numerical and exact solutions $\phi_X = X_{\text{num}} - X_{\text{exe}}$, for each property.*

do not change significantly with time.

In Figure 3.11, I show a comparison of results from the first and second order solvers. Even at this low resolution, the LDA2 solver does not show the spurious oscillations found with LDA1, demonstrating a key advantage of the second order formulation. Once again LDA2 shows more diffusion, as discussed with the Gaussian pulse test. This manifests as the smoother transitions at the region interfaces. The N1 and N2 results are essentially identical, showing no significant improvement in recovering the profiles at the interfaces. Overall, the RD solvers can reproduce the fundamentals of the physical shock, with only small spurious features in schemes known to not handle shocks well. Therefore, in more complex situations, it should be able to handle shocks well.

3.5.2 2D Tests

The tests discussed so far demonstrate how well it recovers solutions that are well known for 1D flows, where the exact solution can be known. The key difference that this RD approach has, compared to the standard mesh based methods of most approaches currently used in the field, is the truly multi-dimensional way in which the equations are solved. With the 2D tests discussed in this section, it is possible to demonstrate the ability of this solver to handle complex multi-dimensional flows.

Kelvin Helmholtz Instability

Kelvin-Helmholtz instabilities form at the interface between shear flows. These occur in terrestrial and astrophysical contexts, such as between cloud layers in our atmosphere, or, at the other end of the size scale, in jets from AGN. This test sets up such a scenario, with two regions of gas moving alongside each other in opposite directions. The periodic box of side length $L_x = L_y = 1$ has a central region, with boundaries $0.25 < y < 0.75$, with density $\rho_0 = 2$, moving in the x-direction with velocity $v_{x0} = 1$. The outer region is moving with velocity $v_{x1} = -1$, and has density $\rho_1 = 1$. The difference in density is not important for the instability itself, but is useful in observing the mixing of the two flow. To generate the instability in a systematic way, a very small transverse velocity is introduced, with sinusoidal variation in the direction of the flow. The instability is expected to develop into a spiral like structure, as the two flow mix at the

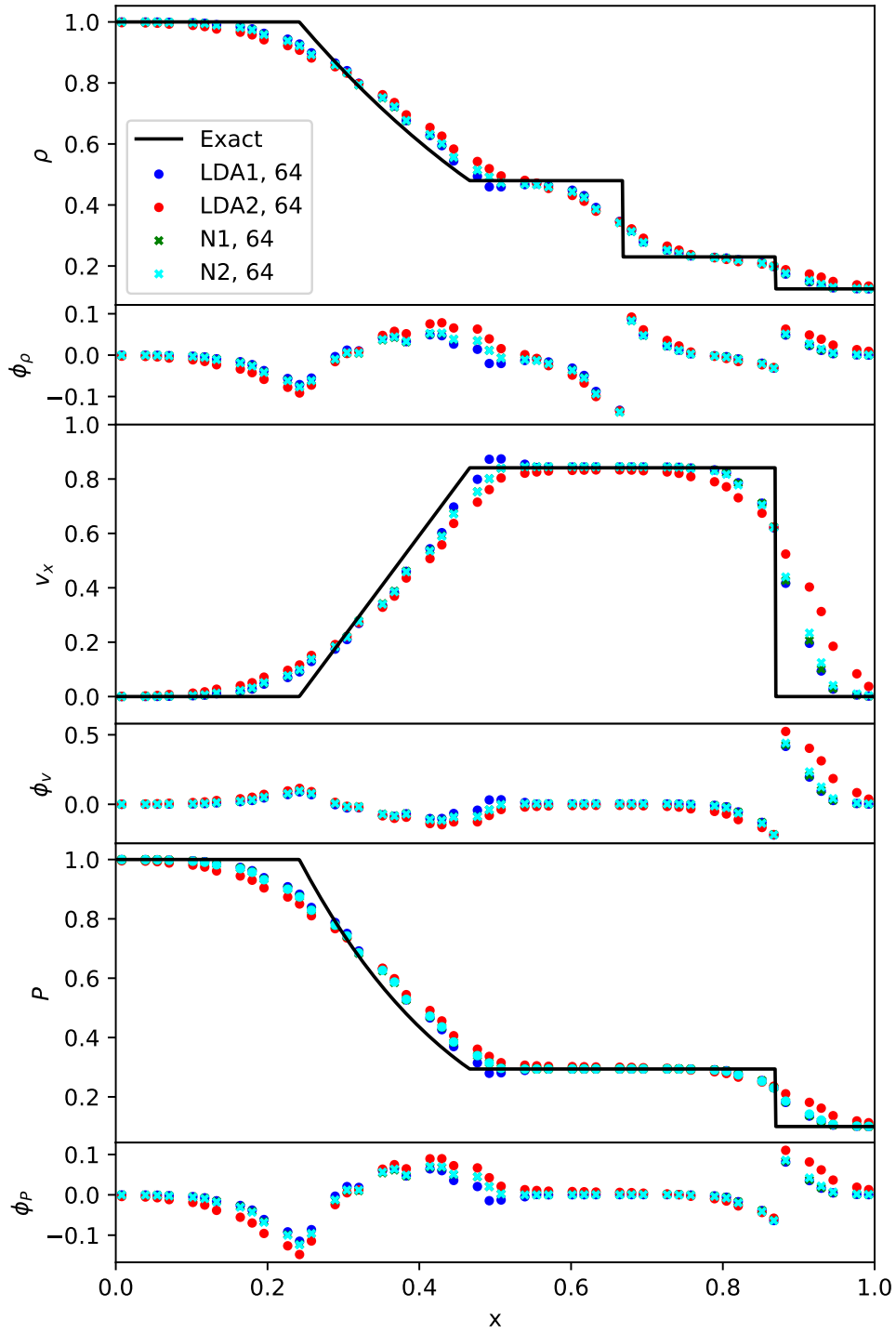

Figure 3.11 *Results from the the Sod shock tube for the second order LDA and N schemes, with comparison to the first order counterparts.*

boundary. The main quantitative test of the results is to compare the growth of transverse kinetic energy. This can only be done while the instability remains linear. Other than that, qualitative comparisons are limited to the sharpness of the boundary between density components, as a measure again of numerical diffusion.

The results of evolving this shear flow setup for the various solvers are shown in Figure 3.12 for $N = 64 \times 64$, with LDA1 (top left), LDA2 (bottom left), N1 (top right), and N2 (bottom right). The expected structures form very clearly in the LDA cases, with the winding structure recovered down to a few cells across. The N scheme results show much less structure, with only the broad curling of the flow being recovered. This is caused by the numerical diffusion of the scheme, which the second order formulation does not significantly change. The LDA results can resolve the structure in the greatest detail, and so are favourable for problems that involve complicated flows, without any shocks. Running the same test with a blended scheme largely reproduces the LDA results, as the blending favours those residuals in these conditions.

The total kinetic transverse energy, found simply by summing the y-direction kinetic energy of each vertex, should grow exponentially (McNally et al., 2012). Figure 3.13, which plots the total transverse kinetic energy K_{tot} for different resolutions, shows this growth between $t \approx 0.3$ and $t \approx 1.2$. The exponential growth appears linear in this log scale. Before this time, the kinetic energy is dominated by the initial sinusoidal perturbation. After this point, the growth becomes non-linear, as the initial turnover of the instability forms the more complex spiral structures. This time is earlier for higher resolutions, as finer structures are recovered.

An extensive resolution test of the LDA1 solver is shown in Figure 3.14, with resolutions $N = 32 \times 32$ (top left), $N = 64 \times 64$ (top right), $N = 96 \times 96$ (bottom left), and $N = 128 \times 128$ (bottom right). The results appear consistent across the increasing resolutions, with the instability developing in the same place in each case. Even in the lowest resolution case, the spiral structure forms. The density contrast is present until the spiral is less than 2-3 vertices across, when the structure diffuses into the background. The higher resolution cases can resolve more detail within the instability, but even at low resolution the numerical solution retains a strong level of fine structure. However, in the $N = 96 \times 96$ and $N = 128 \times 128$ cases, secondary instabilities can be seen to develop. These develop from the small variation in the boundary, created by the positions of the vertices.

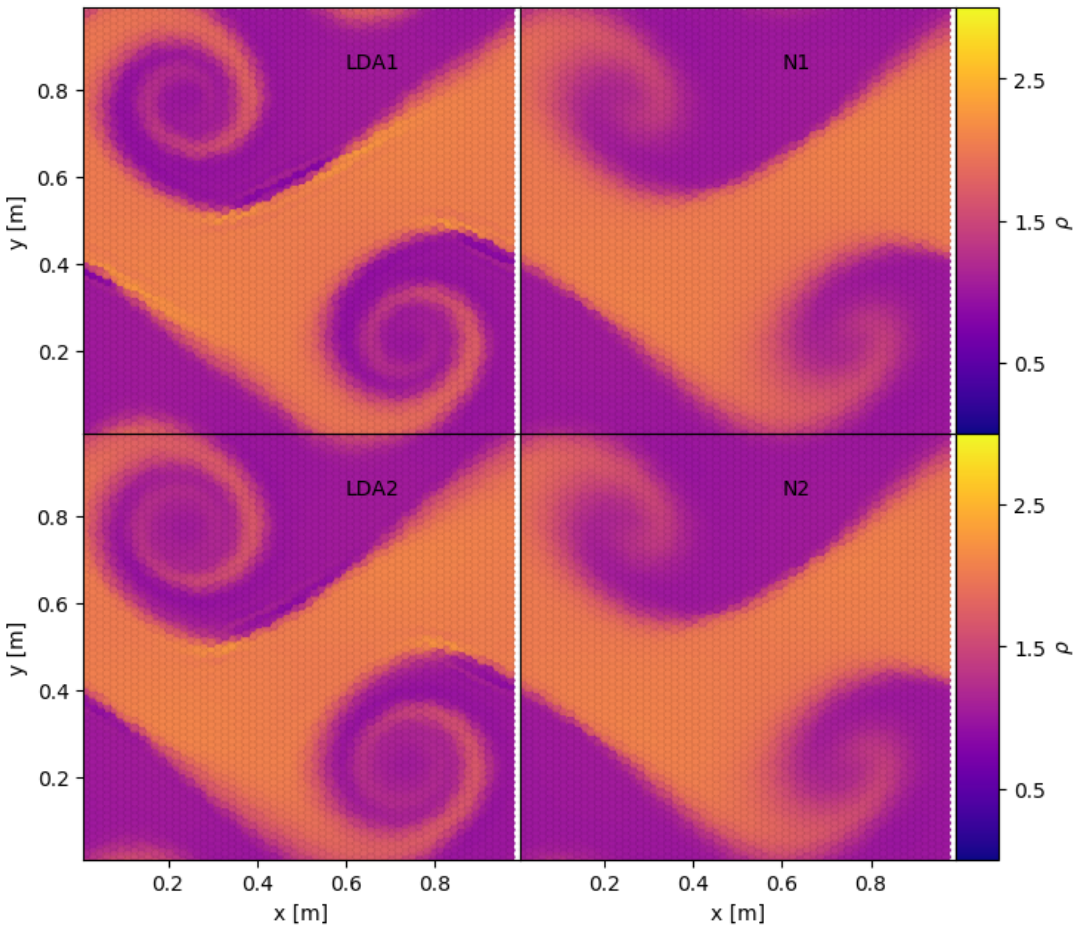

Figure 3.12 *Kelvin-Helmholtz instability for the first (top row) and second (bottom row) order LDA (left column) and N-scheme (right column) solvers. The color scale shows variation in density.*

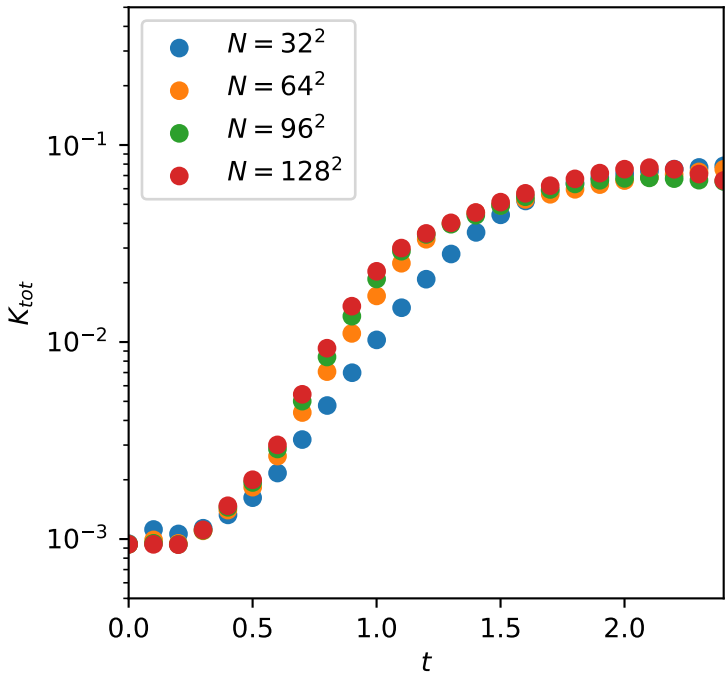

Figure 3.13 *Total transverse kinetic energy for resolutions $N = 32$ (blue), $N = 64$ (orange), $N = 96$ (green), and $N = 128$ (red). The growth converges with resolution, while the non-linearity sets in earlier for higher resolution.*

The variations create high frequency instability modes, which grow in these higher resolution cases. In the low resolution cases, the same variations are present, but the higher numerical diffusion means that they dissipate into the background flow.

The LDA results across the board show erroneous stripe like structure forming at the knee of the instability. These striped spurious oscillations in the solution similar to those seen in Sod test. They become more pronounced the higher the resolution. In the test cases shown so far, the structured mesh has been used to make the boundary between flows as clean as possible. A random distribution of vertices would lead to a ragged edge, which has the potential to trigger instabilities at shorter wavelength modes, as discussed above, which make analysis of the results more difficult. A common technique to avoid this problem is to smooth the boundary layer with an exponential density and velocity profile. This smooths out any ragged edge of the two flows, and ensures the stimulated instability mode will dominate the evolution.

The smoothed boundary is achieved by defining two new functions, $f(\theta)$ and $g(\theta)$.

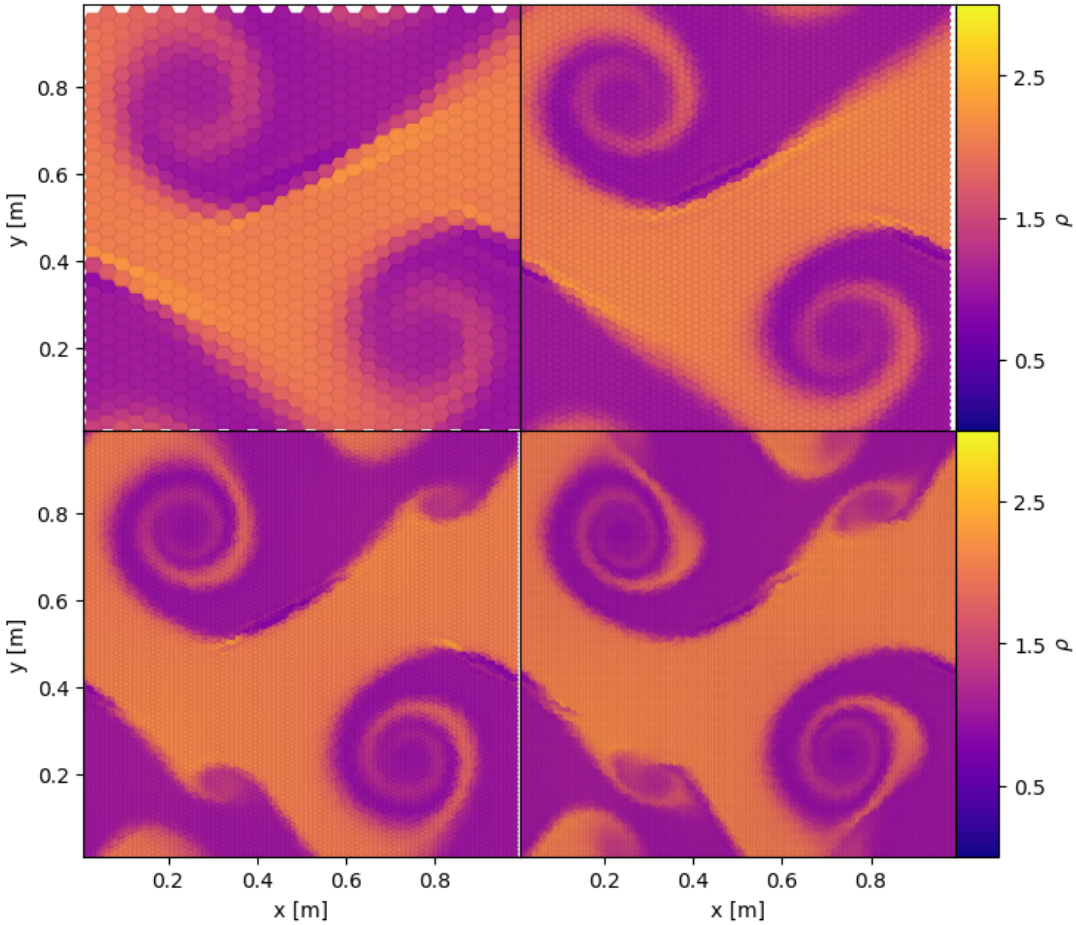

Figure 3.14 *Kelvin-Helmholtz instability for the first order LDA solver, for different spatial resolutions, with* $N = 32 \times 32$ *(top left),* $N = 64 \times 64$ *(top right),* $N = 96 \times 96$ *(bottom left), and* $N = 128 \times 128$ *(bottom right). The color scale shows variation in density.*

The first of these has the form

$$f(\theta) = e^{-1/\theta}, \tag{3.79}$$

where theta is limited to $0 \leq \theta \leq 1$. The second function is given by

$$g(\theta) = \frac{f(\theta)}{f(\theta) + f(1-\theta)}. \tag{3.80}$$

Together these are used to smooth the density and velocity boundary between the sheer flows by setting the density and velocity respectively as

$$\rho(y) = (\rho_0 - \rho_1) g\left(\frac{1}{2} + \frac{4y-1}{4d}\right) g\left(\frac{1}{2} - \frac{4y-3}{4d}\right) + \rho_1 \tag{3.81}$$

and

$$v_x(y) = 2v_{x0} g\left(\frac{1}{2} + \frac{4y-1}{4d}\right) g\left(\frac{1}{2} - \frac{4y-3}{4d}\right) - v_{x0}. \tag{3.82}$$

The width of the boundary layer is dictated by d. Results using this smoothed boundary are shown in Figure 3.15. A clear effect of the smoothed boundary is the equivalent smoothing of the boundary between the spiral structures. As the width of the boundary layer d is decreased, the results converge on the original solution. The higher frequency modes, which grow in the higher resolution cases above, are completely absent with these smooth ICs. The spurious oscillations at the knee of the instability are also suppressed, thanks again to the smooth transition removing the density discontinuity.

So far I have shown how well the different RD schemes perform with the KH test. In Figure 3.16 I show a comparison to the Meshless Finite Mass (MFM) solver described in Section 2.3.2. These all show density contours based on the underlying distribution of vertices, or the free moving particles in the MFM cases, instead of the previous Voronoi tessellations. While the tessellations better represent the discretisation used by the RD solvers, they do not work as well for the particle based results. I have switched to this approach to provide the best comparison between the results. The left hand column shows results from the LDA1 solver using $N = 64^2$ and $N = 128^2$ vertices. The right hand column shows results from the MFM solver. These runs were produced by a collaborator using identical ICs, except for the third dimension. The results from the non-RD solvers use a 3D box, with five layers of particles. Each layer has the same setup in the

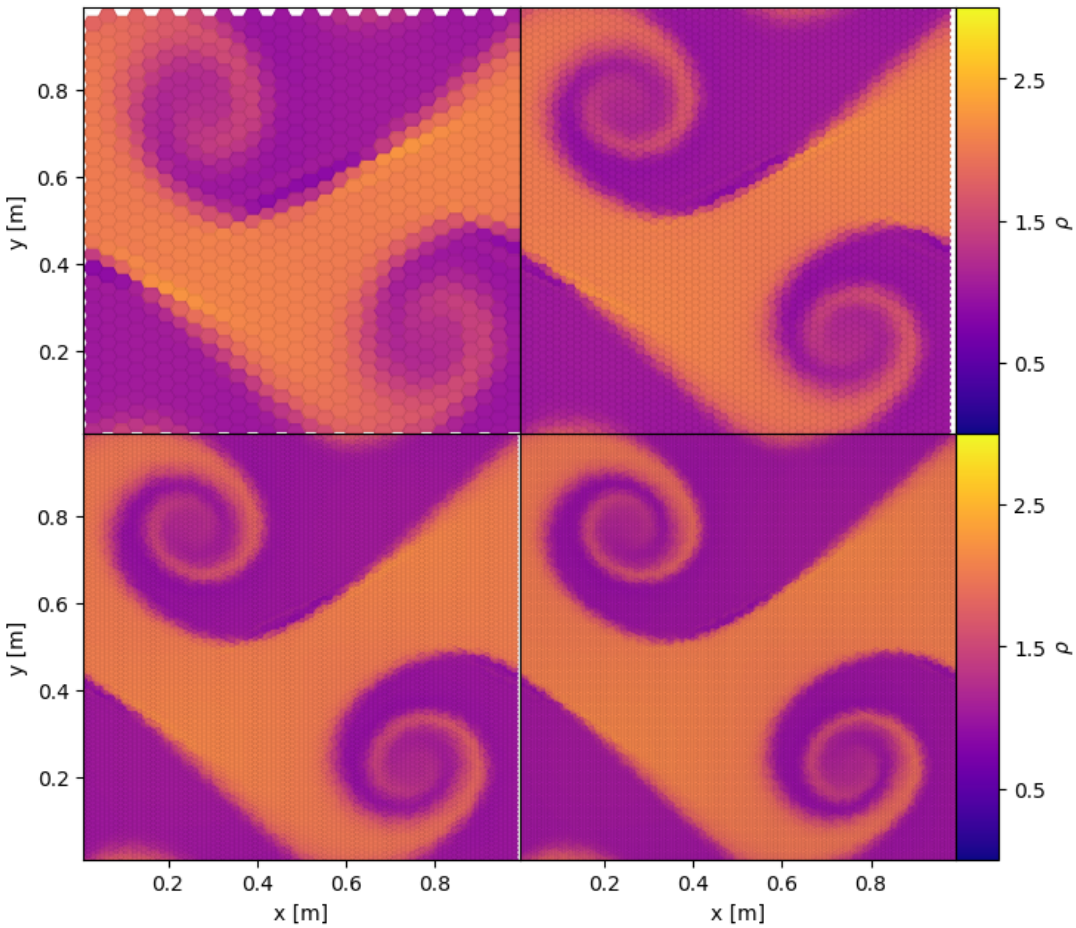

Figure 3.15 *KH instability from smooth initial conditions at different resolutions. These are $N = 32 \times 32$ (top left), $N = 64 \times 64$ (top right), $N = 96 \times 96$ (bottom left), and $N = 128 \times 128$ (bottom right). The higher mode instabilities are completely gone, and the stripe structures are significantly reduced.*

x and y dimensions, so effectively recreate a 2D problem, since the conditions do not vary in the z-direction. During the tests, very few particles moved between layers, so these setups still provide an effective comparison to the truly 2D RD solver results.

The MFM results show significant disruption of the spiral instability structures by secondary modes. The MFM does not experience the numerical dissipation found in the RD methods, due to its Lagrangian nature, and so any small variation in the initial state will grow into these shorter wavelength instabilities. Small differences can also be introduced by the method itself, since it is a numerical approximation, and these can also contribute secondary instability modes. These dissipate in the LDA1 runs, due to the numerical dissipation. This dissipation is an important property, of the method, to keep in mind. The RD solver results do not produce these undesired modes in this case, but they would also suppress such modes even if they were introduced on purpose, so the choice of which solver handles the situation 'better' depends on the specific conditions that one wants to model. For this test, the RD solver arguably resolves more of the fine structure of the main instability, even though the Lagrangian MFM solver naturally refines resolution. The strength of the multi-dimensionality of the RD solver is therefore clearly on display in these results. It is able to resolve the spiral structure down to the point where each density component is only a 3-5 elements across.

Sedov Blast

The Sedov blast (Sedov, 1959) replicates an explosion in a zero pressure environment. It reproduces conditions similar to the explosion from a highly idealised supernova. This is achieved with a static, uniform density and pressure, background medium at all positions. The explosion is triggered by injecting a large amount of energy into the centre of the domain. In this case, I do this by setting the pressure to an extreme value. Ideally the pressure would only be injected into one vertex, to replicate a point explosion, but when this is done the initial propagation of the explosion can only follow the connections to the nearest vertices, leading to highly asymmetric wave. To avoid this, a circular region is defined, within which the energy is injected. The region is large enough that the outward flow is approximately radial, but small enough that the analytic solution is still applicable.

The explosion is expected to create a spherically expanding wave with a shock at

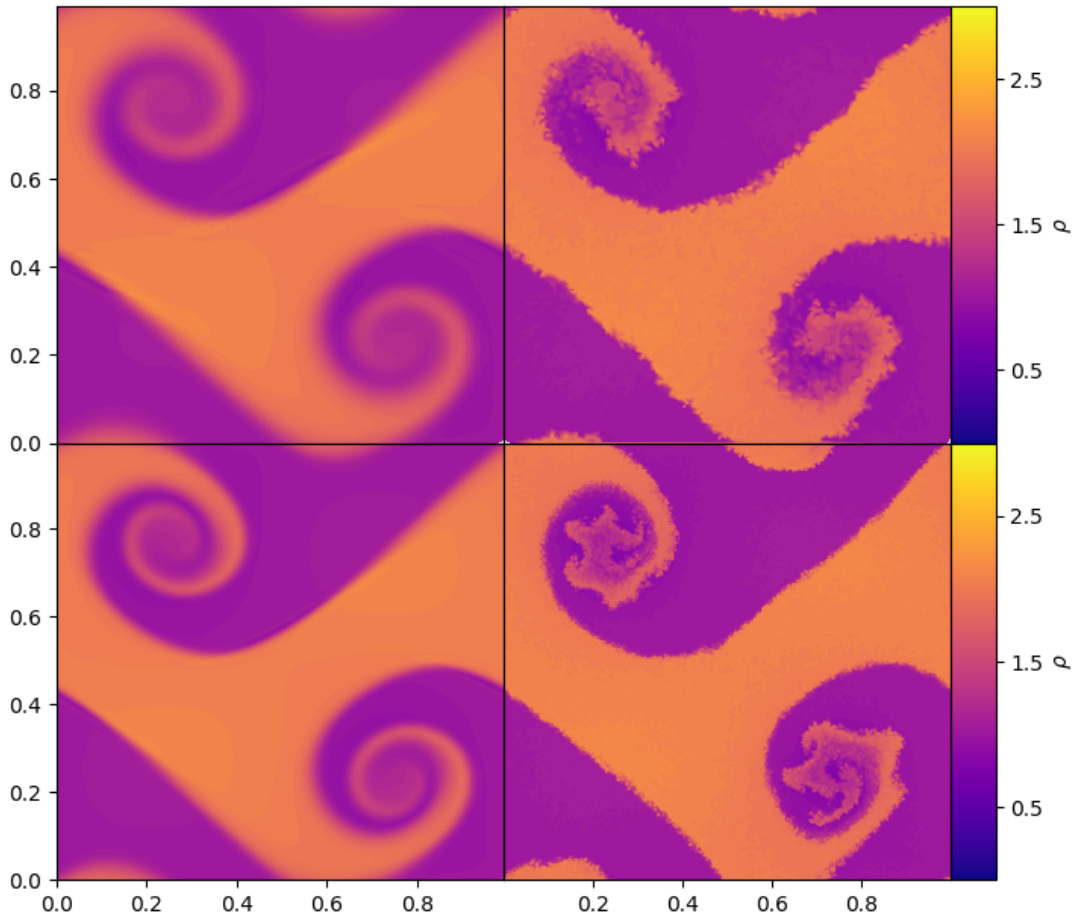

Figure 3.16 *KH instability from the smooth initial conditions, comparing the LDA1 solver (left column), to the MFM solver (right column). Top row shows results for $N = 64^2$, bottom row for $N = 128^2$.*

the expansion front. The velocity at which the front move is set by the density of the background medium and the initial energy of the explosion, with the radius of the blast wave given by

$$r(t) = \lambda \left(\frac{E t^2}{\rho_0} \right)^{\frac{1}{5}} . \tag{3.83}$$

Here I denote the total energy of the explosion by E, and the background density with ρ_0. The coefficient λ depends on adiabatic gas constant γ, at $\lambda \approx 1.12$ for the $\gamma = 5/3$ used here. Behind the shock front is an exponential density profile, falling to close to zero at the centre of the explosion.

In Figure 3.17, I show a comparison of the propagation of the explosion for three spatial resolutions, $N = 64^2$ (left column), $N = 128^2$ (middle column), and $N = 256^2$ (right column), for the N1 solver. The results from the N2 scheme are effectively identical in this comparison, so are not shown. The LDA1 and LDA2 schemes do not produce stable results, due to their poor handling of strong discontinuities. The blended schemes heavily favour the N-schemes in this test, and so produce results identical those shown for N1. The structure of the blast wave is recovered at all resolutions, with the dense wave sweeping up material as it moves radially outwards. As resolution is increased, the basic structure does not change, but the density profile does narrow. The narrower profile also shows a higher peak density. This is shown in more detail in Figure 3.18. The key result from this comparison, however, is the consistency between resolutions. The blast wave is in essentially the same position at a given time, though the extent of the profile differs.

The solution is azimuthaly symmetric, so I can simply compare the radial density profile to the predicted solution. This is shown in Figure 3.18, with the analytic prediction (solid black line), and results for the different resolutions as dash-dot lines, with $N = 64 \times 64$ (blue), $N = 128 \times 128$ (orange), and $N = 256 \times 256$ (green). The basic structure of the solution is recovered at all resolutions, but the profile of the front is significantly smoothed at $N = 64 \times 64$. At that resolution the peak is at the correct position, but at higher resolutions the peak lags behind the predicted position by a small amount. The density peak is higher at higher resolutions, and the width of the profile approaches the predicted shape. The profile behind the front roughly follows the predicted profile in all cases, but with a very small offset. The total energy of the initial injection is exactly equal in all cases, but the exact shape of the region it is inserted into will be slightly different.

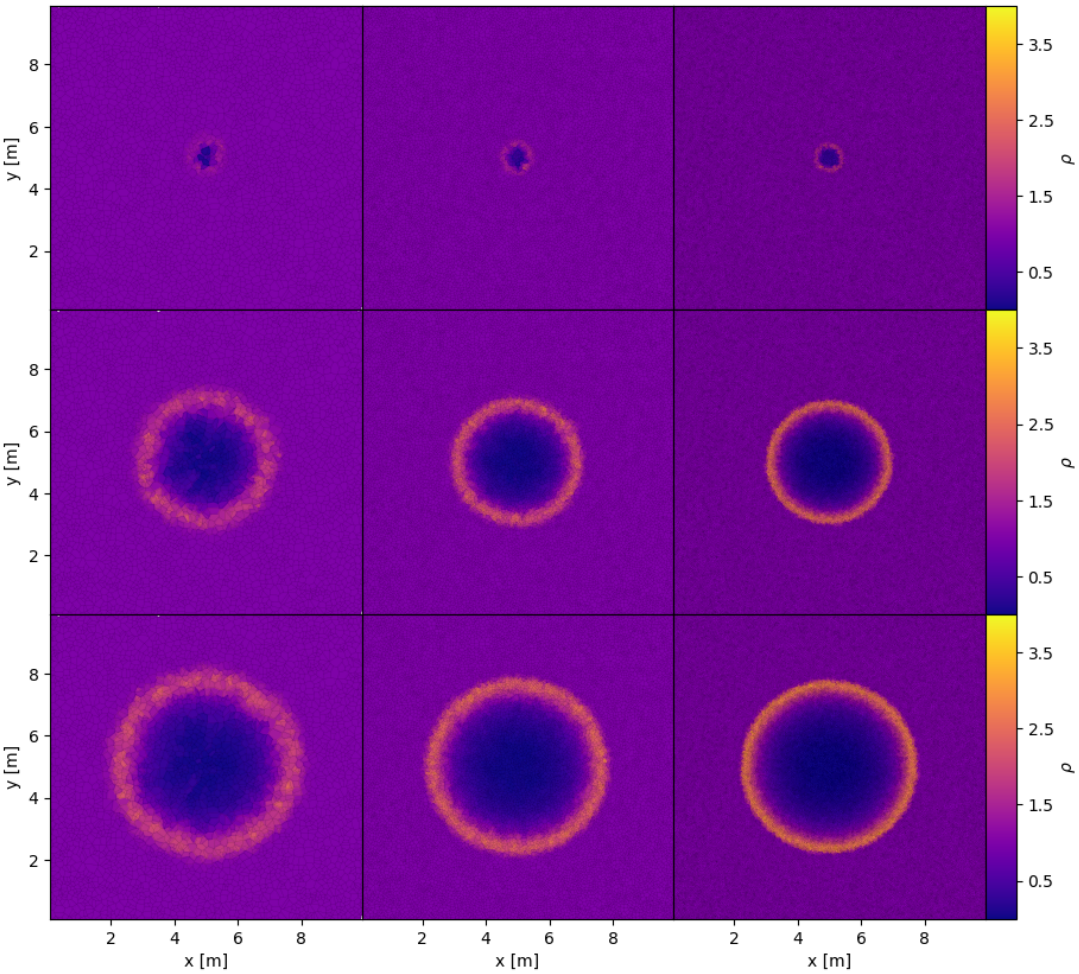

Figure 3.17 *Sedov blast density results with increasing resolution, $N = 64 \times 64$ (left column), $N = 128^2$ (middle column), and $N = 256^2$ (right column). Time increase downwards, with snapshots at $t = 0.0002$, $t = 0.005$, and $t = 0.01$.*

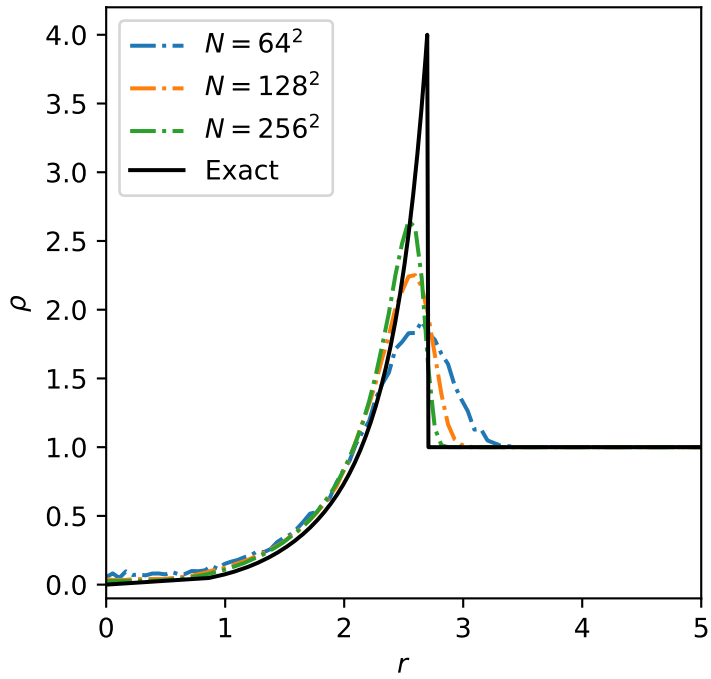

Figure 3.18 *Radial density profile, compared to the analytic prediction (solid black line). Results are shown for the three resolutions, $N = 64^2$ (blue), $N = 128^2$ (orange), and $N = 256^2$ (green).*

This, and the fact that the initial region is not a point, as the solution assumes, likely cause the differences between resolutions.

I compare the RD results to the MFM solver, and the related Meshless Finite Volume (MFV) solver (Hopkins, 2015), using tests run by a collaborator. In Figure 3.19, I show results for the different solvers at three resolutions. Overall, the MFM and MFV solvers reproduce a sharper discontinuity at the blast front, particularly at lower resolutions, with slightly more variation within the density structures. Both of these features can be explained by the Lagrangian nature of these solvers. The motion of the underlying particles allows for natural resolution adaptation, such that the high density shock front is resolved by a greater number of elements. At the lowest resolution, the Lagrangian solvers show more variation from the spherical shape of the blast, while the N1 solver produces a regular shape, even with a relatively small number of elements.

Figure 3.20 shows a comparison of the radial profiles for the different solvers. From left to right, I show results for $N = 64^2$, $N = 128^2$, and $N = 256^2$ vertices/particles. At low resolution, we see that that the MFM results lag behind the predicted solution, while the MFV and N1 results more closely replicate the expected result.

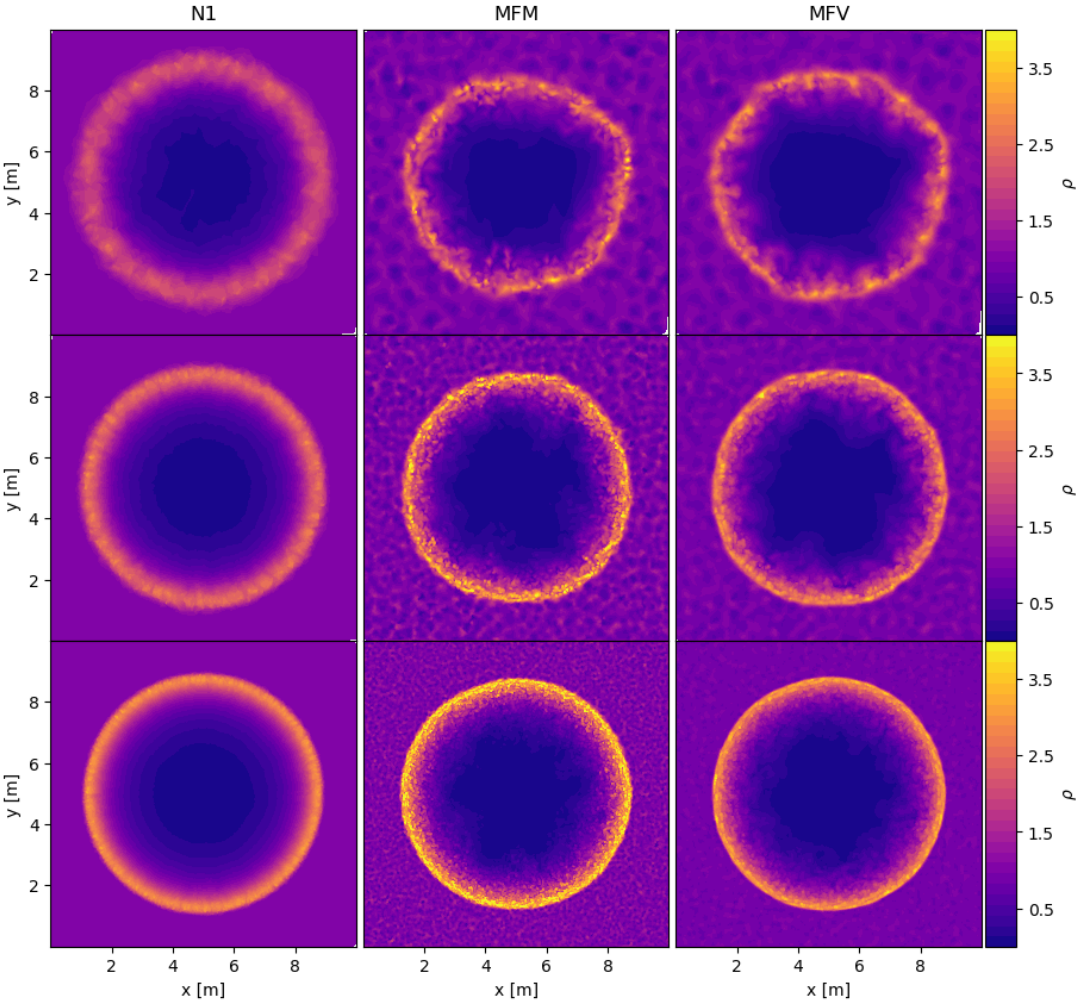

Figure 3.19 *Sedov blast for the N1, MFM and MFV solvers, using* $N = 64^2$ *(top row),* $N = 128^2$ *(middle row), and* $N = 256^2$ *(bottom row) vertices/particles.*

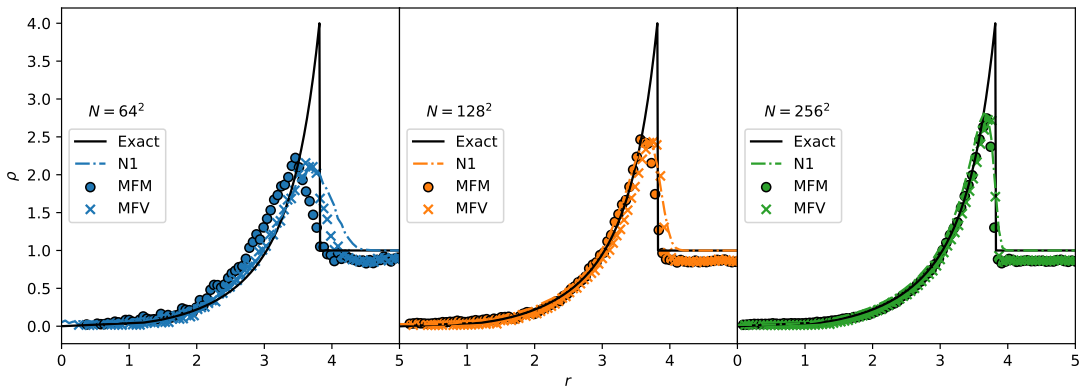

Figure 3.20 *Radial density profile, compared to the analytic prediction (solid black line). Results are shown for the three resolutions, $N = 64^2$ (left column), $N = 128^2$ (middle column), and $N = 256^2$ (right column).*

As mentioned above, the MFM and MFV solvers produce narrower peak profiles. This is most clearly seen at the higher resolutions, where the MFM results, in particular, produce a very sharp blast front profile. At this high resolution, the results shown only minimal differences between methods. While the RD solver produces a slightly more smeared out profile, for the reasons discussed above, it still produces results that can compete with state-of-the-art solvers such as these. With further optimisation, and the potential for conversion to a moving mesh method, this RD approach could become even more effective.

Noh Problem

The Noh problem (Noh, 1987; Paardekooper, 2017) tests the ability of a solver to model the conversion of kinetic energy into internal energy. This is similar to the Sedov test, which features the conversion of internal energy to kinetic energy in the injection of energy through pressure. It consists of a uniform density box, where the initial velocity at every position points radially inwards towards the centre of the box. The ideal initial conditions have zero pressure throughout the box. I use a cube box of side length $l = 2$, with initial density $\rho_0 = 1$. When the pressure is exactly zero, the K-matrix becomes singular, and so cannot be inverted. Instead, I initialise the problem with a negligible, but non-zero, pressure of $P = 10^{-6}$. The adiabatic gas constant is taken to be $\gamma = 5/3$. With these initial conditions, the cylindrically symmetric exact solution (Paardekooper, 2017) is as follows. At time t after the start of the problem, the density at radius r from the

centre of the box is

$$\rho(r,t) = \begin{cases} 16 & \text{if } r < t/3 \\ 1 + t/r & \text{if } r \geq t/3 \end{cases}, \tag{3.84}$$

the velocity magnitude is

$$|v(r,t)| = \begin{cases} 0 & \text{if } r < t/3 \\ 1 & \text{if } r \geq t/3 \end{cases}, \tag{3.85}$$

and the pressure is

$$P(r,t) = \begin{cases} 16/3 & \text{if } r < t/3 \\ 0 & \text{if } r \geq t/3 \end{cases}. \tag{3.86}$$

These equations describe the build up of a uniform density cylinder in pressure equilibrium. The cylinder expands as material flows towards the centre, with a shock at its surface. Material outside the cylinder continues to flow inwards at its initial rate.

I show the difference in the density and pressure results with increasing resolution, for the first order N scheme, in Figure 3.21, with $N = 32^2$, $N = 64^2$, $N = 128^2$ vertices, at $t = 0.8$. I only show the N1 solver here, because the LDA solver struggles with the extreme conditions at the very centre of the box, when the shock first forms. The N2 results are not significantly different, for this test. The density inside the shock increases with resolution, suggesting this density is somehow dependent on the formation formation of the shock at the centre of the box. At lower resolution, this initial radius will be larger, since the elements are larger and the central shock will form over a wider physical area. The shock front at the surface sharpens with the increasing resolution. The shock is resolved by a approximately 3-5 mesh vertices, represented here by their dual Voronoi cells. There is some variation in density within the cylinder, particularly in the centre, where there is a small low density cavity. This region gets smaller with increased resolution. This is likely an artifact of the finite resolution. When the initial flow builds up material in the innermost region, material can only flow in a small number of directions, limited by the exact structure of the mesh. The inward radial flow is therefore not well resolved, leading to this less exact solution. The position of the shock is not well defined when the circular shape is only resolved by a few vertices. The pressure, on the other hand, is significantly more uniform

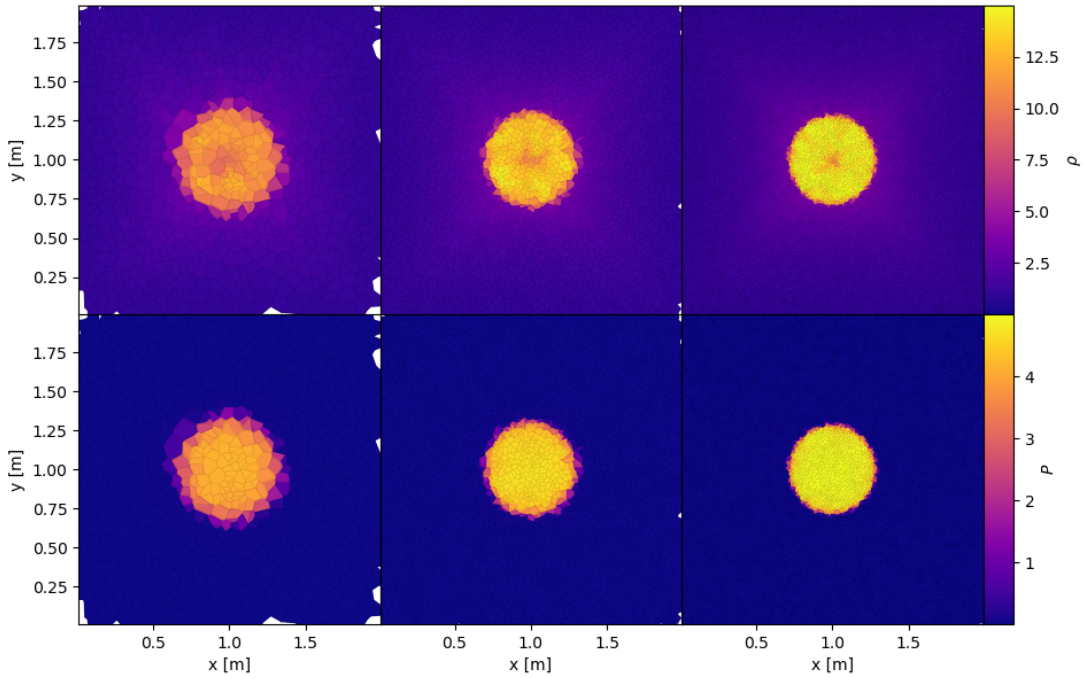

Figure 3.21 *Density (top row) and pressure (bottom row) distributions from the Noh problem at $t = 0.8s$, using the N1 solver, for resolutions using $N = 32^2$ (left), $N = 64^2$ (middle), and $N = 128^2$ (right) vertices.*

across the whole cylinder, demonstrating the expected pressure equilibrium. The lower density and equal pressure show that the temperature in the inner region must be higher. This heating phenomenon is known as 'wall heating', and has been previously identified in a number of Riemann-type hydro-solvers (Noh, 1987; Rider, 2000; Stone et al., 2008).

As mentioned before, the solution is radially symmetric about the centre of the box. I compare radially averaged profiles of the numerical solution to the analytic solution in Figure 3.22. The top panel shows the radial average of the density, the middle shows the magnitude of the velocity, and the bottom shows the pressure. Below each panel, I show the difference between the numerical and exact solutions $\phi = X_{\rm num} - X_{\rm exe}$ for each property. I show the absolute difference, instead of the fractional difference, because large parts of the exact solutions are zero, which produce an undefined fractional difference. These show several important features, most notably how the density within the shock increases as the resolution increases. Some of this can be explained by the decreasing width of the transition zone. As discussed above, the shock at the surface of the cylinder is resolved by several vertices, instead of the sharp transition of the physical shock. This means that the numerical solution has some material spread out

beyond the expected position of the shock, which is caused by a combination of numerical dissipation and the requirement that space be discretised into finite regions. This alone cannot contribute enough material to explain the difference. The velocity, shown in the middle panel, matches the expected values very well, with the exception of the transition zone, where the change from static to inwardly moving is once again spread out.

The position of the shock is replicated reasonably well, although its exact position in the numerical results is complicated by the widening of the shock's profile. As the resolution increases, however, the shock narrows, pulling in towards the predicted position. The inner edge of the shock remains in the same place, suggesting the numerical solution will converge on the analytic prediction with further increases in resolution. This pattern is the same for the density, velocity and pressure profiles. Beyond the shock, the density profile matches well until the outermost radii, where the numerical density drops below the exact solution. This becomes more pronounced at later times, as the cylinder expands further, and is likely caused by interference from the boundary conditions. I set the boundary vertices to constantly hold the initial conditions, whereas previous works have force the boundaries to the exact solution (Paardekooper, 2017).

Blob Test

The blob test combines both Kevin-Helmholtz instabilities and Rayleigh-Taylor instabilities by embedding a cold cloud within a hot flow (Agertz et al., 2007). This test consists of a high density, static, spherical cloud ρ_H, placed within a low density background ρ_L which moves with a bulk velocity v_0. In 2D this cloud is represented by a disk. The high density region is an order of magnitude more dense than the background medium. The whole region is in pressure equilibrium, with the low density wind much hotter than the cold cloud. The background medium is given a supersonic initial velocity, with Mach number $\mathcal{M} = v_0/c_s$, where c_s is the sound speed of the gas. Astrophysically, this corresponds to high density clouds moving relatively supersonically through a lower density background, such as a region of cold ISM close to a supernova.

The initial linear stages of the evolution of this set up can be predicted with some degree of confidence (Agertz et al., 2007). The collision of the supersonic flow with the static density front will produce a bow shock upwind of the cloud, with a subsonic region behind the front. The cloud itself will be accelerated

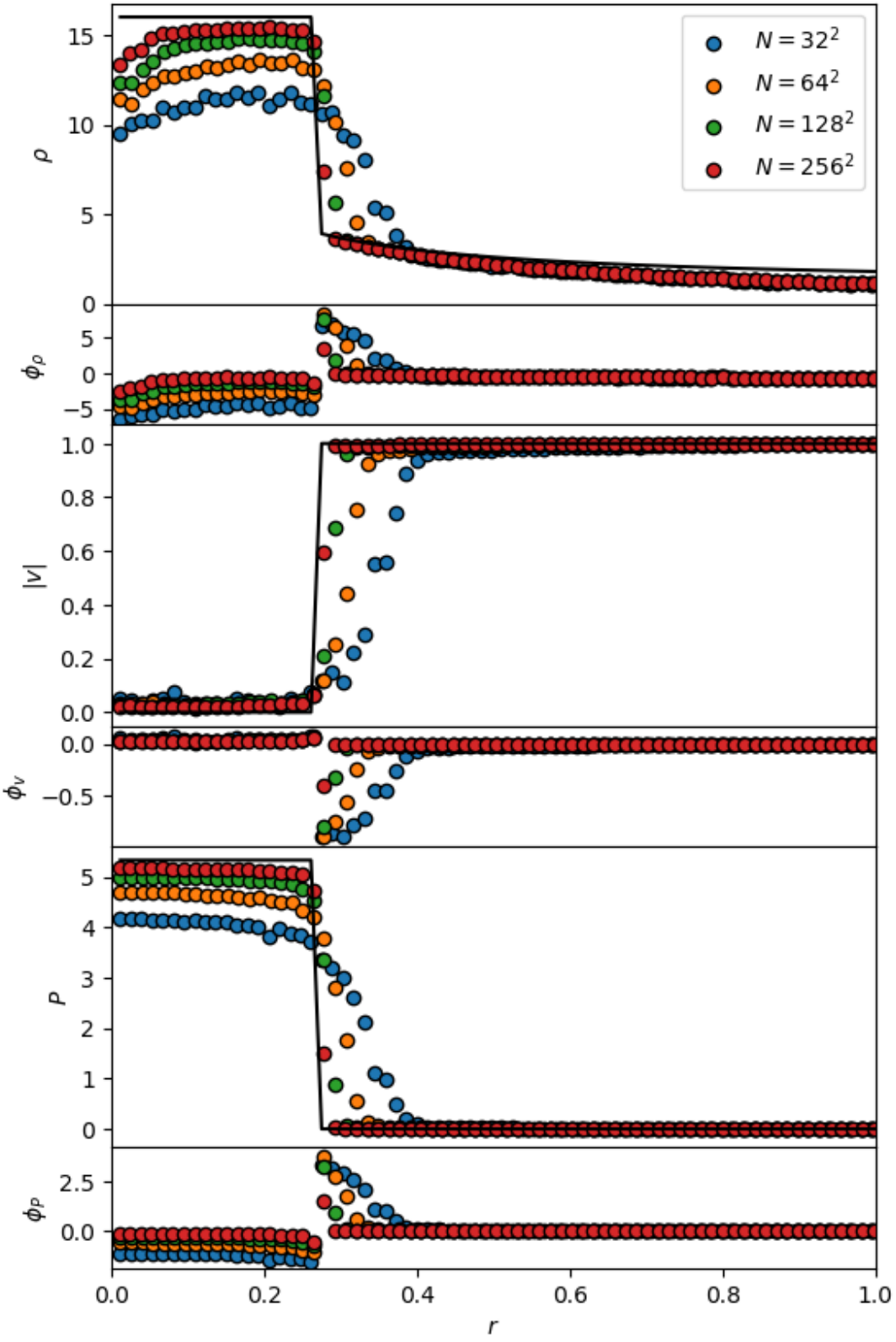

Figure 3.22 *Radial profiles from the Noh problem, showing the density (top), velocity magnitude (middle), and pressure (bottom). The blue dots show the results for $N = 32^2$, the orange show $N = 64^2$, the green show $N = 128^2$, and the red $N = 256^2$. The black line shows the exact solution. Below each profile is the residual $\phi = X_{\text{num}} - X_{\text{exe}}$ for each property.*

by its interaction with the flow. Kelvin-Helmholtz (KH) instabilities, discussed in isolation in Section 3.5.2, build at the boundaries between shear flows, such as the boundary between the cloud and the background medium, where the radial vector is orthogonal to the flow. At the same time, Rayleigh-Taylor (RT) instabilities evolve where the cloud is pushed into the downwind low density medium (Chandrasekhar, 1961). Together these instabilities lead to the breakup of the original cloud. Tendrils of high density are pushed downwind, and the original sphere is crushed by the incoming flow, and is eventually destroyed. The lower limit of the time for this to happen is predicted by the crushing time (Agertz et al., 2007)

$$\tau_{\rm cr} = \frac{2r_{\rm cl}\chi^{1/2}}{v_0}, \tag{3.87}$$

where r is the radius of the cloud, χ is the initial density contrast, and v is the relative velocity of the cloud and the background flow. The crushing time comes from the time it takes for the wind to cross the extent of the cloud, scaled by the ratio of cloud density to wind density. A greater difference will result in a longer time to disrupt. This can be used as a reasonable gauge of time scale for the cloud to be disrupted. The full physical evolution of the cloud is highly non-linear, and so cannot be easily predicted.

In Figure 3.23, I show the results for the blob test, using the N1 solver at three resolutions, with $N = 32^2$, $N = 64^2$, and $N = 128^2$. These are shown in columns from left to right. Time increases as we go down the panels, with the top row showing the initial conditions, the middle showing the state at $t = 1\tau_{\rm cr}$, and the bottom row at $t = 2\tau_{\rm cr}$. I use periodic boundaries to produce the results shown here. Inflow/outflow boundaries are currently in the process of being implemented and tested. The lowest resolution case inevitably starts with a very irregular cloud, rather than the desired spherical cloud. The higher resolution cases have much more regular initial shapes. It moves out from the initial density discontinuity at the windward cloud surface, eventually overlapping the edges of the box. By $t = 1\tau_{\rm cr}$ (middle row), the initial irregularities have largely been smoothed out, with only small remnants such as the high density prominence at about $(2.5, 4)$ in the central panel.

At this time, Rayleigh-Taylor instabilities are starting to develop in the medium and high resolution cases. Two tails can be seen forming behind the main cloud, as some material flow downstream from the cold cloud. Something similar is

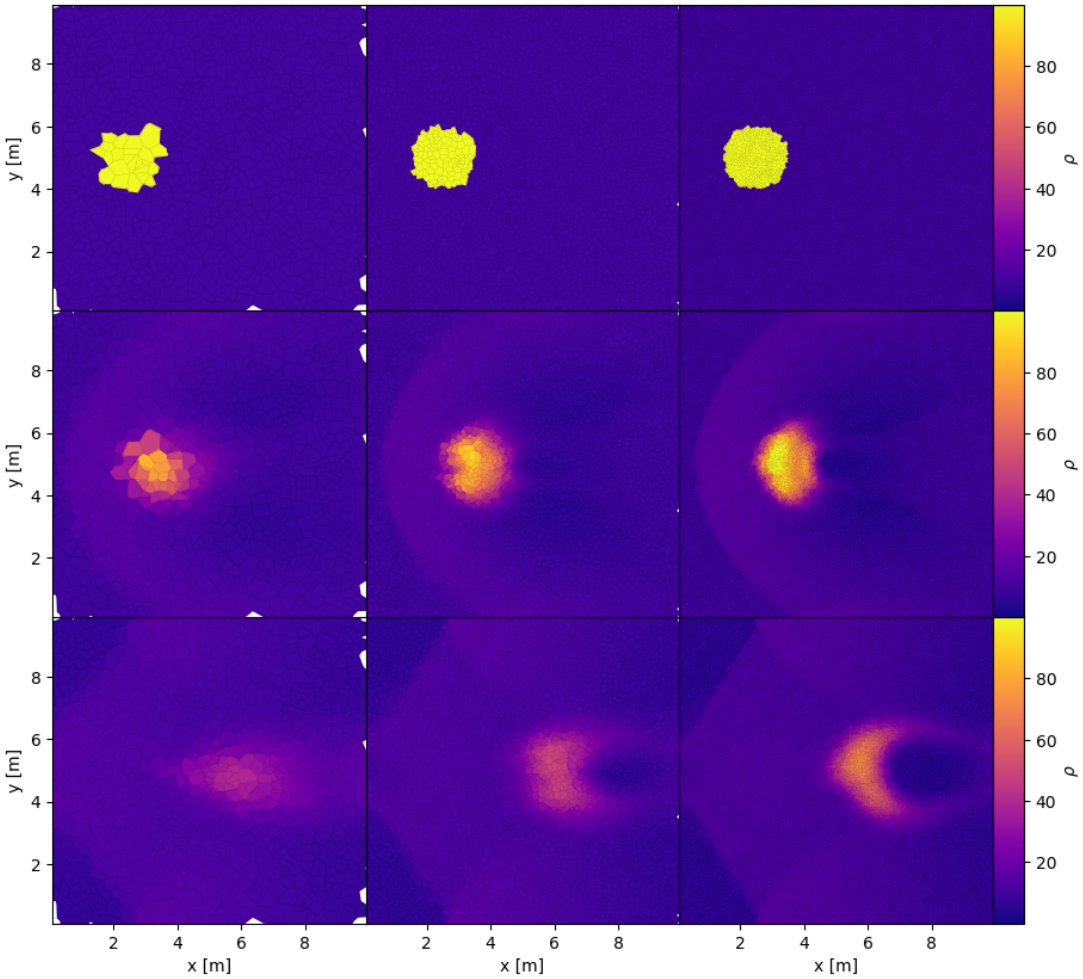

Figure 3.23 *Disruption of cold gas cloud by hot flow using the N1 solver. Time increases downwards, with the top row at $t = 0$, the middle at $t = 1\tau_{\rm cr}$, and the bottom at $t = 2\tau_{\rm cr}$. Each column has a different resolution, with, from left to right, $N = 32^2$, $N = 64^2$, and $N = 128^2$.*

happening in the lower resolution case, although the effect is less clearly defined. The bulk of the cloud remains intact at this point. In the highest resolution case, the cloud is being flattened, more so than at the lower resolutions. We see that the front is being forces downstream by the oncoming hot wind, and that this process is changing the shape of the cloud, to produce this more elongated shape. The lower resolution cases do not have enough elements in the cloud to resolve this effect well.

The cloud is significantly more disrupted in the last snapshot, at $t = 2\tau_{\rm cr}$. The lowest resolution case shows the cloud almost completely dissipated into its surroundings, while the medium and high resolution cases show more structured remnants. The differences are likely caused by the small number of resolution elements in the low resolution case being less able to shield the material behind them from the wind. The tails in the higher resolution cases are clearer at this later time, as more material is stripped from the cloud. Even in the high resolution case, the density of the cloud itself is much diminished, dropping from the initial $\rho = 100$ to $\rho = 60$. The point of cloud destruction is not well defined, but it is clear the process is well on it way to completion by $t = 2\tau_{\rm cr}$, which fits reasonably well with the estimation of $t_{\rm cr}$ as a crushing time.

These runs are close to the limit of the abilities of the code, in terms of number of time steps and resolution. Higher resolutions can be run, but the time to complete this test becomes prohibitively high. The problem has not converged with resolution, so further tests are required to understand the results in full, once greater resolutions can be reached through optimisations of the code. However, with the results presented here, we can see that the solver performs well at resolving the bow shock and tails, even at very low resolution.

Dynamical Friction

Here I show the first results for the dynamical friction (DF) test. In Chapter 2, I propose using this setup as a standard test of hydrodynamics and gravity. I place a Plummer potential for a mass M_p in a uniform density ρ_0 tube, at pressure P_0, and with sounds speed c_s. The gas is given an initial bulk velocity V_0, such that it has Mach number $\mathcal{M} = V_0/c_s$. There is no fully tested implementation of gravity in the code at this point. Instead, I apply the appropriate acceleration to the gas at the position of each vertex in the mesh. This is simply the acceleration from the Newtonian force created by the Plummer potential. For example, the

acceleration in the x-direction takes the form

$$a_x = -(x - x_0)\frac{GM_p}{(r^2 + \epsilon^2)^{3/2}}, \tag{3.88}$$

where G is the gravitational constant and epsilon is the softening length of the potential. The gas does not interact with itself gravitationally, so this alone can handle the gravity part of the DF test. As discussed in detail in Chapter 2, the problem is effectively scale free, being completely described by the Mach number $\mathcal{M}$, and the A parameter, which link the perturber mass to the gas state and the softening of the potential. It is calculated from

$$A = \frac{GM_p}{c_s^2 r_s}. \tag{3.89}$$

Any setup with the same Mach number and A value should produce the same over-dense wake. The results shown here are for cases run using $A = 0.1$, at $\mathcal{M} = 1.3$. As this is within the linear regime ($A \ll 1$), these can be compared directly to the analytic solution (Ostriker, 1999), both for the structure of the wake. The structure of the wake is recovered well, shown in Figure 3.24, although there is some evidence of the smeared density profile discussed in the previous chapter. The Mach cone and spherical parts of the wake are clearly present. The resolutions used here are far below those used in the previous chapter, here using $N = 64^2$ and $N = 128^2$. The resolution equivalent to that previous work would be $N = 512^2$, beyond what the code is currently capable of in a reasonable time frame. The fact that this spurious effect is of the same order at this lower resolution demonstrates the potential advantages of this truly multi-dimensional solver.

The low density trough in front of the leading edge of the cone is a new feature, and it is not yet clear what is causing it to form. The gravity implementation used here creates the expected gravitational accelerations, when checked with simple setups, and correctly reproduces a spherical density profile when there is no initial bulk flow. The trough is therefore a numerical artifact created by the scheme. The formation of the trough is independent of the underlying grid, forming no matter the structure of the grid. Further work is required to understand the differences between the RD solver results and the previous results from the MFM solver, as well as the differences between the RD numerical results and the analytic prediction.

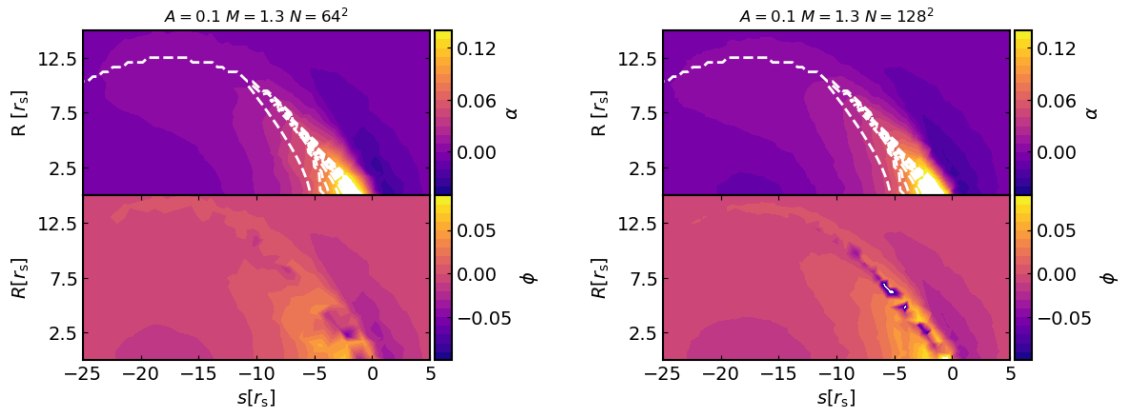

Figure 3.24 *Gravitationally induced wakes from the N1 solver, with $N = 64^2$ (left) and $N = 128^2$ vertices. Upper Panels: Colours show the over-density α, with the analytic prediction shown as white dashed contours. Lower Panels: Difference between the numerical and analytic wakes $\phi = \alpha_{\text{num}} - \alpha_{\text{ana}}$.*

In Figure 3.25, I show the force from the gravitationally induced wakes, for both $N = 64^2$ and $N = 128^2$. The force is found using the following process. The net force from mass at the position of every vertex is calculated by direct Newtonian force summation. The mass is the density at that position, multiplied by the dual volume of that vertex in the Voronoi mesh. As a reminder, the Voronoi mesh is simply the dual of the Delaunay triangulation used in the RD method. All vertices inside the softening scale r_s are excluded. In these cases, the force from the $N = 128^2$ case has the expected net zero force in the initial conditions, but the $N = 64^2$ case shows a large offset from zero. This is caused by a single vertex that is coincidentally very close to the edge of the excluded region. At this low resolution, a single close value will have a dramatic effect on the net force. A symmetric vertex distribution would remove this problem, but could also introduce spurious numerical artefacts, such as the carbuncles mentioned earlier. Instead, I show the force with the initial value subtracted from it. Since the mesh is static, the asymmetric contribution from the background density will be exactly constant. In my previous Lagrangian work, the set of particles move, so the exact background level varied with time. The force is given in its dimensionless form, normalised by

$$F_0 = \frac{4\pi\rho_0 \left(GM_p\right)^2}{c_s^2}. \tag{3.90}$$

The $N = 64^2$ force is shown in red, and the $N = 128^2$ in blue. Both resolutions systematically over-produce the net force on the perturber, except at very early

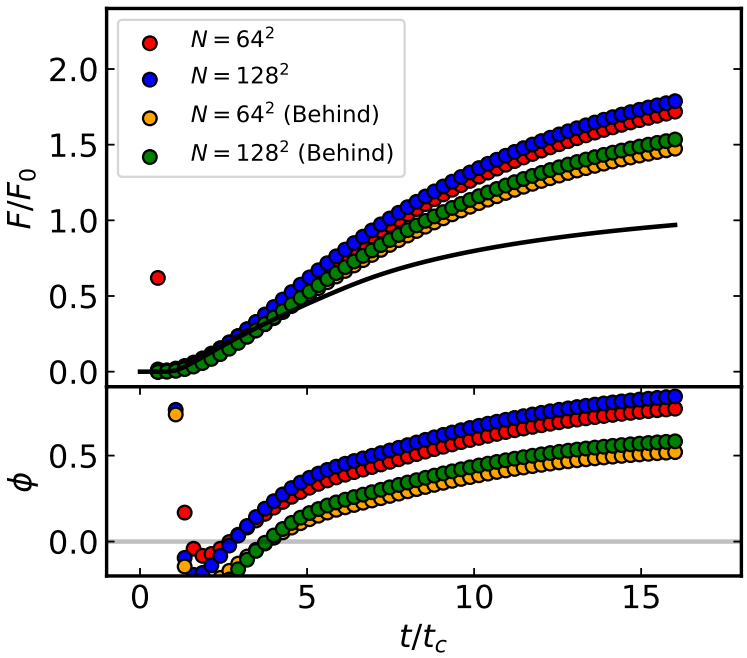

Figure 3.25 *Upper panel: Net force on the massive perturber. Results for $N = 64^2$ (red) and $N = 128^2$ (blue) vertices using all particles, or for the same resolutions, but using just those behind the perturber (orange and green respectively). Lower panel: Residual between numerical and analytic results $\phi = (F_{\text{num}} - F_{\text{ana}})/F_{\text{ana}}$.*

times ($t < 3t_c$), where they briefly under-produce the force. The force at very early times is unreliable, because the wake is either within, or will not have extended much beyond, the softening scale of the potential. At later times, the force is clearly too high, and gets slightly higher with increased resolution. The fractional residual $\phi = (\phi_{\text{num}} - \phi_{\text{ana}})/\phi_{\text{ana}}$ between the numerical and analytic results increases with time, showing that the result gets proportionally worse with time. However, the trajectory of the residual, while still increasing, does appears to be leveling off in all cases.

The second set of forces, this time with $N = 64^2$ in orange, and $N = 128^2$ in green, have the force calculated using only those vertices behind the perturber. This excludes all particles with $s > 0$, as well as those inside the softening scale. This excludes much of the low density trough in front of the cone, and does improve the match between the forces somewhat. However, there is still a significant offset between the results, which must be caused by inconsistencies in the structures of the numerical and analytic wakes. Once again, the higher resolution results are a slightly worse match.

The early results for DF with the RD solver are very promising, but further

work is need to understand why the numerical solution does not fully match the analytic prediction. It could simply be a matter of resolution, in which case future runs, produced once the code is optimised for larger problems, should produce a better match. On the other hand there is little difference, in the actual shape of the wake, between the two resolutions shown here. There is also the curious low density trough, that was not found in my previous DF work. This also remains largely unchanged between resolutions.

3.6 Conclusion

In this chapter, I introduce the fundamental concepts behind the development of the residual distribution hydrodynamics solvers, including the one dimensional equivalent and the two dimensional form. This solver is truly multi-dimensional, as it avoids all dimensional splitting, and contains a whole family of methods that are all built around the same base: calculating a residual over a triangular element, in a single calculation, and then distributing this to the vertices of the element, to update the solution to the set of PDEs that are being solved. I cover the various choices that can be made when designing a specific implementation of such a solver, which define the resultant characteristics and abilities of the code. This includes the required linearisation of the Euler fluid equations, first laid out by Roe. I also introduce Delaunay triangulation, with a brief description of its definition, properties and construction. I discuss the extensive testing I performed on my RD solver implementation, covering one and two dimensional test cases. These tests demonstrate the strengths of the solver in recovering multi-dimensional flows, while also handling shocks and other extreme situations well. The RD implementation that I describe and test in this chapter represents what the solver can do without significant optimisation and tailoring to a given problem. It performs well when compared to current stat-of-art solvers, and it can resolve complex structures at low resolution. There is still significant scope to optimise the RD solvers to any desired problem, with a straightforward framework for implementing different distribution and blending schemes. The fact that it is built around an unstructured mesh makes it the perfect candidate for conversion into a moving-mesh scheme. It thus has great potential for further improvement. Overall, the RD solvers presented in this implementation are well on their way to being powerful new tools for running astrophysical simulations of a range of scenarios.

Chapter 4

Extensions to RD Approach

4.1 Introduction

In the previous chapter, I introduced the residual distribution family of solvers, and derived the specific form of solution for the 2D Euler equations, alongside a large number of aggressive tests that demonstrate it strengths and weaknesses. In this chapter, I describe the extensions that I have derived for, and implemented into, the basic form of the solver. This includes the investigation of variable time stepping mechanisms, to improve the computational efficiency of the approach, and the derivation and implementation of the 3D form of the solver. I also briefly discuss the most effective strategies for introducing the effects of gravity.

4.2 Variable Time Stepping

As discussed before, the time step is defined by the CFL condition, requiring that the numerical domain of dependence encloses the entire physical domain of dependence. In the standard formulation, the time step at which the fluid state at every vertex is updated, is dictated by the smallest time step required by any vertex in the mesh. In scenarios with regularised meshes, or with only small variations in density and velocity across the grid, this implementation is not particularly significant, as all vertices will require similar time steps. However, scenarios with extreme density and velocity contrasts will be computationally inefficient to run. If only a few cells require a time step that is orders of magnitude

shorter than the rest of the mesh, then the residuals will be recalculated far more often than is numerically required for many triangles. This problem is encountered in all numerical methods for solving fluid dynamics problems. There is always some limit on the time step at which the cells or particles can be updated, and large variations in this number lead to inevitable inefficiencies. To combat this problem, and produce methods that can utilise the available computational power more efficiently, many numerical methods have introduced mechanisms that allow different fluid elements to be updated with different time steps.

Typically, these approaches divide elements into groups based on their required time step, and then recalculate the evolution of the state based on these time step bins. For simplicity I will only discuss grid based approaches from here, but equivalent particle methods are widely used. As discussed in detail in the introduction, in a standard cell based approach (for either a structured static mesh, or an unstructured or even moving mesh), the fluid state is updated by calculating the flux of material through the faces of cells. A simple way to implement a varied time step in such methods is outlined in Springel, 2010. You first bin each cell by the required time step, then to recalculate the flux through faces based on the smallest value either side of that face. The fluid state is still updated at the smallest time step of the whole mesh, but the flux is not recalculated for every face at every small time step. The flux through faces whose required time step is longer than this is simply kept the same, until it is necessary to update it. Using old updates is known as drifting, as the state continues on the same trajectory for multiple small time steps.

To produce the equivalent effect with the RD solver, it is the residual that is calculated at different steps for different triangles, as this is the analogous calculation to the flux in the standard grid approach. I have implemented two novel strategies to achieve this outcome. While both achieve the same end result, they do so in different ways, and have slightly different properties. Both methods start in the same way. The minimum time step for every *vertex* is calculated using the limit described in the previous chapter. Each *triangle* is then binned based on the smallest time step required by any of its vertices. The time step bins have limits based on powers of two times the overall minimum time step Δt_{min}, such that the smallest bin has limits $\Delta t_{\text{min}} < \Delta t_{\text{req}} < 2\Delta t_{\text{min}}$, the next bin has $2\Delta t_{\text{min}} < \Delta t_{\text{req}} < 4\Delta t_{\text{min}}$, and so on. Now when the simulation is evolved, residuals are only recalculated after the lower limit of their time step bin has elapsed since last calculation. Every triangle is checked, but only some have

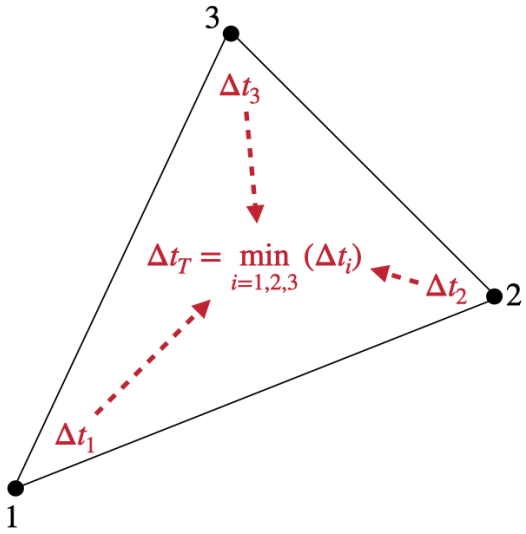

Figure 4.1 *The time step bin of the triangle Δt_T is the minimum of the bins assigned to the vertices of the triangle (Δt_1,Δt_2,Δt_3).*

their residual recalculated, which are referred to as *active* triangles. This saves significant computational time by not recalculating residuals more frequently than required. Over the course of the large time step, which is defined as the lower limit of whichever is the largest time step bin, the following will happen. Taking the simplest example of only two time step bins, there will be two small time steps modelled for the one large time step. At the beginning of the large time step, the residual of every triangle is calculated, but one small time step later, only the triangles in the smallest bin will recalculate their residual. Another small time step later, we have completed a full top level time step, and so start the process over. The way that the residual is passed to the vertices is different for the two methods I have implemented, which I describe in more detail below. Once the top level time step has been completed, the binning process is repeated.

4.2.1 Drift Method

The first adaptive time step method effectively 'drifts' the states of vertices associated with triangles in bins above the bottom level. The update is passed based on the current residual of that triangle, even if that residual has not been recently recalculated. The updates from long time step bins can be said to drift the state of the associated vertices because the changes continue along constant trajectories, as if they are drifting in some direction, without being deflected by additional forces, hence the approach's label as the `DRIFT` method. A 1D

representation of the concept of this method is shown in Figure 4.2. The red discs represent vertices that are in the bottom level, shortest time step bin, while the blue discs represent those in the top level bin (assuming a two bin system). The spaces between each disc represents the element for which the residual is calculated. Time increases in the y-direction, with each row representing the vertices at a given time. The arrow represents the passing of an update, based on the element's residual, to each associated vertex. Solid arrows show the distribution of residual calculated that time step, while dashed arrows show the passing of residual calculated at a previous time. For the first small time step dt, the residual passed to each triangle is exactly the same as it would be in the original system, but for the second small time step, the residual from the left two elements are based on an old fluid state. They have not been updated, as the fluid states at the vertices of these elements do not require the short time steps.

I found that this approach provides a significant boost to performance (see Section 4.2.4), but also that there is a loss of exact conservation. The total mass and energy of the box change over time, while for the basic universal time step method preserves conservation to machine precision. The loss of conservation is caused by the peculiarities of the residual method itself. In typical grid based methods, where the change in the fluid state of a cell is calculated by estimating fluxes through the surface of the cell, conservation is trivial to maintain for most situations. Any material that flows from one cell is added to its neighbour. Conservation is explicitly maintained across every cell boundary. The residual method, however, does not maintain conservation across the equivalent structures: the triangular elements. For a given element the net change in the states of all vertices, produced by the residual, is not zero. It is zero in the special case of a steady flow, when the element residual is also zero. Conservation is ensured between a vertex, and all of its neighbours, but only once all residuals have been distributed. In the normal, global time step, setup, this is perfectly adequate, as all residual are recalculated every time step, and so consistent updates are present everywhere.

However, when I use the `DRIFT` approach, some updates are based on outdated residuals. These updates assume vertex states that no longer exist. In the standard flux approach, this is not a problem for conservation, because even if the current flux is not exactly physically correct, it is the same incorrect value on either side of the face. In the RD case, neighbouring triangles are using residuals from inconsistent states, effectively breaking the guarantee of conservation, that

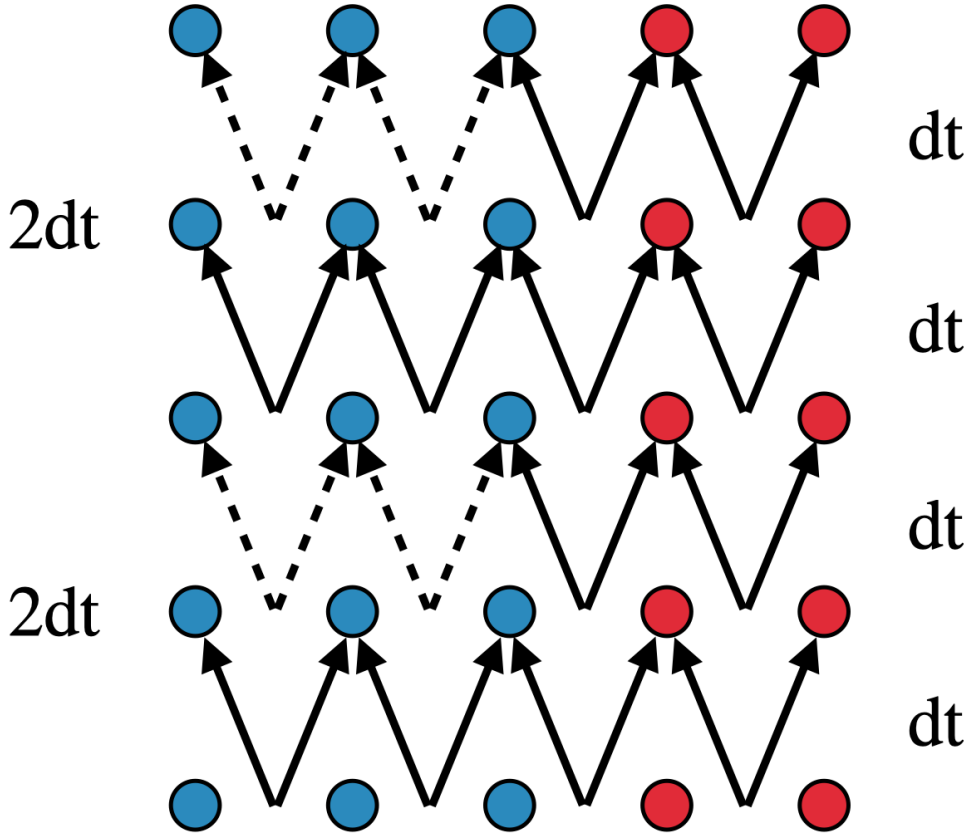

Figure 4.2 *Stencil for the* `DRIFT` *adaptive showing distribution of residual in 1D, with time increasing in the y-direction. Dots represent the vertices where the fluid state is held, while the spaces between them in the x-direction are the elements for which the residuals are calculated. Red vertices require time steps of dt, and the blue dots 2dt. The arrows show where residuals are distributed. The solid arrows represents residuals that have been recalculated that turn, while the dashed line represents residuals that have not been updated.*

relies on neighbouring triangles calculating residuals from the same states at the shared vertices. The total change in mass and energy across standard hydro tests is very small, and does not preclude using this variable time step approach, but it does add a potential complication to using these RD methods. I discuss the conservation loss, in more detail, in Section 4.2.3, showing the time evolution of mass and energy at different resolutions.

4.2.2 Jump Method

The second method, which I refer to as the `JUMP` method, attempts to avoid the conservation problem by passing updates for the whole required time step of the given triangle. Instead of drifting the state of vertices attached to long time step triangles, the states are only updated by active triangles. Vertices at the borders between time bins receive updates from triangles at different rates, sometimes being updated by all associated triangles, and sometimes from a subset of them. This is illustrated in Figure 4.3. The vertex in the central column is such a boundary vertex. This tests whether conservation can be maintained if the residuals that are passed are all based on a consistent residual at the time that they are calculated. Effectively, the main difference is that the boundary vertices do not receive some of their update until the end of the large time step. Unfortunately, this is not enough to fully solve the problem, and the conservation loss persists.

4.2.3 Conservation

I test the effect of the loss of conservation present in the two potential methods. To do this, and to test how it varies with both the number of time step bins used, and the resolution of the simulation, I run a set Kelvin-Helmholtz instability cases. These use the setup described in the previous chapter (Section 3.5.2), but this time using the `DRIFT` and `JUMP` adaptive time step approaches.

Figure 4.4 shows the fractional change in total mass and total energy in the top part and bottom part of each panel respectively. These results are for the `DRIFT` method. The blue line show the change for only one time bin $N_{\rm bin} = 1$, orange for $N_{\rm bin} = 2$, green for $N_{\rm bin} = 4$, and red for $N_{\rm bin} = 8$. The different panels show results for different resolutions, with the top showing results for $N = 32^2$, the

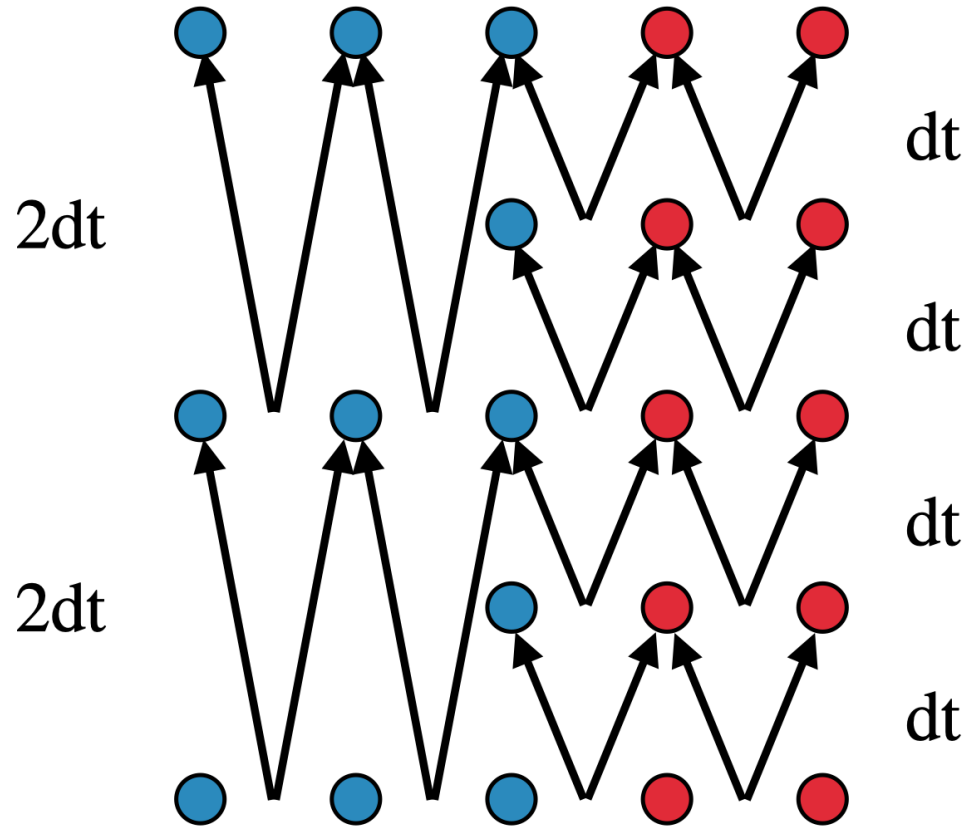

Figure 4.3 *Stencil for the* `JUMP` *adaptive time stepping approach. The blue vertices do not receive updates from the 2dt triangles until the end of the long time step.*

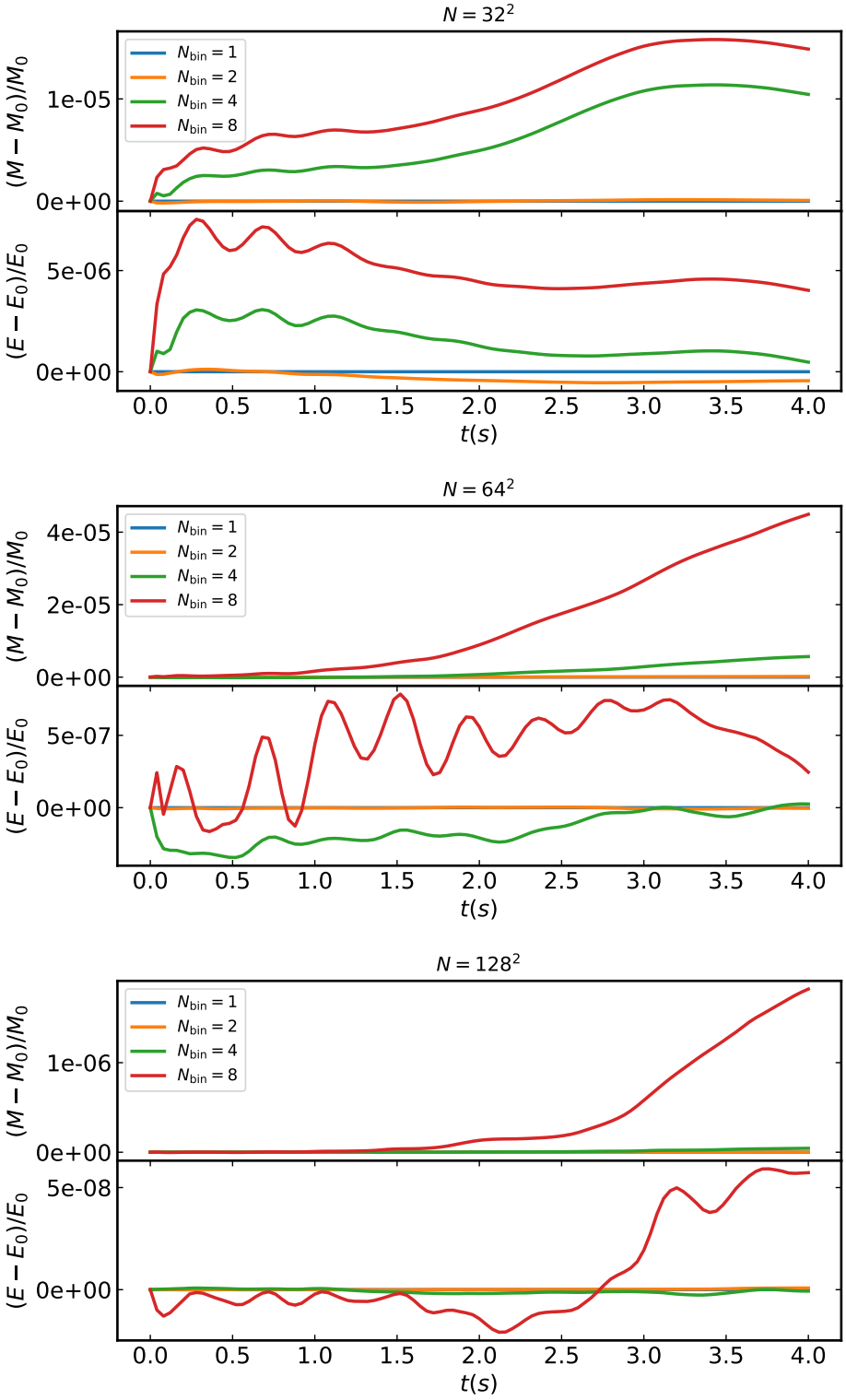

Figure 4.4 *Variation in total mass (upper part of each panel) and energy (lower part of each panel) using the* `DRIFT` *method, for* $N = 32^2$ *(top panel),* $N = 64^2$ *(middle panel), and* $N = 128^2$ *(bottom panel). These show the fractional change from the initial total mass and energy. The lines show the results for different numbers of time step bins, where we have* $N_{\text{bin}} = 1$ *(blue),* $N_{\text{bin}} = 2$ *(orange),* $N_{\text{bin}} = 4$ *green), and* $N_{\text{bin}} = 8$ *(red).*

middle for $N = 64^2$, and the bottom for $N = 128^2$. Across all resolutions, we see that the greater the number of time bins, the greater the change in mass and energy. This fits with the previously discussed reason for the change, as more bins means more places where there are neighbouring triangles in different bins. The change is reduced with increased resolution, which is a positive sign. The variable time stepping is of most use with large vertex numbers, so this reduction is welcome. The exact change will vary from case to case, depending on the specific distribution of bins, but in this example, where the total change is shown over tens of thousands of time steps, the absolute change is very small. It is a tiny fraction of one percent, and so should not pose a major problem, especially when high resolution is used. The trend in the absolute change of mass or energy also varies, seemingly randomly across resolution. For instance the energy in the $N = 64^2$ case oscillates significantly, while the mass simply increases. This is likely again dictated by the specifics of both the mesh and the problem.

The equivalent `JUMP` results are shown in Figure 4.5. The same pattern observed in the `DRIFT` results is seen here, except for the $N = 32^2$ results in the top panel. For some reason the $N_{\text{bin}} = 4$ is much worse than all the others. It is not clear why this is happening in this case, as the other resolutions show the same pattern as before. Possibly some specific combination of time bins is causing a constant mass and energy loss that is dominating the change, but at the higher bin number the higher level bins are counteracting this effect, leading to a much smaller net change. The results for this method, as a whole, are worse than those for the `DRIFT` approach. We see net changes approximately two orders of magnitude worse in this case. Clearly drifting the vertex states is the better option.

This systematic difference can be seen clearly in Figure 4.6, where I show the root-mean-square change from one output to the next for the different runs. Each point represents a different simulation setup. The dots represent the runs with $N_{\text{bin}} = 2$, the pluses are $N_{\text{bin}} = 4$, and the crosses $N_{\text{bin}} = 8$. The blue points show results from the `DRIFT` approach, and red the `JUMP` method. The improvement with resolution is clear in all cases. The blue points are all below their red counterparts, so the superiority of `DRIFT` is clear.

In the vast majority of the cases shown here, the deviation from the conserved value is several orders of magnitude below the percent level, even over tens of thousands of time steps. This shows that, while the approach is not perfect, it can be used effectively without endangering the accuracy of a simulation. The underlying method is itself a numerical approximation, so this mechanism can

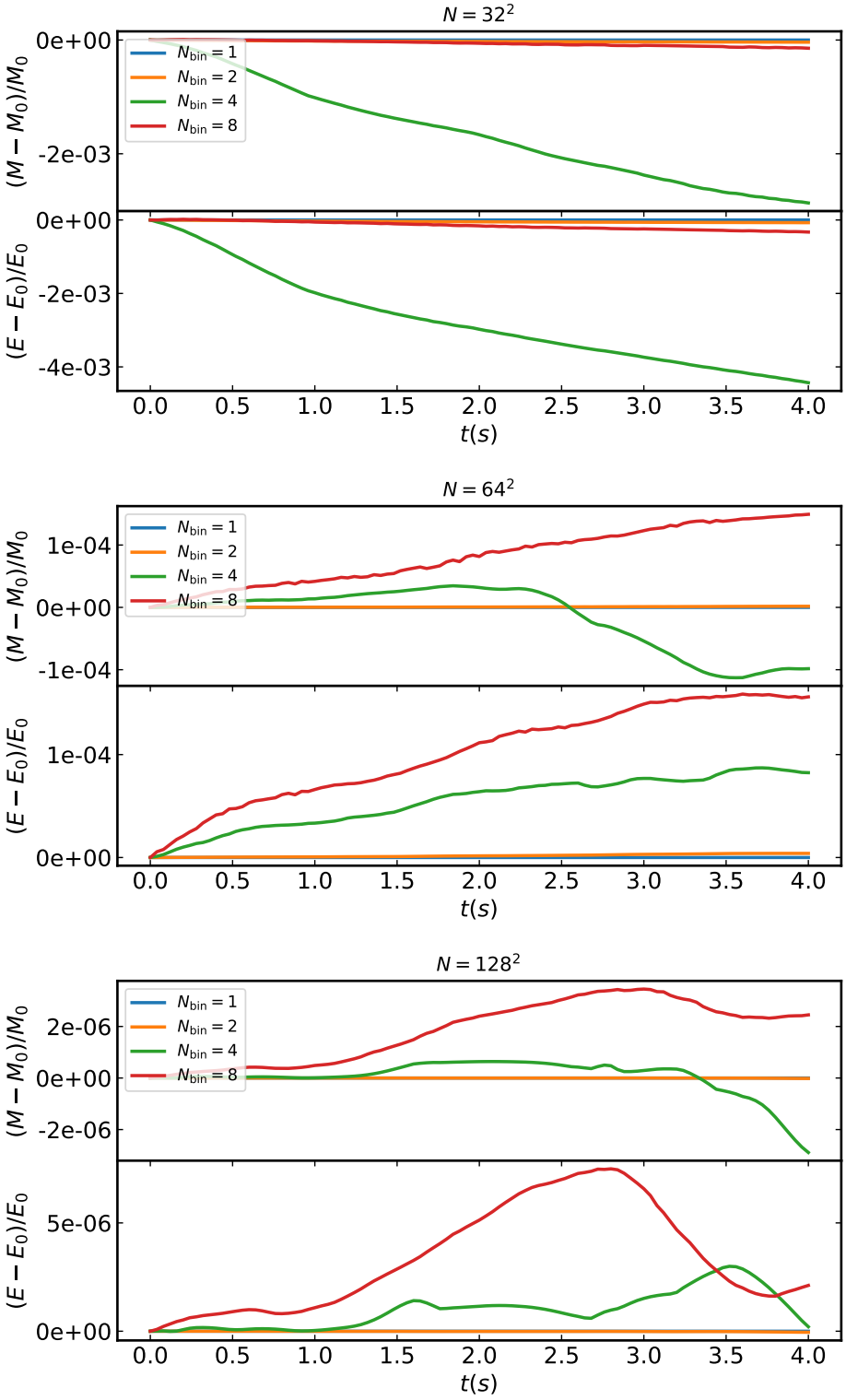

Figure 4.5 *Variation in total mass (upper part of each panel) and energy (lower part of each panel) using the JUMP method, for $N = 32^2$ (top panel), $N = 64^2$ (middle panel), and $N = 128^2$ (bottom panel). These show the fractional change from the initial total mass and energy. The lines show the results for different numbers of time step bins, where we have $N_{\rm bin} = 1$ (blue), $N_{\rm bin} = 2$ (orange), $N_{\rm bin} = 4$ green), and $N_{\rm bin} = 8$ (red).*

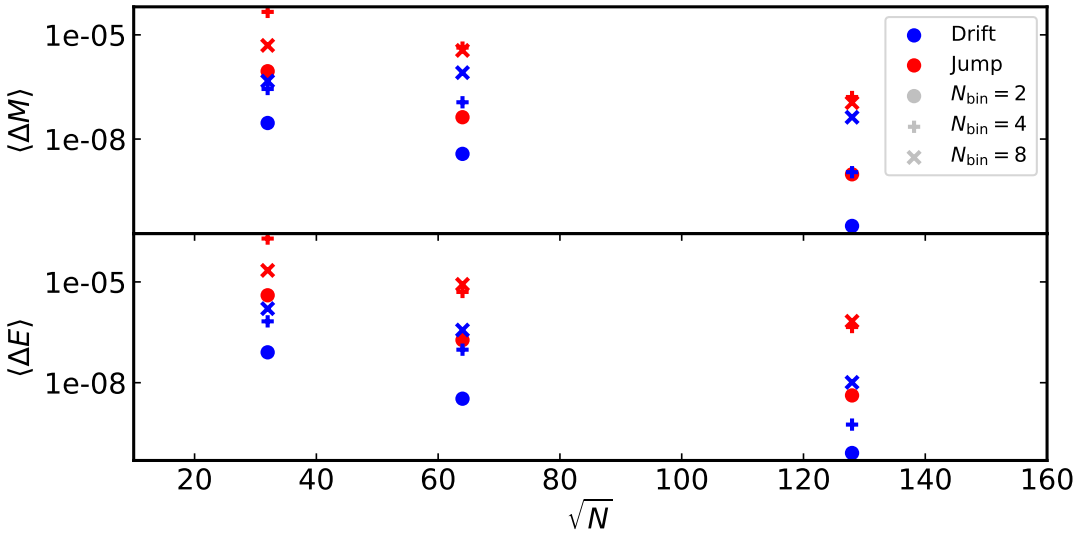

Figure 4.6 *Comparison of the root-mean-squared change in total mass for each simulation. The changes ΔM and ΔE are the change in total mass and energy from one snapshot to the next, and the RMS is calculated for all these changes in a given run. Each run has a method,* `DRIFT` *(blue) or* `JUMP` *(red), and a number of time step bins, $N_{\mathrm{bin}} = 2$ (dots), $N_{\mathrm{bin}} = 4$ (pluses), and $N_{\mathrm{bin}} = 8$ (crosses).*

simply considered an additional approximation that also contributes a significant resource advantage. When using this approach, it is important that one always remembers that this new approximation is present, and monitor its behaviour when using it on a new problem. If this is done, then the variable time step approaches described here can be used effectively in future work.

4.2.4 Performance

The main aim of implementing the variable time stepping is to improve the computational efficiency of the code. This can be simply measured by comparing the run time for different numbers of time step bins. The precise change in computational time will depend on the specific problem, as different situations will produce different time bin distributions. Here I show an example of the improvement in run time that can be achieved using these methods. In Figure 4.7, I show the time taken to run the same KH problem, against number of time step bins. Both methods perform very similarly, which is expected, since the primary speed up comes from the binning mechanism, which is the same in both cases. Both show a significant decrease in run time from $N_{\mathrm{bin}} = 1$ to $N_{\mathrm{bin}} = 2$, with the time almost halving in this step. The further increases in bin number

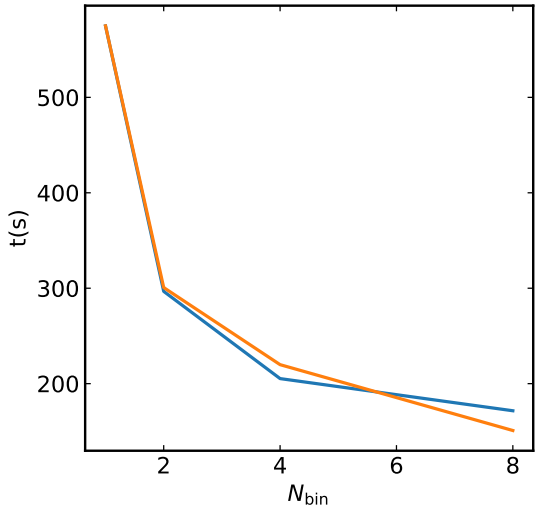

Figure 4.7 *Time take to run a Kelvin-Helmholtz ($N = 64^2$) test case, with varying numbers of time step bins, using* `DRIFT` *(blue) and* `JUMP` *(orange). There is strong improvement in the run time, as the number of time bins is increased.*

also show improvements in run time, although the effect is less stark.

I show the time step bin of each triangle in the mesh, in Figure 4.8, for the $N = 64^2$ KH test. Blue triangles, of which there are very few, represent $t_{\text{bin}} = 1$, while orange triangles are for $t_{\text{bin}} = 2$, red for $t_{\text{bin}} = 4$, and grey for $t_{\text{bin}} = 8$. This distribution is also clearly seen in Figure 4.9, with the bar chart showing the number of triangles in each time step bin, for the same snapshot. The jump in performance from $N_{\text{bin}} = 1$ to $N_{\text{bin}} = 2$ can be explained by this result, since all those bins in red and grey will be in the $t_{\text{bin}} = 2$ bin, when there are only two available. Therefore almost all bins require residual calculations at half the previous frequency, accounting for the majority of the performance boost. Another improvement of about a third is found when allowing four bins, which is explained in the same way, with a slightly lower proportional effect because of the larger number of orange triangles.

The new time stepping formalism shows a lot of potential for improving the computational efficiency of this RD implementation. The complex question of maintaining the exact conservation of mass and energy requires some careful consideration, but, at this stage, the effect is very small, and the improvement in performance large enough to justify continued use of the mechanism. A potential avenue to consider in the future would be the development of a distribution scheme that can conserve the properties over a single triangle. It is not clear, at this time, if this is possible within the RD framework.

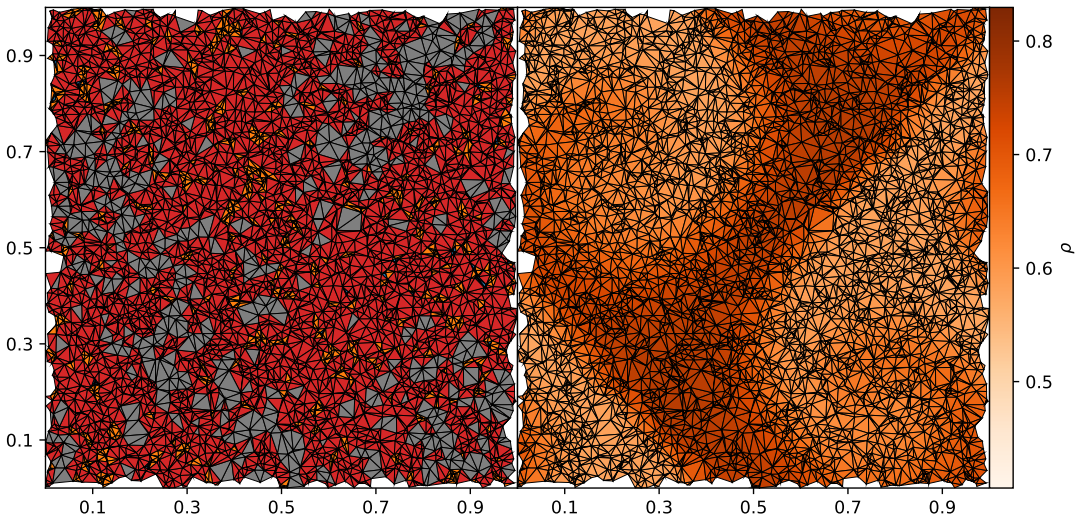

Figure 4.8 *Left Panel: Triangle time bin, with blue showing the smallest time step bin* $t_{\text{bin}} = 1$*, orange* $t_{\text{bin}} = 2$*, red* $t_{\text{bin}} = 4$*, and grey* $t_{\text{bin}} = 8$*. From this particular time, we can see that only a very small number of triangles are in the smallest time bin, with the most being in the top two levels. Going from* $N_{\text{bin}} = 1$ *to* $N_{\text{bin}} = 2$*, therefore, doubles the time step for almost all triangles, resulting in the greatest speed up. Right Panel: Density distribution from the same output, showing the highest level bins are mostly found in the high density regions.*

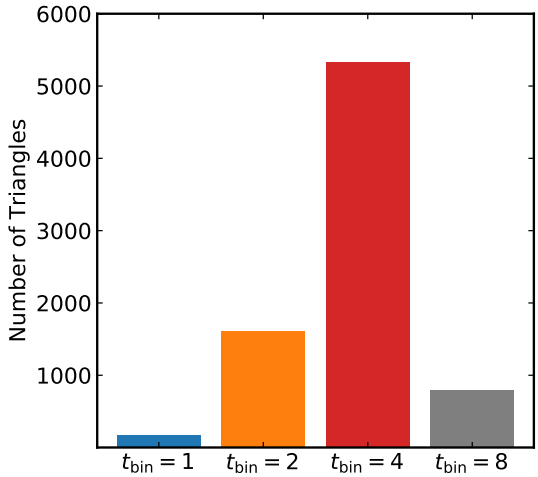

Figure 4.9 *Bars show the number of triangles in each time step bin, using the same color coding as the plot of the whole mesh.*

4.3 3D Extension

The solver presented in the previous chapter demonstrates the abilities of the residual distribution method to solve the fluid equations in a truly multidimensional manner, but only in 2D. Some astrophysical systems can be effectively modelled using only two dimensions, such as thin discs, where useful results can be found without calculating flows in the third direction. However, many more systems, from the cosmic web down to giant molecular clouds, are more accurately described by the full three dimensional flows. The RD approach naturally extends to extra dimensions, since the basic form is generalised to any number of dimensions. The key problem that must be solved is for the form of the K matrix (see Section 4.3.2).

Since I am now working with 3D flows, the fluid equations must be extended to include conservation of z-momentum, and z-velocities in the kinetic energy. The 3D form of the Euler equations is given by the fluid variable vector $\mathbf{Q}$, and flux vectors $\mathcal{F}(\mathbf{Q}) = (\mathbf{F}_x(\mathbf{Q}), \mathbf{F}_y(\mathbf{Q}), \mathbf{F}_z(\mathbf{Q}))$, which take forms clearly analogous to the 1D and 2D cases, with additional terms for the new dimension. The fluid variable and flux vectors are given by

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ \rho E \end{pmatrix}, \quad \mathbf{F}_x(\mathbf{Q}) = \begin{pmatrix} \rho v_x \\ \rho v_x^2 + p \\ \rho v_x v_y \\ \rho v_x v_z \\ \rho v_x H \end{pmatrix}, \quad \mathbf{F}_y(\mathbf{Q}) = \begin{pmatrix} \rho v_y \\ \rho v_y v_x \\ \rho v_y^2 + p \\ \rho v_y v_z \\ \rho v_y H \end{pmatrix}, \quad \mathbf{F}_z(\mathbf{Q}) = \begin{pmatrix} \rho v_z \\ \rho v_z v_x \\ \rho v_z v_y \\ \rho v_z^2 + p \\ \rho v_z H \end{pmatrix}, \tag{4.1}$$

where all terms have their usual meanings. Together these describe the conservation of mass momentum and energy in three dimensions. The corresponding Roe parameter becomes $\mathbf{Z} = \sqrt{\rho}(1, v_x, v_y, v_z, H)$. This parameter is used to linearise the Euler equations, as before. To recap the various terms, which are defined similarly to their 2D equivalents, but with the addition of the new direction. The pressure is defined the ideal gas equation of state, as

$$P = \rho(\gamma - 1)\left(E - \frac{\mathbf{v} \cdot \mathbf{v}}{2}\right), \tag{4.2}$$

where the velocity is now $\mathbf{v} = (v_x, v_y, v_z)$, and the enthalpy H is

$$H = E + \frac{P}{\rho}. \tag{4.3}$$

The sound speed c_s of the gas is

$$c_s = \sqrt{\frac{\gamma P}{\rho}}. \tag{4.4}$$

These describe all physical variables required to model the evolution of an inviscid fluid. The equation that is being solved is still written compactly as

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathcal{F}(\mathbf{Q}) = 0. \tag{4.5}$$

In this section, I will derive the formulation of the 3D residual distribution solver. To the best of my knowledge, this has not been explicitly shown in the literature to date. This will include the changes to the discrete form of the fundamental equations, including the calculation of the element residual, and the various geometric parameters that must be constructed. This is followed by a description of the steps required to formulate the 3D form of the inflow matrix. I also present initial tests of the 3D RD solver, comparing results to both the analytic predictions and the 2D formulation results.

4.3.1 Discrete Update

The 2D formulation of the RD method, laid out in Section 3.2, is easily extended to three dimensions. The definition of the domain and mesh remain the same, but now the domain has three dimensions, and mesh is made of simplices that are now tetrahedrons, rather than triangles. As before the simplices are defined by the Delaunay condition, which now requires an empty circumsphere, in place of the empty circumdisk in 2D. The corresponding dual tessellation, once again the Voronoi tessellation, but in 3D, has dual volumes $|V_i|$. Construction of this mesh, still sometimes referred to as a triangulation, is performed using 3D version of the same techniques as before.

The spatial part of the differential equation is now solved by integrating over volume, not area. The element residual is effectively unchanged, in that it is the integral over an element, it is simply that the element has an additional dimension. When it comes to the form of the update, therefore, the piece-wise constant integral over the temporal part of the PDE introduces the dual volume, where before it produced the dual area $|S_i|$. Following exactly the same reasoning as in the 2D case, it is clear that the intermediate state update, to the fluid state

at vertex i, is now given by

$$\mathbf{Q}_i^* = \mathbf{Q}_i^n - \frac{\Delta t}{|V_i|} \sum_{T|i\in T} \phi_i, \tag{4.6}$$

and the final update by

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^* - \frac{\Delta t}{|V_i|} \sum_{T|i\in T} \Phi_i. \tag{4.7}$$

The dual volume of a given vertex is found by assigning one quarter of the volume of each associated element to the vertex

$$|V_i| = \sum_{T|i\in T} \frac{|T|}{4}. \tag{4.8}$$

The residual is calculated as the sum of contributions from all vertices. The contribution is defined by the inflow matrix $\mathbf{K}_i$, which combines the Jacobians of each dimensions, and linearised average state $\hat{\mathbf{Q}}$, which together define

$$\phi^T = \sum_{i=1}^{4} \mathbf{K}_i \hat{\mathbf{Q}}. \tag{4.9}$$

The summation is now over four vertices, and the total residual is defined as before. The inflow matrix will be addressed in Section 4.3.2. As for the linearised average state, this has the same definition as before (see Equation 3.77), but now with the additional terms from the conservation of z-momentum. This produces

$$\hat{\mathbf{Q}}_i = \begin{pmatrix} 2\bar{Z}_1 Z_1 \\ \bar{Z}_2 Z_1 + \bar{Z}_1 Z_2 \\ \bar{Z}_3 Z_1 + \bar{Z}_1 Z_3 \\ \bar{Z}_4 Z_1 + \bar{Z}_1 Z_4 \\ \frac{1}{\gamma}\left(\bar{Z}_5 Z_1 + \gamma_1\left(\bar{Z}_2 Z_2 + \bar{Z}_3 Z_3 + \bar{Z}_4 Z_4\right) + \bar{Z}_1 Z_5\right) \end{pmatrix}. \tag{4.10}$$

The various distribution schemes are defined identically to before. Together these allow one to construct the three dimensional version of an the RD solver. The key part of the extension, however, is the formualtion of the inflow matrix.

4.3.2 K-Matrix

The flow at any given vertex is parameterised by the inflow matrix K_i, which can be seen as representing the flow of material into the element at vertex i. It does this by multiplying the Jacobians in each dimension with the component of the normal of the opposite face to that vertex. The formulation of the matrix is directly analogous to the 2D solvers K-matrix, where the Jacobian is combined with the opposite edge normal. The inflow matrix is generalised by

$$K_i = \frac{1}{d}\left(\sum_{j=1}^{d} A_j \hat{\mathbf{x}}_j\right) \cdot \mathbf{n}_i, \tag{4.11}$$

where d is the number of dimensions, $A_j = \partial \mathbf{F}_j / \partial \mathbf{Q}$ is the Jacobian matrix in dimension j, and $\hat{\mathbf{x}} = (\hat{x}, \hat{y}, \hat{z})$ is the unit vector, where j denotes the appropriate dimension. The Jacobian is evaluated with the average Roe parameter for the element. The $\mathbf{n}_i = (n_x, n_y, n_z)$ parameter is the component of the opposite face area in the i-direction. For the 3D case, this becomes

$$\mathbf{K}_i = \frac{1}{3}\left(\mathbf{A}_x n_x + \mathbf{A}_y n_y + \mathbf{A}_z n_z\right). \tag{4.12}$$

The partial derivative of each force vector by the Roe parameter vector produces a fifth order square matrix, which combine to give the K matrix. To find the exact form of each Jacobian, I first take the flux and state terms as functions of the Roe parameters, where, for example, the state and x-component of the flux are

$$\mathbf{Q} = \begin{pmatrix} Z_1^2 \\ Z_1 Z_2 \\ Z_1 Z_3 \\ Z_1 Z_4 \\ \frac{Z_1 Z_4}{\gamma} + \frac{\gamma-1}{2\gamma}(Z_2^2 + Z_3^2 + Z_4^2) \end{pmatrix}, \quad \mathbf{F}_x = \begin{pmatrix} Z_1 Z_2 \\ \frac{\gamma-1}{\gamma} Z_1 Z_4 - \frac{\gamma-1}{2\gamma}\left(Z_2^2 + Z_3^2 + Z_4^2\right) \\ Z_2 Z_3 \\ Z_2 Z_4 \\ Z_2 Z_5 \end{pmatrix}. \tag{4.13}$$

The y and z flux vectors have equivalent forms, with the more complex momentum term moved appropriately. The elements of each Jacobian are found individually by solving the appropriate differential. For instance, the first row of elements of

the x-Jacobian is

$$
\begin{aligned}
A_{x,11} &= \frac{\partial F_{x,1}}{\partial Q_1} = \frac{\partial}{\partial (Z_1^2)} Z_1 Z_2 = 0 \\
A_{x,12} &= \frac{\partial F_{x,1}}{\partial Q_2} = \frac{\partial}{\partial (Z_1 Z_2)} Z_1 Z_2 = 1 \\
A_{x,13} &= \frac{\partial F_{x,1}}{\partial Q_3} = \frac{\partial}{\partial (Z_1 Z_3)} Z_1 Z_2 = 0 \\
A_{x,14} &= \frac{\partial F_{x,1}}{\partial Q_4} = \frac{\partial}{\partial (Z_1 Z_4)} Z_1 Z_2 = 0 \\
A_{x,15} &= \frac{\partial F_{x,1}}{\partial Q_5} = 0
\end{aligned}
$$

The other elements of the x-Jacobian are found in a similar manner, and again for the y and z Jacobians. Combined with the components of the normal, the matrix takes the form

$$
\mathbf{K}_i = \begin{pmatrix}
0 & n_x & n_y & n_z & 0 \\
\alpha n_x - v_x \Omega & \Omega - \gamma_2 v_x n_x & v_x n_y - \gamma_1 v_y n_x & v_x n_z - \gamma_1 v_z n_x & \gamma_1 n_x \\
\alpha n_y - v_y \Omega & v_y n_x - \gamma_1 v_y n_x & \Omega - \gamma_2 v_y n_y & v_y n_z - \gamma_1 v_z n_y & \gamma_1 n_y \\
\alpha n_z - v_z \Omega & v_z n_x - \gamma_1 v_z n_x & v_z n_y - \gamma_1 v_z n_y & \Omega - \gamma_2 v_z n_z & \gamma_1 n_z \\
(\alpha - H)\Omega & H n_x - \gamma_1 v_x \Omega & H n_y - \gamma_1 v_y \Omega & H n_z - \gamma_1 v_z \Omega & \gamma \Omega
\end{pmatrix}, \quad (4.14)
$$

where $\alpha = \gamma_1 (v_x^2 + v_y^2 + v_z^2)/2$ and $\Omega = v_x n_x + v_y n_y + v_z nz$. As with the two dimensional case, this matrix must be decomposed into its eigenvalues and eigenvectors to produce the Schur decomposition, such that $\mathbf{K}_i = \mathbf{R}^{-1} \mathbf{\Lambda} \mathbf{R}$. The product of these matrices provides the final form of $\mathbf{K}_i$. The form must include the eigenvalues as variables, otherwise, it is not possible to calculate the positive and negative inflow parameters, $\mathbf{K}_i^+$ and $\mathbf{K}_i^-$, which are found by using only positive or negative eigenvalues.

Decomposing this 5×5 matrix into its eigenvalues and eigenvectors is a complex task, with the multiple terms in each element magnifying the problem. All that is required, however, is the final form of the K-matrix. The exact form of the eigenvectors is not required. It is therefore possible to extrapolate the 3D form from its 2D counterpart. It is possible to then simply check this is consistent with the derived form of base K-matrix, which is not explicitly dependent on its eigenvalues.

There is a clear pattern in the final form of the 2D K-matrix. Since the matrix is found by differentiating the flux by the fluid variables themselves, the rows and columns effectively represent the interaction of the equations with one another. Each row corresponds to the flux of a given conserved variable. The first row is the mass flux, with each column representing the effect of each fluid parameter on this flux. Similarly, the second and third rows represent the effect of each conserved quantity on the flux of x and y momentum, and the fourth on energy. The importance of this is that the middle two rows and columns represent the interaction of the momentum terms with each other. In 2D, K_{22} is the x-momentum working on the flux of the x-momentum, K_{23} is the flux of the x-momentum, and so on. From this, it is clear that element is only ever going to represent the interaction of one dimension with one other dimension (or the dimension with itself). A single element does not need to represent the interaction of one dimension with two others. Extrapolating the 2D form to 3D is trivial for the mass flux and energy flux rows, as the terms follow an obvious pattern. Therefore, in 3D, K_{23} is the interaction of the x-momentum flux and the z-momentum, K_{33} is the interaction of the y-momentum flux and the z-momentum, etc. It is therefore possible to follow the pattern of velocities and normals in the 2D form to predict a form for z-momentum row and column.

I extrapolated this final form of $\mathbf{K}_i$ to be

$$
\begin{aligned}
K_{11} &= \frac{\alpha_c}{c}\lambda_{123} - \frac{\Omega}{c}\lambda_{12} + \lambda_3, \\
K_{12} &= -\frac{\gamma_1 v_{xc}}{c}\lambda_{123} + \frac{n_x}{c}\lambda_{12}, \\
K_{13} &= -\frac{\gamma_1 v_{yc}}{c}\lambda_{123} + \frac{n_y}{c}\lambda_{12}, \\
K_{14} &= -\frac{\gamma_1 v_{zc}}{c}\lambda_{123} + \frac{n_z}{c}\lambda_{12}, \\
K_{15} &= \frac{\gamma_1}{c^2}\lambda_{123},
\end{aligned}
$$

$$
\begin{aligned}
K_{21} &= (\alpha_c v_{xc} - \Omega n_x)\lambda_{123} + (\alpha_c n_x - v_{xc}\Omega)\lambda_{12},\\
K_{22} &= (n_x^2 - \gamma_1 v_{xc}^2)\lambda_{123} - \gamma_2 v_{xc} n_x \lambda_{12} + \lambda_3,\\
K_{23} &= (n_x n_y - \gamma_1 v_{xc} v_{yc})\lambda_{123} + (v_{xc} n_y - \gamma_1 v_{yc} n_x)\lambda_{12},\\
K_{24} &= (n_x n_z - \gamma_1 v_{xc} v_{zc})\lambda_{123} + (v_{xc} n_z - \gamma_1 v_{zc} n_x)\lambda_{12},\\
K_{25} &= \frac{\gamma_1 v_{xc}}{c}\lambda_{123} + \frac{\gamma_1 n_x}{c}\lambda_{12},
\end{aligned}
$$

$$
\begin{aligned}
K_{31} &= (\alpha_c v_{yc} - \Omega n_y)\lambda_{123} + (\alpha_c n_y - v_{yc}\Omega)\lambda_{12},\\
K_{32} &= (n_x n_y - \gamma_1 v_{xc} v_{yc})\lambda_{123} + (v_{yc} n_x - \gamma_1 v_{xc} n_y)\lambda_{12},\\
K_{33} &= (n_y^2 - \gamma_1 v_{yc}^2)\lambda_{123} - \gamma_2 v_{yc} n_y \lambda_{12} + \lambda_3,\\
K_{34} &= (n_y n_z - \gamma_1 v_{yc} v_{zc})\lambda_{123} + (v_{yc} n_z - \gamma_1 v_{zc} n_y)\lambda_{12},\\
K_{35} &= \frac{\gamma_1 v_{yc}}{c}\lambda_{123} + \frac{\gamma_1 n_y}{c}\lambda_{12},
\end{aligned}
$$

$$
\begin{aligned}
K_{41} &= (\alpha_c v_{zc} - \Omega n_z)\lambda_{123} + (\alpha_c n_z - v_{zc}\Omega)\lambda_{12},\\
K_{42} &= (n_x n_z - \gamma_1 v_{xc} v_{zc})\lambda_{123} + (v_{zc} n_x - \gamma_1 v_{xc} n_z)\lambda_{12},\\
K_{43} &= (n_y n_z - \gamma_1 v_{yc} v_{zc})\lambda_{123} + (v_{zc} n_y - \gamma_1 v_{yc} n_z)\lambda_{12},\\
K_{44} &= (n_z^2 - \gamma_1 v_{zc}^2)\lambda_{123} - \gamma_2 v_{zc} n_z \lambda_{12} + \lambda_3,\\
K_{45} &= \frac{\gamma_1 v_{zc}}{c}\lambda_{123} + \frac{\gamma_1 n_z}{c}\lambda_{12},
\end{aligned}
$$

$$
\begin{aligned}
K_{51} &= (\alpha_c H_c - \Omega^2)\lambda_{123} + \Omega(\alpha_c - H_c)\lambda_{12},\\
K_{52} &= (\Omega n_x - v_x - \alpha_c v_{xc})\lambda_{123} + (H_c n_x - \gamma_1 v_{xc}\Omega)\lambda_{12},\\
K_{53} &= (\Omega n_y - v_y - \alpha_c v_{yc})\lambda_{123} + (H_c n_y - \gamma_1 v_{yc}\Omega)\lambda_{12},\\
K_{54} &= (\Omega n_z - v_z - \alpha_c v_{zc})\lambda_{123} + (H_c n_z - \gamma_1 v_{zc}\Omega)\lambda_{12},\\
K_{55} &= \frac{\gamma_1 H_c}{c}\lambda_{123} + \frac{\gamma_1 \Omega}{c}\lambda_{12} + \lambda_3,
\end{aligned}
$$

where $X_c \equiv X/c$. The new λ terms have the same meaning as in the 2D case, with $\lambda_{123} = (\lambda_1 + \lambda_2 - 2\lambda_3)/2$ and $\lambda_{12} = (\lambda_1 - \lambda_2)/2$. To find the corresponding $\mathbf{K}_i^+$

and $\mathbf{K}_i^-$, one simply uses only the positive λ^+ or negative λ^- eigenvalues, where

$$\lambda_i^+ = \max(0, \lambda_i) \quad \text{and} \quad \lambda_i^- = \min(0, \lambda_i). \tag{4.15}$$

The validity of this prediction for the form can be determined by two checks. First, it must be consistent with the derived form of the 3D K-matrix, shown in Equation (4.14), when the substitutions for the eigenvalues are made. The eigenvalues are analogous to those in the 2D case, with $\lambda_1 = \Omega + c$, $\lambda_2 = \Omega - c$ and $\lambda_3 = \lambda_4 = \lambda_5 = \Omega$. This is straight forward for all terms. For instance, the first row becomes

$$\begin{aligned}
K_{11} &= \frac{\alpha_c}{c}\lambda_{123} - \frac{\Omega}{c}\lambda_{12} + \lambda_3 = \frac{\alpha}{2c^2}(0) - \frac{\Omega}{2c}(2c) + \Omega = -\Omega + \Omega = 0, \\
K_{12} &= -\frac{\gamma_1 v_{xc}}{c}\lambda_{123} + \frac{n_x}{c}\lambda_{12} = -\frac{\gamma_1 v_{xc}}{2c}(0) + \frac{n_x}{2c}(2c) = n_x, \\
K_{13} &= -\frac{\gamma_1 v_{yc}}{c}\lambda_{123} + \frac{n_y}{c}\lambda_{12} = -\frac{\gamma_1 v_{yc}}{2c}(0) + \frac{n_y}{2c}(2c) = n_y, \\
K_{14} &= -\frac{\gamma_1 v_{zc}}{c}\lambda_{123} + \frac{n_z}{c}\lambda_{12} = -\frac{\gamma_1 v_{zc}}{2c}(0) + \frac{n_z}{2c}(2c) = n_z, \\
K_{15} &= \frac{\gamma_1}{c^2}\lambda_{123} = \frac{\gamma_1}{2c^2}(0) = 0.
\end{aligned}$$

These match the derived form, and so the predicted form is consistent with the original. The same can be shown for all terms. The second condition is that the 3D form return the 2D form when all effects from flows in the z-direction are disallowed. When one sets all terms in the fourth row and column to zero, then there is no longer any signal from the z-velocity and component of the normal. When the matrix multiplication is made, the z-momentum residual will always be zero, and so resultant updates will be identical to the result from the 2D case. Together these two conditions show that the extrapolated form of the K-matrix is acceptable. The details of the implementation are given in Appendix A.

4.3.3 Time Step Limitation

The calculation of the required time step proceeds largely as before, except now the estimate is of the time taken for information to travel across a volume, rather

than an area. It therefore takes the form

$$\Delta t \leq \min_{i \in \mathcal{T}} \frac{3|V_i|}{\sum_{T|i \in T} A^T_{\max} \lambda^T_{\max}}, \tag{4.16}$$

where $A^T_{\max}$ is the largest face of the tetrahedral element, and $|V_i|$ is the dual volume of the vertex i. The maximum information speed $\lambda^T_{\max}$ has the same definition as before. Once again this is an estimate of the numerical physical domain of dependence, so a fraction of the derived time step will be used in practice.

4.3.4 1D Tests

I show here a small number of the basic 1D tests, where there is no variation in the additional dimensions, in the initial conditions. These will show the consistency between new 3D implementation and the results from the 2D version. The results can be directly compared when using equivalent uniform particle distributions, such as the Cartesian grid, where the 3D grid will simply be multiple layers of the same underlying 2D grid.

Gaussian Pulse

As before, I test the ability of the solvers to handle advection by modelling the propagation of a Gaussian density profile in one dimension. The initial conditions are, in essence, identical to those described for the equivalent test in Section 3.5.1, as there is no variation in either the y or z directions. In Figure 4.10, I show the results form this setup for three times, the initial conditions (blue), $t = 0.1s$ (orange), and $t = 0.2s$ (green). The top left panel shows results for the LDA1 solver, the top right N1, bottom left LDA2, and bottom right N2. A key take away from all solvers is the close match between the results from the 2D (crosses) and 3D solvers (dots), demonstrating that the 3D extension can recover results from the original. The only real difference is in the LDA1 and LDA2 results, where the 2D results appear to have slightly more numerical diffusion. The addition of the third dimension, has reduced the numerical diffusion for these solvers. It is not particularly clear why this is the case, and the difference is small.

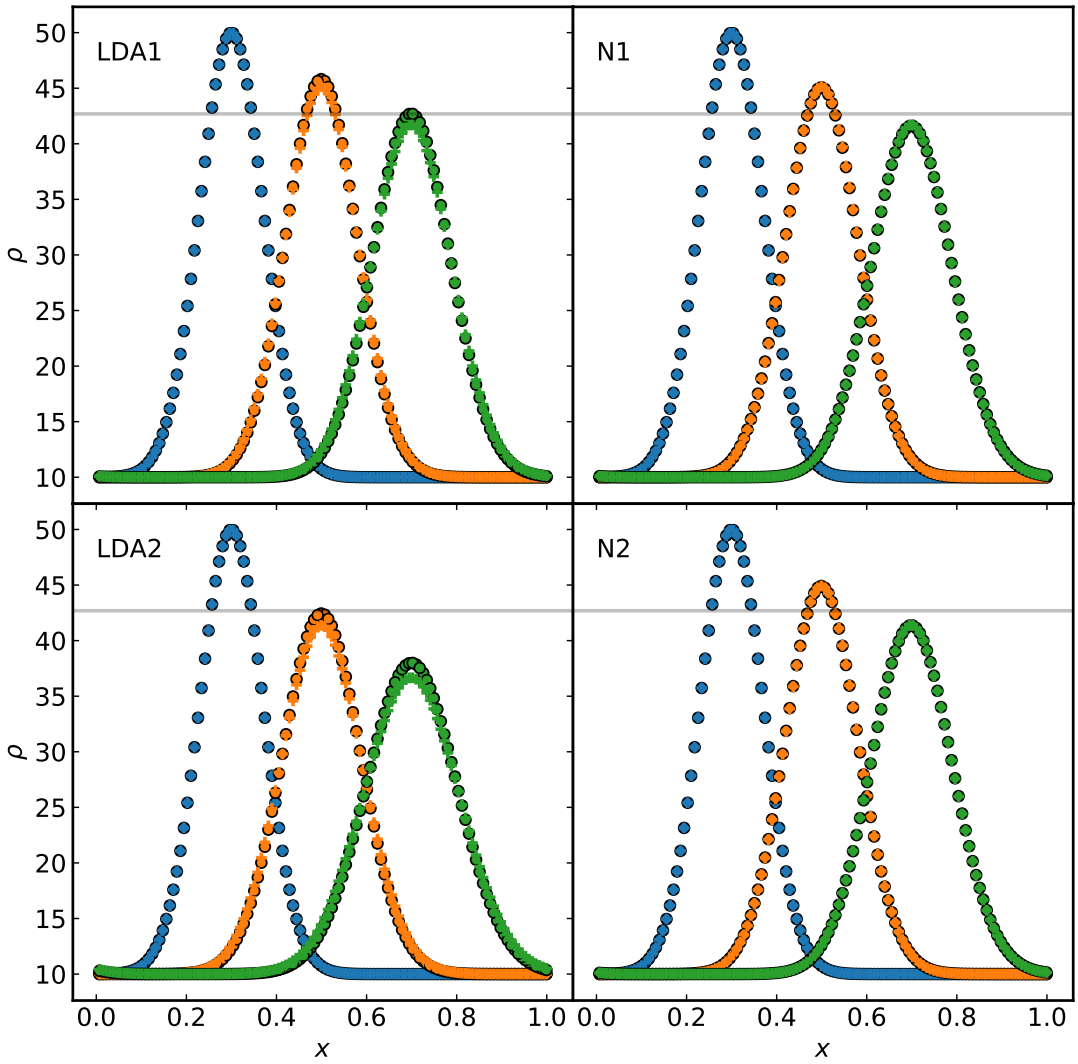

Figure 4.10 *Propagation of a one dimensional Gaussian density pulse for LDA1 (top left), N1 (top right), LDA2 (bottom left), and N2 (bottom right). The dots show results for the 3D solver, while the crosses represent results from the 2D solver. Blue points show the initial conditions, orange show the results at* $t = 0.1s$*, and green at* $t = 0.2s$*. The grey line shows the peak in the LDA1 3D results at the final time. The other solvers have significantly more numerical dissipation, particularly the LDA2 solver.*

Sod Shock Tube

Again, I am essentially checking that the 3D extension produces results consistent with its 2D counterpart. This time I test shock handling with the classic hydrodynamics test of the Sod shock tube, with identical initial conditions to the 2D version (see Section 3.5.1). I compare the results from the 3D solver directly to the 2D equivalent in Figure 4.11. These results are found using $N = 128$ vertices in the x-direction. The density, velocity and pressure results all match very closely between the two possible number of dimension. With the only slight difference coming in the 3D LDA1 solver producing an ever so much steeper gradient in the velocity front. The difference is tiny, and likely caused by the small reduction in numerical diffusion observed in the previous section. As before, we see that the LDA solver is less diffusive than the N-scheme, producing slightly less smoothing of the transitions.

Together the Gaussian pulse and Sod shock tube show that the 3D extension is the correct extension of its 2D base, and that it will recover the same results when additional dimensions are suppressed. This behaviour is present for both simple advection and shocks, covering the basic behaviours of hydrodynamics flows.

4.3.5 3D Tests

Here I show results for full 3D tests. Many have 2D counterparts, such as the Sedov blast, and the Blob test, but are separate from them due to their flows in all three dimensions, even if there are symmetries that mean they are very similar, such as the spherical Sedov blast. These tests go beyond showing that the 3D solver can recreate the results from the 2D solver, demonstrating how it handles flows in all three dimensions. Plots, such as those of density distribution, that show distributions in two dimensions as heat maps, will now be done using contour plots. The 2D solver results were shown using the dual cells that form a fundamental part of the RD method. Constructing a 2D Voronoi tessellation from a plane within a 3D distribution could lead to misleading cell shapes, and potentially cause confusion about the underlying triangulation used to calculate the evolution of the fluid. Instead, the colour contour plots will approximate the field, without this downside.

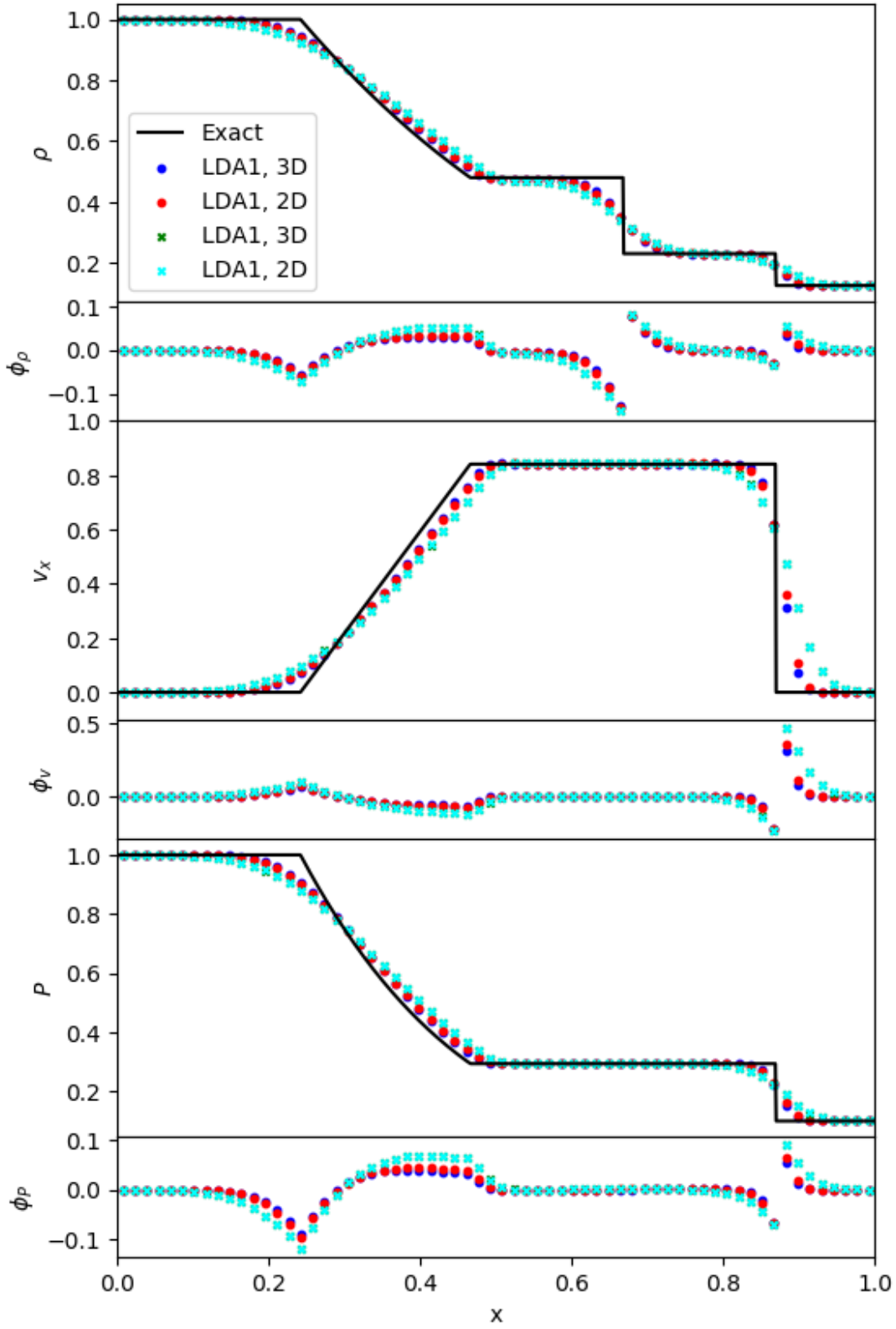

Figure 4.11 *Sod shock tube results, using $N = 128$ in the x-direction, for 2D and 3D solvers. From top to bottom density, x-velocity, and pressure. Difference between numerical and exact results shown below each panel. Blue dots show results for the 3D LDA1 solver, red dots for 2D. For the N1 solver, green crosses show 3D solver, cyan show 2D results. The black line is the exact solution.*

Sedov Blast

The Sedov blast, previously introduced in Section 3.5.2, models the evolution of a point like explosion. As before, the initial condition for this test include a uniform density static background medium, into which energy is injected by setting the pressure of a small region extremely high. This energy is injected at the centre of the box, in this case within $r = 0.2$m, where r is the spherical radius. The box is a cube of size $l_{\text{box}} = 10$m. Below I show results for three resolutions, with slices through the centre of the box in all three dimensions. These results are produced using the LDA1, N1, and B1 solvers. Here, B1 refers to the first order blended scheme, here using the blending described in Section 3.2.2. This mechanism combines the residuals from both LDA1 and N1 methods, weighting the contribution by the fluid state across the triangle. In situations where the N1 solver is generally better suited, that residual is weighted more heavily, and the LDA1 residual less heavily, and vice versa. The weighting is calculated as

$$\phi_i^B = \Theta\phi_i^N + (I - \Theta)\phi_i^{LDA}, \tag{4.17}$$

where I is the identity matrix, and Θ is the blending matrix. The nodal residuals ϕ_i^N and ϕ_i^{LDA} are calculated using the N and LDA schemes respectively. This blending matrix is diagonal, and in this case is calculated with

$$\Theta_{ii} = \frac{|\phi_i^T|}{\sum_j = 1^4 |\phi_{j,i}^N|}. \tag{4.18}$$

The summation in the denominator is performed over all vertices of the element, and i refers to the ith equation of the system.

In Figure 4.12, I show the $N = 32^3$ results, with the left column showing the LDA1 results, the middle column the results for N1, and the right for the blended method B1. The top row is the X-Y plane through the centre of the box, while the middle row is the X-Z plane, and the bottom the Y-Z plane. Even at this very low resolution, the blast wave is very regular, with only small local aberrations, such as at $(2, 6)$ in the middle LDA1 plot. Since the initial injection only encompasses seven vertices, this regularity is a strong result. The low resolution produces a low peak in the blast profile, as the wave is smoothed over several large cells. There is some variation between the solvers, with the LDA1 solver producing more variation within the swept out region, as well as in the front itself. The N1 and B1 results are almost identical. The conditions around the shock front favour

the N scheme over the LDA, in the blending, since there is strong variation across short distances, which explains the similarities.

Increasing the resolution to $N = 64^3$, shown in Figure 4.13, demonstrates the improvement in recovering the results at higher resolution. Higher resolutions still will only be available once further optimisation and parallelisation work has been performed. At present they would require excessive computational walltimes to complete. As before in the 2D case, the peak density increases as the width of the blast front decreases at higher resolution. The shape of the blast becomes increasingly regular as the space is sampled by many more vertices. Once again the LDA1 solver has more internal structure in the blast front and swept region, but the effect is reduced. N1 and B1 are also very similar, for the same reason as before. Recovering the predicted result seems to be most strongly dependent on resolution, with solver choice playing a secondary roll.

These results are best compared to the exact solution with the radial density profile, for results from the various solvers, seen in Figure 4.14. The results are largely identical, with the peak in the LDA1 result being slightly higher than the other solvers. This is likely a result of the lower numerical diffusion for this solver, discussed in Chapter 3. Again, resolution plays a much larger role in recovering the exact solution, than choice of solver. It is also clear, however, that the 3D solver is performing as expected, when compared to both its 2D counterpart, and to the exact solution. The full dimensionality of the problem is well recovered, even at the relatively low resolutions shown here.

Blob Test

As discussed in the previous chapter, the so called 'blob' test (Agertz et al., 2007) characterises the ability of a solver to model the disruption of a spherical cloud of cold gas sitting in a hot flow. To set up this test, a high density static sphere is placed within a low density background, with the high density region being an order of magnitude more dense than the low density medium. The pressure is equalised everywhere, to produce an initial equilibrium, which requires the low density region to be much hotter than the cold cloud. The background medium is given a supersonic initial velocity. I describe the predicted evolution, in detail, in Section 3.5.2. To summarise, Kelvin-Helmholtz instabilities are expected to form where the hot flow is tangential to the cloud, with Rayleigh-Taylor instabilities forming behind the cold cloud, created by the high density material being forced

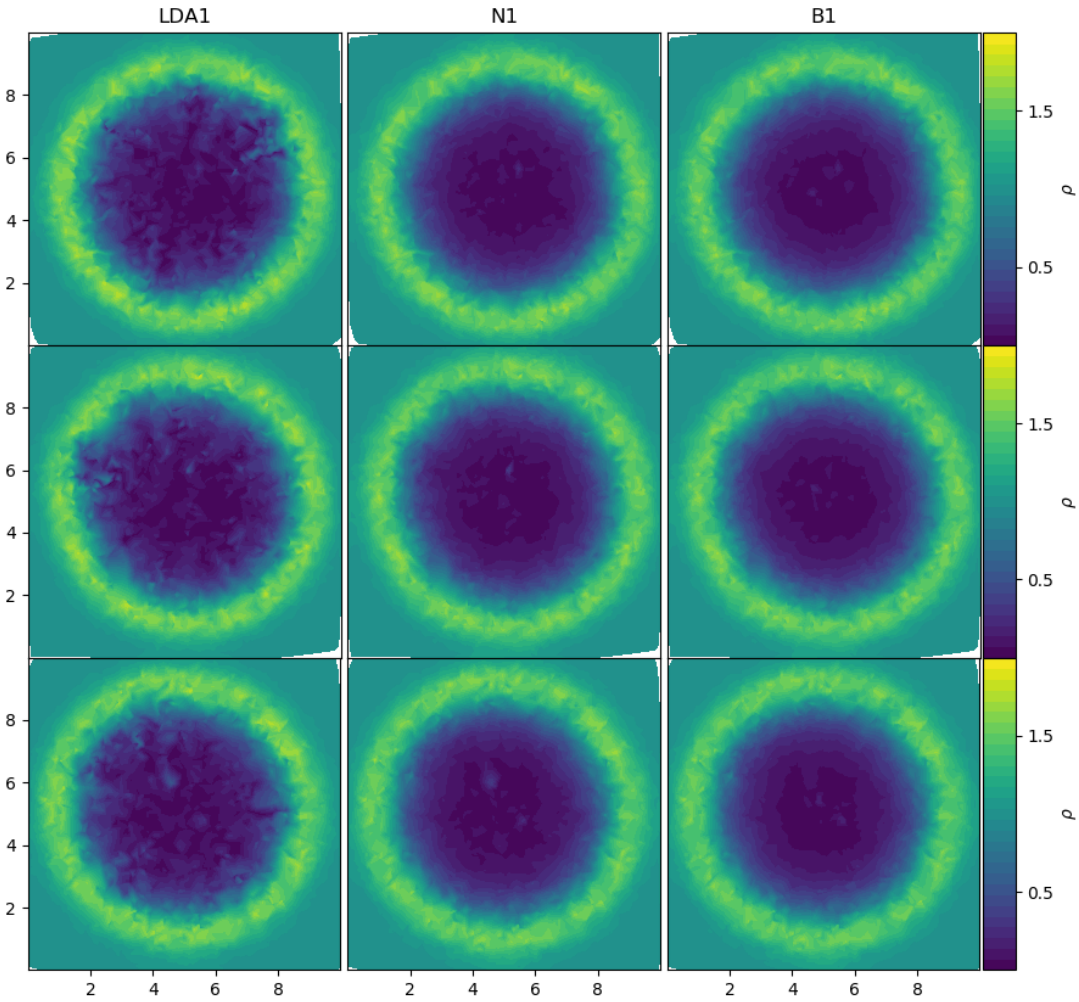

Figure 4.12 *Sedov blast results, with $N = 32^3$ randomly distributed vertices, from the LDA1 (left column), N1 (middle column), and B1 (right column). Each row shows the slice through the centre of the box, with the X-Y plane in the top row, the X-Z plane in the middle row, and Y-Z in the bottom row. The blast wave is remarkably spherical, with an even shape in every dimension, given that the high pressure region used to inject the explosion only included seven vertices.*

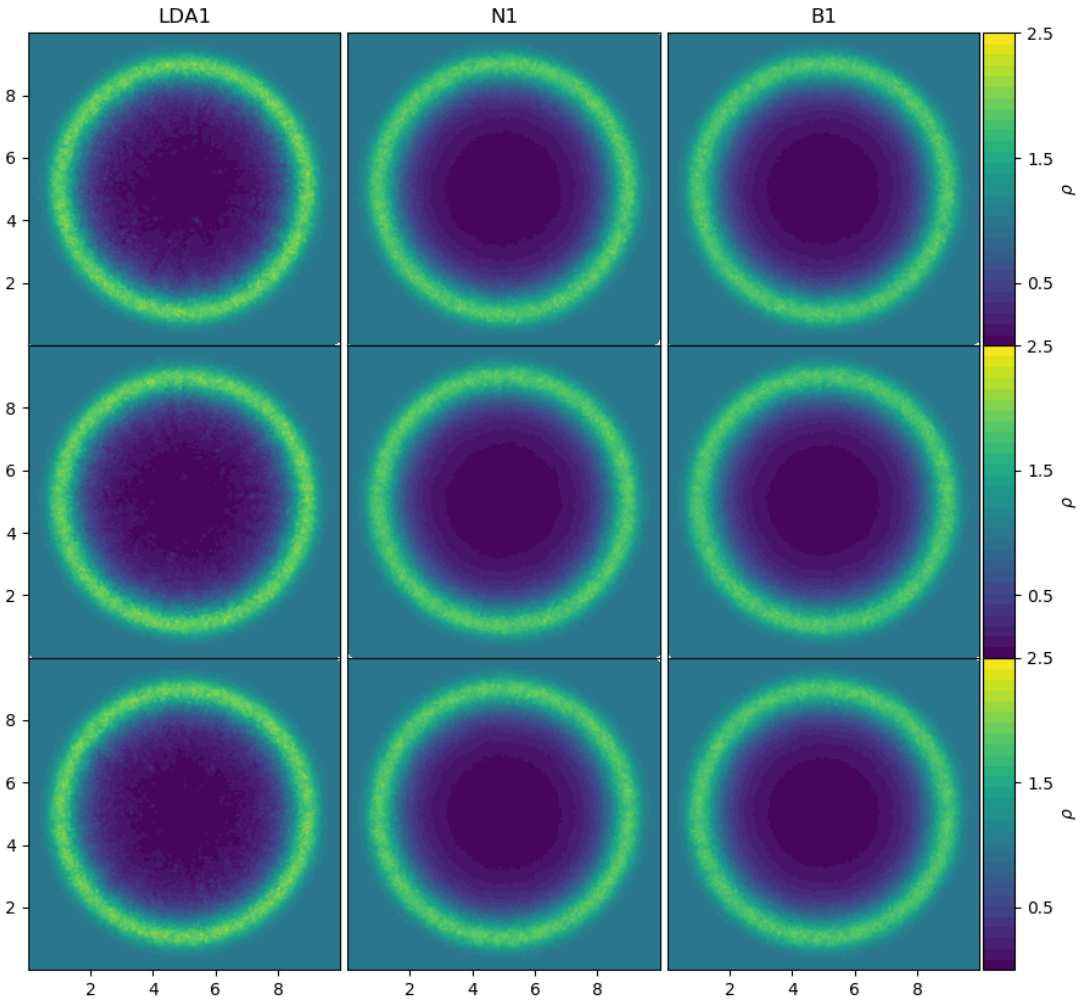

Figure 4.13 *Sedov blast results, with $N = 64^3$ randomly distributed vertices, from the LDA1 (left column), N1 (middle column), and B1 (right column). Each row shows the slice through the centre of the box, with the X-Y plane in the top row, the X-Z plane in the middle row, and Y-Z in the bottom row.*

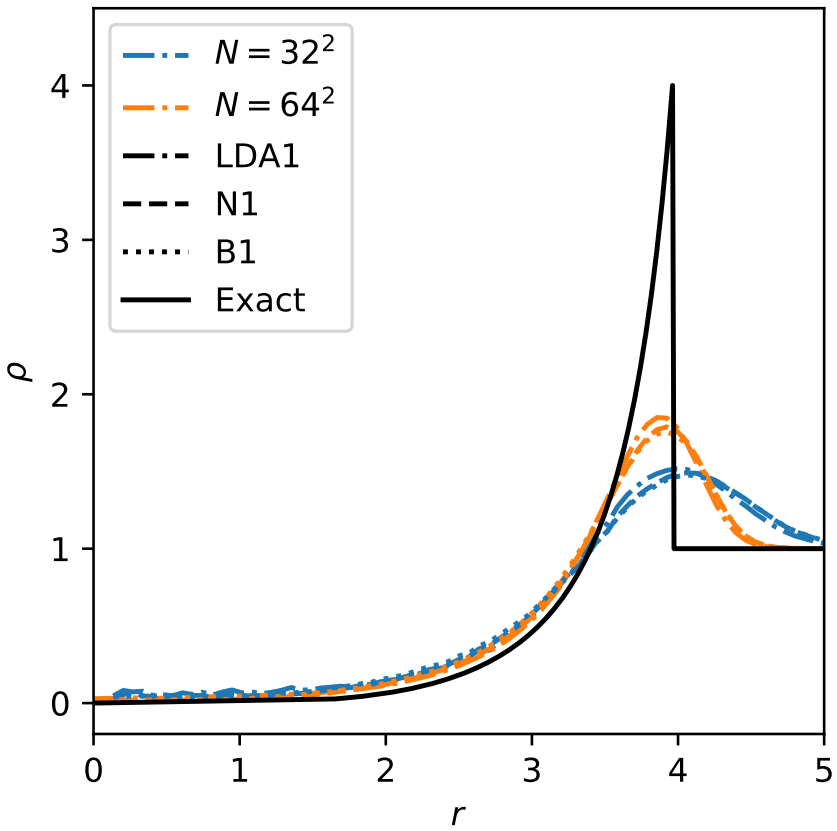

Figure 4.14 *Radial density profile for the Sedov blast, compared to the analytic prediction. The solid black line shows the exact solution, while the dot-dashed line show the results from the LDA1 solver, the dashed from the N1 solver, and the dotted from B1. The blue lines show results from the very low density $N = 32^3$ runs, and orange from $N = 64^3$. The position of the front is recovered reasonably well, though the low resolution smooths the profiles considerably.*

backwards into the lower density material by momentum transfer from the wind.

I choose values of $\rho_{\rm cl}$ = 100kg/m^3, for the cloud, and ρ_0 = 100kg/m^3 for the background. The cloud extend r = 1m from its centre. The box dimensions are 10m × 10m × 10m, and the initial velocity has Mach number $\mathcal{M}$ = 1.5, which corresponds to v_0 = 6.2m/s. This produces a crushing time scale of $\tau_{\rm cr}$ 1s. The crushing time estimates the time for the cold dense cloud to be destroyed by the flow (see Section 3.5.2). I have run a number of such setups, varying resolution and solver type. I compare the predicted evolution to the results from these runs.

In Figure 4.15, I show results for the density distribution, with $N = 32^3$ the N1 (left column), N2 (middle column), and B1 (right column) solvers, at t = 1s = $\tau_{\rm cr}$. As with the Sedov results, each row shows a different plane, through the centre of the cloud. From top to bottom, these are X-Y, X-Z and Y-Z. Unlike the Sedov case, the evolution is not spherically symmetric. The bottom row of panels show the head on view of the cloud, showing the symmetry in the other dimensions. Even at the low resolution, we see the development of the bow wave, and some disruption of the cloud itself. The edges of the cloud being shredded by instabilities, and material from the cloud is accelerated by the wind, as expected. Each solver has very similar results, with the B1 solver showing a slightly more structured bow wave, with lower density cavities behind the wings of the wave.

Increasing the resolution to $N = 64^3$, shown in Figure 4.16, we see a more finely structured bow shock. The head on view shows this clearly. The RT instabilities behind the cloud are more clearly seen here as well, with the greater number of resolution elements recovering the effect in more detail. The higher resolution is also able to recover the low density regions behind the bow wave with much greater detail. However, the resolution is still too low to really compare the evolution to previous blob test results, with the test typically presented with results at $N = 128^3$ and $N = 256^3$. The 3D solver is not currently optimised to cope with these resolutions, but this is planned for the near future.

Instead of increasing resolution, I instead show the time evolution of the results I can produce. In Figure 4.17 I compare the evolution of the density distribution for $N = 32^3$ (left column) and $N = 64^3$ (right column) blob tests. Time increases downwards, with the first row showing the initial conditions at $t = 0$, the second at $t = 0.5\tau_{\rm cr}$, the third at $t = \tau_{\rm cr}$, and the fourth at $t = 1.5\tau_{\rm cr}$. The progressive build up of the bow wave is clear, as is the acceleration of the cloud by the hot flow. The low resolution case shows a greater amount of mixing, between

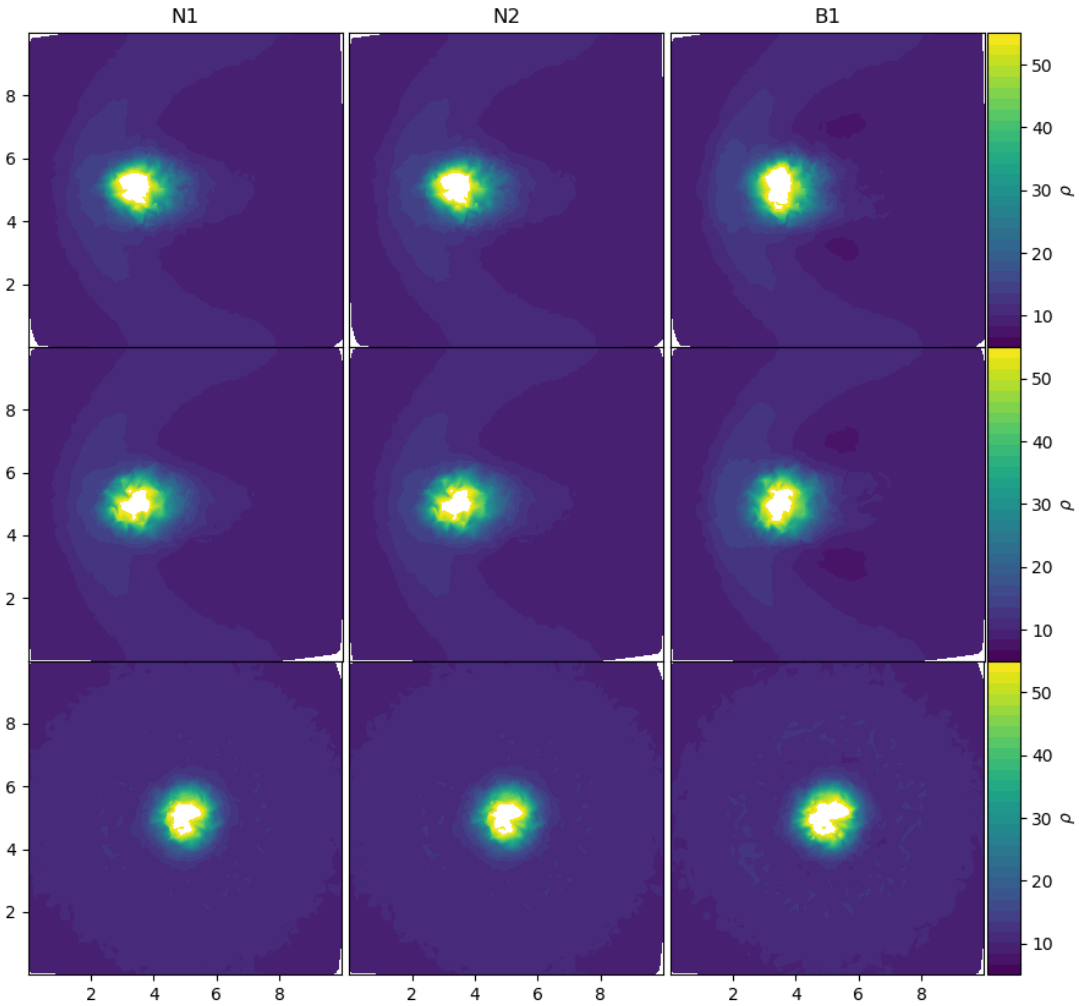

Figure 4.15 *Blob test results, with $N = 32^3$ randomly distributed vertices, from the N1 (left column), N2 (middle column), and B1 (right column) solver. Each row shows the slice through the centre of the blob, with the X-Y plane in the top row, the X-Z plane in the middle row, and Y-Z in the bottom row.*

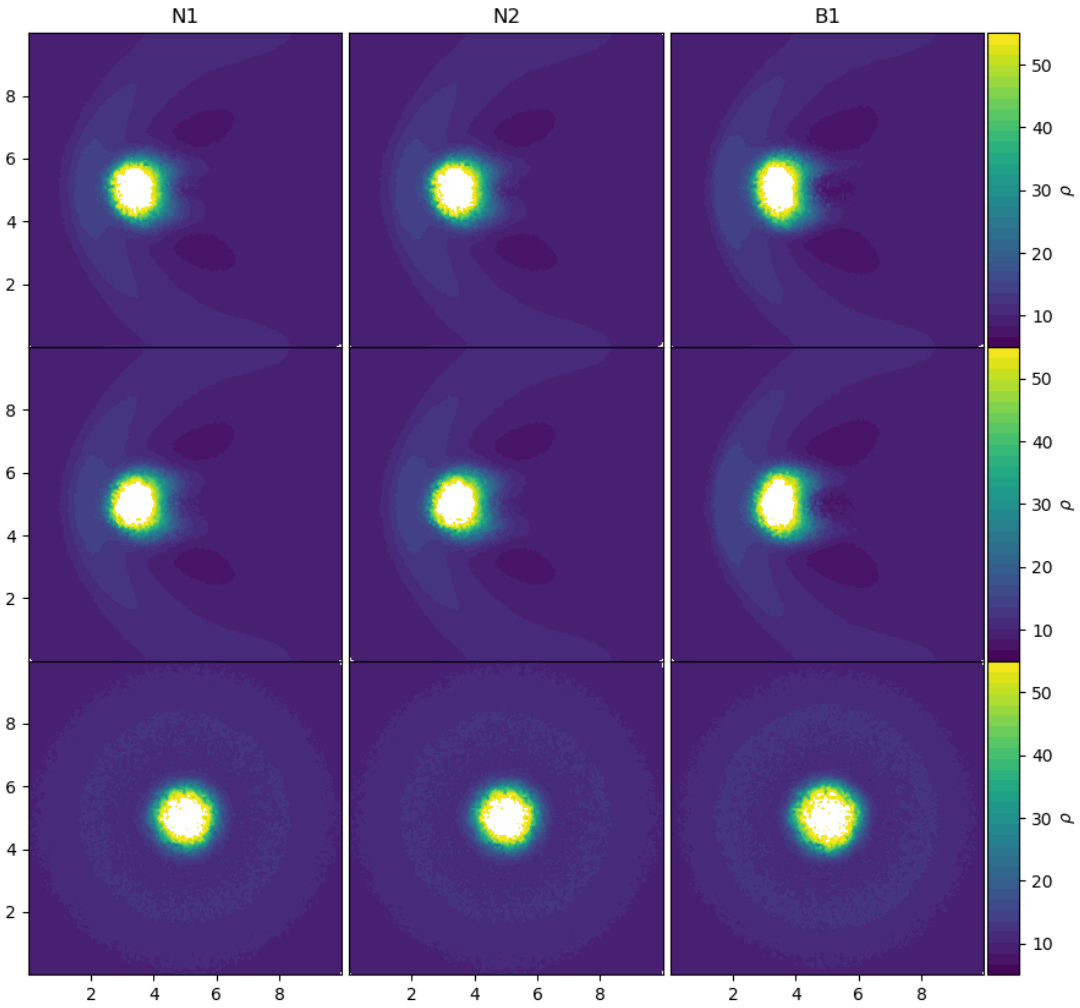

Figure 4.16 *Blob test results, with $N = 64^3$ randomly distributed vertices, from the N1 (left column), N2 (middle column), and B1 (right column) solver. Each row shows the slice through the centre of the blob, with the X-Y plane in the top row, the X-Z plane in the middle row, and Y-Z in the bottom row.*

the cloud and the background, by $t = 1.5\tau_{\rm cr}$, while the higher resolution cloud retains some of its integrity. This is likely a direct effect of the small number of resolution elements that make up the $N = 32^3$ cloud. Once again, it is clear we have not converged with resolution. The fundamental evolution is still present in both cases, showing the solver performs well with these highly multi-dimensional flows, even when only given a small number of vertices with which to work.

This test is fundamentally different from the Sedov blast, in that it is truly three dimensional. While the Sedov blast has flows in all three cardinal directions, it remains spherically symmetric. Therefore the 3D test is not significantly different from its 2D counterpart. The blob test, on the other hand, contains both KH and RT instabilities. A given mode of the KH instability can characterised by only one wave number, and as the vortices form between the shear flows, they rotate on a plane. In 2D, there is only one plane in which they can form, but in 3D there is an additional degree of freedom, complicating the growth of this instability. The RT instabilities are fundamentally 3D, and requiring two wave numbers to characterise a given mode (Agertz et al., 2007). The effect of these differences is that the disruption of the blob will proceed in an inherently different manner in 3D, as opposed to 2D. The complex behaviour and interaction of these instabilities mean the detailed evolution of the blob will be different for the 2D and 3D cases, making it an important case study for the 3D solvers. It is also a structurally complex evolution that is a strong test of solvers in general.

4.4 Gravity

The gravitational interaction of massive objects is clearly a fundamental part of almost all astrophysical systems. In particular, the effect of gravity on baryonic gas drives processes from the formation of the large scale structure of galaxy clusters, filaments and voids, down to the formation of planets. Any method of modelling the hydrodynamics of this gas must be capable of including the effects of gravity, if it is be used to solve astrophysical problems. In Section1.2.4, I covered a number of numerical techniques for efficiently modelling gravitational interaction. These can be used to calculate the force of gravity from a distribution of mass, be it baryonic or otherwise, acting on a given region of gas. However, there remains the question of how exactly this force is combined with residual distribution hydro solver.

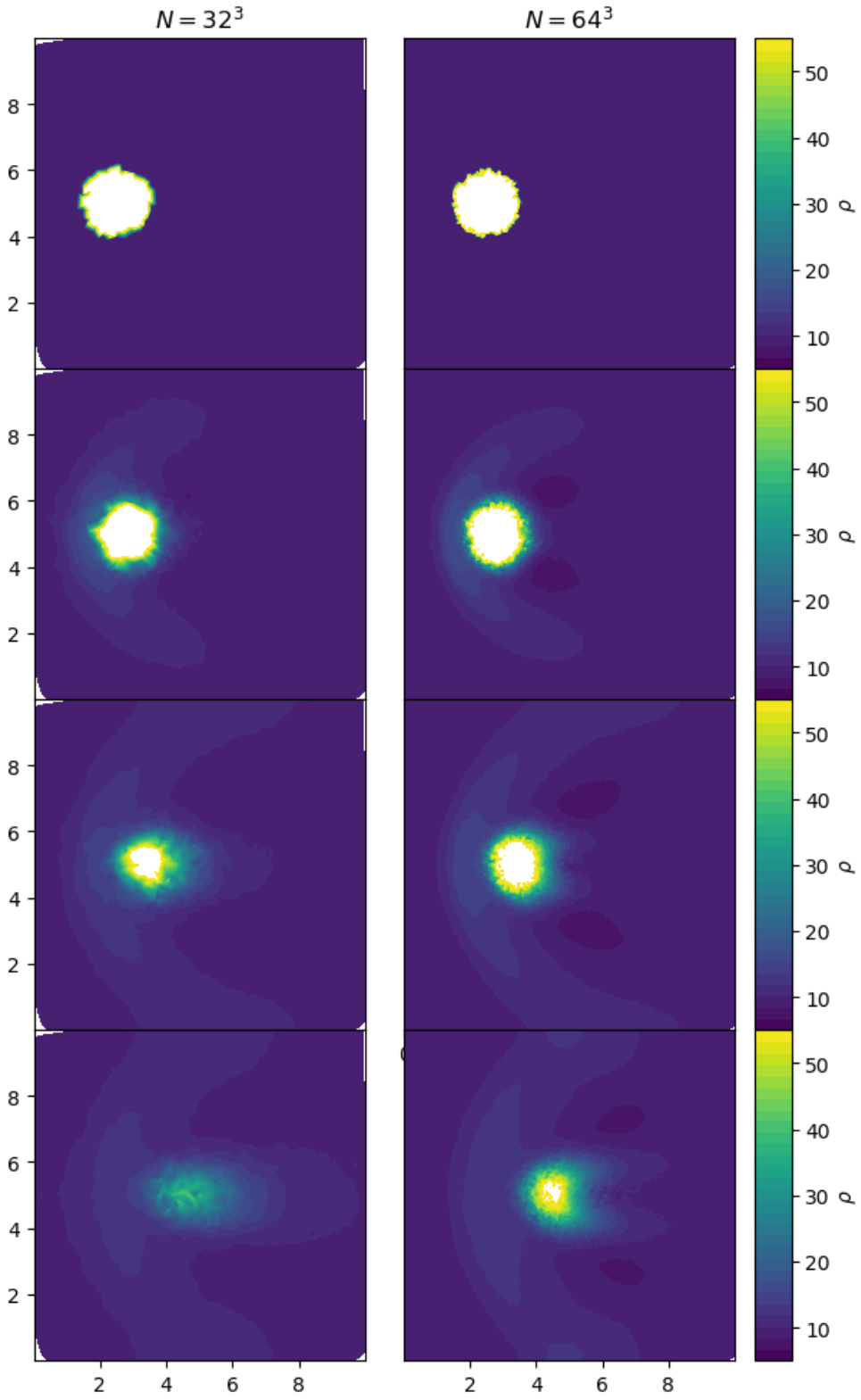

Figure 4.17 *Evolution of the cold gas cloud using the N2 solver, for $N = 32^3$ (left column) and $N = 64^3$ (right column) vertices. The bow wave builds as the hot flow collides with the cloud, with wings extending to the edge of the box. The extent of the disruption depends on the resolution, with the low resolution case struggling to resist breakup and diffusion into the surroundings, while the higher resolution case survives longer.*

In the case of the RD solver, there are two closely linked possibilities, assuming the underlying gravitational potential has already been calculated in a suitable manner. The first option is to embed the effect of gravity in the fluid equations themselves, and then reformulate the calculation of the residual around the new equation. Taking the 2D Euler equations for simplicity, gravity can be included as source terms for momentum and energy. For some gravitational potential Ψ, the gravitational Euler equations can be written in their compact form as

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathcal{F}(\mathbf{Q}) = \mathcal{S}(\mathbf{Q}), \tag{4.19}$$

where $\mathcal{F}(\mathbf{Q}) = (\mathbf{F}_x(\mathbf{Q}), \mathbf{F}_y(\mathbf{Q}))$, and $\mathcal{S}(\mathbf{Q})$ is the source term vector. The explicit forms of the original vectors are

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho E \end{pmatrix}, \quad \mathbf{F}_x(\mathbf{Q}) = \begin{pmatrix} \rho v_x \\ \rho v_x^2 + P \\ \rho v_x v_y \\ \rho v_x H \end{pmatrix}, \quad \mathbf{F}_y(\mathbf{Q}) = \begin{pmatrix} \rho v_y \\ \rho v_x v_y \\ \rho v_y^2 + P \\ \rho v_y H \end{pmatrix}, \tag{4.20}$$

with the gravity source terms given by

$$\mathcal{S}(\mathbf{Q}) = \begin{pmatrix} 0 \\ -\rho \nabla_x \Psi \\ -\rho \nabla_y \Psi \\ -\rho \mathbf{v} \nabla \Psi \end{pmatrix}. \tag{4.21}$$

Here, $\mathbf{v} = (v_x, v_y)$ is the full velocity vector. The additional terms can then be included in the residual calculation, such that the residual is now strictly defined as

$$\phi^T(\mathbf{Q}_h) = \int_T \nabla \cdot \mathcal{F}_h(\mathbf{Q}_h) - \mathcal{S}_h(\mathbf{Q}_h) dx dy. \tag{4.22}$$

Following from this, the numerical approximation of the residual would have to be reformatted to include the source terms. This could possibly involve reconstructing the $\mathbf{K}$-matrix, and decomposing it to find the new eigenvalues, and corresponding eigenvectors. Even if one assumes that the new matrix has a simple set of eigenvalues and eigenvectors, this process is non-trivial algebraic exercise, one which would need to be performed for both the 2D and 3D forms. I am not aware of this being achieved, for gravity, yet.

The second approach follows closely from the first. Instead of reformulating the

whole K-matrix structure, one can simply break the integral of Equation 4.22 in two, leaving the residual as

$$\phi^T(\mathbf{Q}_h) = \phi_f^T + \phi_g^T = \int_T \nabla \cdot \mathcal{F}_h(\mathbf{Q}_h) dxdy - \int_T \mathcal{S}_h(\mathbf{Q}_h) dxdy. \tag{4.23}$$

The residual for the basic fluid equations ϕ_f^T is then calculated in the same manner as before, and the contribution to the complete residual from the gravitational source terms ϕ_g^T is computed separately. I have not determined the exact form the discrete approximation of this residual would take, at this time. This second approach requires significantly less reworking, and provides a framework for the inclusion of other source terms in the future, such as cooling terms, or other sub-grid physics. These can be added as new contributions to the complete residual ϕ^T, without changing the fundamentals of the rest of the method. In the literature, there is only limited discussion of the inclusion of source terms, in general, most notably in Csik et al., 2002, who use a source term in their magneto-hydrodynamics work to maintain the divergence of the magnetic field as zero. Deconinck & Ricchiuto, 2007 also discuss the inclusion of source terms, and utilise this second approach. They state that including source terms in this manner may not maintain the stability of the method.

It is currently unclear whether or not the two methods are in fact exactly mathematically equivalent. At the stage of splitting the integral into two calculations, they are the same, but once the appropriate numerical approximation is applied to both parts, it is possible that the combined version produces systematically different flow. At the moment, I would argue that they are in fact equivalent, and that the second approach is an adequate solution to the inclusion of gravity and other source terms. However, a more thorough investigation is still required.

Dynamical Friction

I have replicated the idealised dynamical friction test, which I proposed in Chapter 2. I recreate the setup used in Section 3.5.2, but now in 3D. As before, I apply an external Plummer potential, to model the gravitational effect of a perturbing mass. The gravitational acceleration is applied using effectively using the second approach described above, with the gravitational effect added independently of the basic residual. This implementation is still being tested, hence the preliminary nature of these results. The test is set up to have $A = 0.1$, at $\mathcal{M} = 1.3$, with

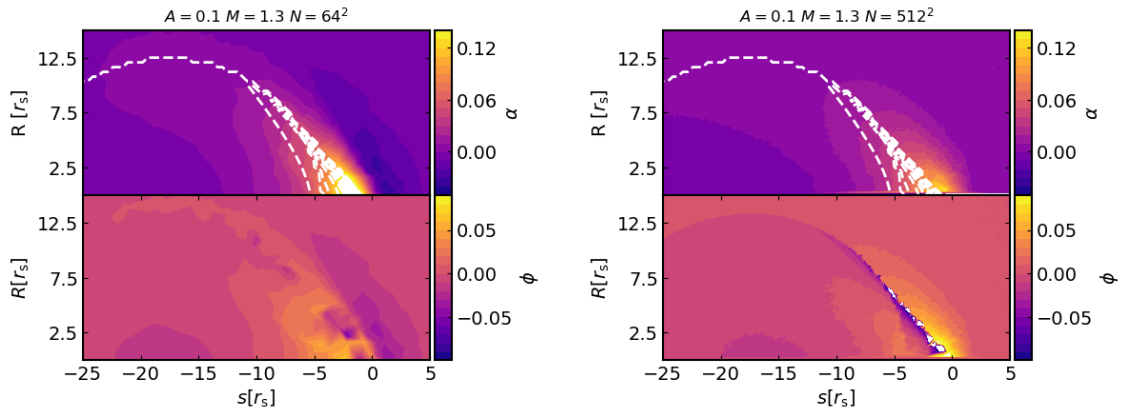

Figure 4.18 *Gravitationally induced wakes from the 2D N1 solver, with* $N = 64^2$ *vertices (left) and the 3D solver (right), using* $N = 64^3$ *vertices. Upper panel: Colours show the over-density* α*, with the analytic prediction shown as white dashed contours. Lower panel: Difference between the numerical and analytic wakes* $\phi = \alpha_{\text{num}} - \alpha_{\text{ana}}$ *is shown below.*

$N = 64^3$ vertices. As a reminder, the A-parameter links the perturber to the fluid state with $A = GM_p/c_s^2 r_s$. The simulation is run for approximately $15t_c$, where t_c is the sound speed c_s crossing time of the softening scale r_s of the Plummer potential.

I show preliminary results for the DF test, using the N1 RD solver, in Figure 4.18. I compare the results from the 2D solver (left) to those from the 3D solver. There are clear disparities between the produced structures. The 2D result has the edge of the cone fitting better with the analytic prediction (shown in both cases as the white contours). In contrast, the 3D case has a wake that is much closer to being spherical, with a less well defined cone structure. It does not contain the trough in front of the cone, seen in the 2D results. These runs are at a low resolution, compared to the full DF analysis I performed in Chapter 2, but this does not explain the significant difference between the two cases, since they share effective resolutions. The initial conditions are identical, except for the obvious change to add the state in the third dimension.

Chapter 5

Dynamical Friction with Cooling

5.1 Introduction

The dynamical friction work, presented in Chapter 2, shows that certain state-of-the-art Lagrangian hydro solvers systematically under-produce the dynamical friction retarding force, when compared to analytic predictions. The situation modelled in these tests is highly idealised, designed to closely replicate the assumptions made in deriving the linear approximation of the solution. Of the assumptions, two critical conditions are that the gas is adiabatic, and that the gas does not experience self gravity (Ostriker, 1999). In other words, there are no additional heating or cooling terms that allow the gas to change temperature, other than the adiabatic contraction caused by the gravity of the perturber, and the only gravitational force that the gas experiences comes from the perturbing body. These assumptions make the derivation of the simplified physical solution possible, but exclude significant physical processes from the model.

Baryonic gas, in many astrophysical environments, is subject to several sources of radiative cooling, whereby the gas radiates energy as photons. If the surrounding gas is transparent to the frequency of these photons, which is true if they have low interaction cross sections in that regime, then the energy can be carried from the region of gas. To understand the temperature evolution of a region of gas, therefore, one must be able to accurately include the appropriate cooling processes. As discussed in Chapter 1.1.3, atomic cooling, metal line cooling and molecular cooling dominate sources of cooling in different conditions. Broadly,

above temperatures of $T = 10^4$K, atomic cooling dominates. In this regime, collisional excitation and ionisation produces a very high ionisation fraction in the gas. Recombination, and spontaneous decay of from excited energy levels to lower ones, produce the photons that can escape, cooling the gas. Collisional excitation can also remove energy from the gas by transferring energy from the kinetic energy of one particle to the excitation of another.

Below this temperature threshold, atomic cooling is not efficient. The excitation energy of hydrogen is greater than the average kinetic energy of collisions. With fewer excitations, atoms cannot radiate energy efficiently, making molecular and metal line cooling the dominant processes. The formation of these molecules is catalysed by the metals present in the gas, as dust formed from these metals provide sites for molecular hydrogen formation (Hollenbach & McKee, 1979; Perets & Biham, 2006). In primordial environments, which are almost entirely metal free, molecules must instead form by direct gas-phase collisions (McDowell, 1961). The formation rate is significantly reduced in such environments, where molecule formation is then strongly dependent on the density of the gas. In the pristine primordial gas, molecules provide the only efficient cooling mechanism below $T = 10^4$K (Omukai et al., 2005), and so are crucial in producing the first stars and galaxies (Abel et al., 2002; Bromm et al., 2009). Molecules cool the gas through the radiative decay of the rotational and vibrational transitions of the molecule. For cooling to occur by this process, the rate of de-excitation of these roto-vibrational modes by collisions, which do not result in cooling, must be lower than the rate of radiative decay (Galli & Palla, 1998).

When there are metal atoms present in the gas, they provide an additional cooling mechanism (Omukai, 2000; Omukai et al., 2005; Smith et al., 2008). Metal line cooling occurs when electrons in excited states in metal atoms radiatively decay to lower energy states. Excitation comes from both absorption of photons and collisions with other components of the gas. These excited states decay spontaneously, or in some specific cases through further collisions, releasing photons, and removing energy from the gas (Omukai, 2000). Metals have complex fine structure in their lines, resulting in a greater range of possible photon wavelengths, and so have many more channels through which cooling can proceed. It takes less energy to excite an electron through a fine line structure transition, as the energy levels are close together. This makes excitation, and so cooling, easier to achieve at low temperatures, when compared to similar excitation transitions in atomic hydrogen.

A massive perturber moving through some low density background will produce an over-dense wake. In the early Universe, as the first stars begin to form, there is already significant amounts of dark matter substructure that can act as such a perturber. These sub-halos move through the extended gaseous structure of their host halo. At this early time, molecular cooling is the only way to remove energy from the gas, and so is the only process that can produce clouds that collapse to form the first stars. The formation of these molecules, in turn, is strongly dependent on the density of the gas. The wakes created by the perturbing substructure, as part of the dynamical friction process, are potential candidates for stimulating the production of molecular hydrogen.

In this chapter, I extend the idealised study of dynamical friction, conducted in Chapter 2, to include cooling effects from molecules and metals. To do this I run a new set of idealised DF simulations, with initial conditions similar to those described previously. On top of the gravitational potential of the perturber, I will now include cooling and molecule formation using the chemical network library `Grackle` (Smith et al., 2016). I run scenarios with a variety of background densities, and with a range of metallicities, while also exploring the same parameter space in Mach number, and A parameter, as before. In Section 5.2, I briefly describe the reactions that `Grackle` models, and how it produces the cooling rates for the simulation. Then, in Section 5.3, I cover the specifics of the simulation setups that include the effects of the various cooling processes. Section 5.4 contains the detailed results from these runs, and Section 5.5 a discussion of the implications to the formation of stars in the early universe, and the evolution of galaxies in that epoch.

5.2 Grackle

The chemistry and cooling library `Grackle` is a well established simulation tool, which provides models for primordial chemistry, radiative heating and cooling, and UV radiation backgrounds (Smith et al., 2016). Primordial chemistry is almost entirely metal free, so it largely consists of hydrogen, helium, and the ionised species of the elements, H^+, He^+, and He^{++}. Alongside these components, the corresponding species associated with molecular hydrogen and deuterium can be included. Each species has a set of reactions that either create that species, or use it as an ingredient to form some other species. For example, Table 5.1 shows the reactions modelled from the `Grackle` nine species mode, where hydrogen,

helium, their ions, molecular hydrogen and electrons are used. For each reaction, a temperature dependent fit is used to set a reaction rate k. The change, with time, in the number density n_i, of the ith species is taken as (Smith et al., 2016)

$$\frac{\partial n_i}{\partial t} = \sum_j \sum_l k_{jl} n_j n_l + \sum_j I_j n_j, \tag{5.1}$$

where k_{jl} is the reaction rate between species j and l. The last term, I_j, is the radiative rate of the jth species. All reactions can be lumped into formation and destruction terms

$$\frac{\partial n_i}{\partial t} = C_i(T, n_j) - D_i(T, n_j) n_i, \tag{5.2}$$

where C_i and D_i are the total creation and destruction rates, with respect to all reactions with species n_j, at a given temperature T and number density of that other species n_j. A backwards difference formula is applied (Anninos & Norman, 1996) to produce the numerical approximation of the solution to this set of partial differential equations. This takes the form

$$n^{t+\Delta t} = \frac{\Delta t C^{t+\Delta t} + n^t}{1 + \Delta t D^{t+\Delta t}}. \tag{5.3}$$

This provides the change in the number density of species i over the time t to $t + \Delta t$. The time step, Δt, is a sub-step within the time step passed by the simulation. It is limited such that the hydrogen and electron abundances change by, at most, ten percent, over one sub-step. The sum of the changes over all sub-steps gives the total change for the global time step. Typically, the global time step is significantly shorter than the time it would take for the reactions to reach equilibrium, so this non-equilibrium approach is valid.

Alongside the calculation of these species abundances, `Grackle` calculates the temperature change from heating and cooling processes within the gas. In a primordial gas, this involves tracking the energy lost due to the state transitions for the various ionic species, including collisional excitation and ionisation, recombination, and free-free emission, but does not include the effects of metal line cooling. Heating processes include Compton scattering and photo-ionisation heating, usually from some UV background radiation field. In cases where the equilibrium solution is required, a simpler method, which interpolates from tabulated cooling rates, is used. This takes a redshift, density and temperature value, and returns cooling/heating rates, which is then converted into a change

Table 5.1 *`Grackle` nine species reaction network which models primordial chemistry (Smith et al., 2016), showing the reactions of neutral hydrogen H and helium He, molecular hydrogen H_2, ionised hydrogen H^+ and helium He^+, the negative ion of hydrogen H^-, doubly ionised helium He^{++}, electrons e^-, and photons γ (Tables 3 and 4 of Smith et al., 2016).*

Reaction		
$H + e^-$	$\rightarrow$	$H^+ + e^- + e^-$
$H^+ + e^-$	$\rightarrow$	$H + \gamma$
$He + e^-$	$\rightarrow$	$He^+ + e^- + e^-$
$He^+ + e^-$	$\rightarrow$	$He + \gamma$
$He^+ + e^-$	$\rightarrow$	$He^{++} + e^- + e^-$
$He^{++} + e^-$	$\rightarrow$	$He^+ + \gamma$
$H + H$	$\rightarrow$	$H^+ + e^- + H$
$H + He$	$\rightarrow$	$H^+ + e^- + He$
$H + \gamma$	$\rightarrow$	$H^+ + e^-$
$He + \gamma$	$\rightarrow$	$He^+ + e^-$
$He^+ + \gamma$	$\rightarrow$	$He^{++} + e^-$
$H + e^-$	$\rightarrow$	$H^- + \gamma$
$H^- + H$	$\rightarrow$	$H_2 + e^-$
$H + H^+$	$\rightarrow$	$H_2^+ + \gamma$
$H_2^+ + H$	$\rightarrow$	$H_2 + H^+$
$H_2 + H^+$	$\rightarrow$	$H_2^+ + H$
$H_2 + e^-$	$\rightarrow$	$H + H + e^-$
$H_2 + H$	$\rightarrow$	$H + H + H$
$H^- + e^-$	$\rightarrow$	$H + e^- + e^-$
$H^- + H$	$\rightarrow$	$H + e^- + H$
$H^- + H^+$	$\rightarrow$	$H + H$
$H^- + H^+$	$\rightarrow$	$H_2^+ + e^-$
$H_2^+ + e^-$	$\rightarrow$	$H + H$
$H_2^+ + H^-$	$\rightarrow$	$H_2 + H$
$H + H + H$	$\rightarrow$	$H_2 + H$
$H + H + H_2$	$\rightarrow$	$H_2 + H_2$
$H^- + \gamma$	$\rightarrow$	$H + e^-$
$H_2^+ + \gamma$	$\rightarrow$	$H + H^+$
$H_2 + \gamma$	$\rightarrow$	$H_2^+ + e^-$
$H_2^+ + \gamma$	$\rightarrow$	$H^+ + H^+ + e^-$
$H_2 + \gamma$	$\rightarrow$	$H + H$

in internal energy.

When metals are present in the gas, a similar tabulated approach is used, as a full reaction network would be computationally prohibitive (Smith et al., 2008). The code takes a single metallicity value, in units of solar metallicity, and assumes solar relative abundances. It can handle situations with or without an assumed UV background. The cooling tables are constructed from photo-ionisation simulations using `Cloudy` (Ferland et al., 2013). Once again, a redshift, density, and temperature, are supplied, and a cooling rate is interpolated from the table of values. The redshift variable allows for the different options in UV background model. The change in temperature of the gas, from metal line cooling, is handled independently of the effects from the chosen chemical network. In other words, the change in temperature from the presence of metal is calculated from the same conditions as the cooling from primordial chemicals, and so, in a given time step, the two processes run in parallel, not directly interacting with each other. The effect of both is only combined in the net change in temperature. The cooling from metal line emission is applied on top of the change found from the chosen reaction network.

5.3 Setup

The idealised setup used in Chapter 2 is replicated closely here. I start with a uniform density gas, moving at a bulk velocity, with a massive perturber at its centre. The mass is modelled with a fixed Plummer potential with total mass M_p. This velocity corresponds to a Mach number $\mathcal{M} = V_0/c_s$, where $c_s = \sqrt{\gamma k_b T_0/\mu_0}$ is the sound speed of the gas, T_0 is the initial temperatures, and μ_0 is the initial mean molecular mass of the gas. In these new runs, the gas is not pure hydrogen, as previously assumed. Instead, the gas consists of a combination of hydrogen, helium, and metals. The initial abundances of these components vary across the set of simulations, with some runs using primordial compositions, and others using solar metallicity, and a number with compositions in between these limits. These differences in composition produce different mean molecular masses in the gas. In Table 5.2, I give the initial conditions and other parameters that define each run. All simulations are run using $N = 512^3$ particles, with initial temperatures $T_0 = 10^4$K, in boxes of edge length $L = 1$kpc. All simulations are runs for $A \approx 0.1$ and $\mathcal{M} = 1.3$, using $r_s = 0.055$kpc. The background density is set to one of three densities, with particle number densities $n = 0.01\text{cm}^{-3}$, $n = 1\text{cm}^{-3}$, or $n = 10\text{cm}^{-3}$.

Table 5.2 *Parameters of each simulation, showing what cooling processes are modelled, the Mach number, number density, hydrogen, helium and metal fractions, and the resulting mean molecular mass. These setups all use* $M_p = 1 \times 10^5 \mathrm{M}_\odot$*, and* $r_s = 0.055 kpc$*.*

Cooling	$\mathcal{M}$	n (cm^{-3})	H	He	Z	Metallicity
None	1.3	0.01	0.76	0.24	0	NA
None	1.3	1	0.76	0.24	0	NA
None	1.3	10	0.76	0.24	0	NA
Metals+Mol	1.3	0.01	0.7381	0.2485	0.0134	Solar
Metals+Mol	1.3	1	0.7381	0.2485	0.0134	Solar
Metals+Mol	1.3	10	0.7381	0.2485	0.0134	Solar
Metals+Mol	1.3	0.01	0.76	0.24	1×10^{-6}	Primordial
Metals+Mol	1.3	1	0.76	0.24	1×10^{-6}	Primordial
Metals+Mol	1.3	10	0.76	0.24	1×10^{-6}	Primordial

Using these three densities, I run three process setups. The first is a benchmark set, where no additional cooling or chemistry processes are used. The second set use the `Grackle` 9-species chemical network to model the additional effects of metal cooling, as well as the formation and effects of molecular hydrogen. These use a gas with solar metallicity. The third set of runs use the same setup as the second, but with a primordial metallicity gas.

5.4 Results

The results presented here are direct equivalents of those presented in the previous chapter on DF. I show the detailed structure of the wake, using the same over-density α as before. I also show the evolution of the net force on the massive perturber as a function of time. Direct comparison to the analytic solution is not as useful here, since we are explicitly including additional physics. Instead, the comparison to the results with no additional chemistry can disentangle what comes from the gravity of the perturber, and what is added by the cooling processes, and how important cooling is for DF. The gravitational evolution of the wake, outside the softening scale, would remain linear in the pure gravity case, but this may no longer hold with the addition of cooling.

5.4.1 Solar Metallicity

Let us first consider the solar metallicity case. I first compare the evolution in the over-density $\alpha = \rho/\rho_0 - 1$ for the three densities. In Figure 5.1, I show the over-density in cylindrical coordinates (s, R), where s is the distance along the direction of travel, and R is the cylindrical radius. The perturber is fixed at $s = 0$, with the wake behind the perturber with $s < 0$. The first column shows the results for the benchmark 'no cooling' runs. From left to right, the other columns show the low ($n = 0.01\text{cm}^{-3}$), medium ($n = 1\text{cm}^{-3}$), and high ($n = 10\text{cm}^{-3}$) density cases. Time increases from top to bottom, with the first row showing results for $t = 4t_c$, the second for $t = 8t_c$, and the third for $t = 12t_c$. The sound speed crossing time of the softening scale t_c, is approximately $t = 5Myrs$, for this setup. In the first column, the white contours show the analytic solution, and the lower part of each panel shows the difference between the numerical and analytic results $\phi = \alpha_{\text{num}} - \alpha_{\text{ana}}$. In the other columns, the lower part of each panel shows the difference between that result and the corresponding 'no cooling' result. The simulations are run far beyond the time when the wake reaches the edge of the box, so in the medium and high density cases we clearly see the wake has wrapped around, through the periodic boundaries. This gives us insight into the long term evolution of the temperature, while still maintaining very high mass resolution. A larger box would reduce this resolution.

The 'no cooling' result is essentially identical to the previous DF results for $A = 0.1$ and $\mathcal{M} = 1.3$. Since the problem, in that setup, is essentially scale free, this is to be expected. When we move to the low density results, we see that there is a small deviation from the previous result. The wake in the cone is contained more closely to the expected cone region, demonstrated by the lower alpha beyond the cone edge. This is most clearly seen in the bottom panel of each plot, at this density. The dark region ahead of the cone edge shows α values lower than the 'no cooling' case. The over-density inside the cone region is also slightly higher. This shows that the collapse of material into the over-density is more pronounced when cooling is allowed. The effect is only small in this case. Looking now at the medium density case, the increased concentration of material is much more pronounced. The wake has lost almost all of its detailed structure, with no obvious cone or sphere structures. Instead, the wake forms a dense tail directly behind the perturber, with only small lateral extent.

To understand what is happening to the density, in the different scenarios, it

is useful to look at the distribution of temperatures. In Figure 5.2, I show the temperature across the simulated region, using the same cylindrical coordinate scheme. Each panel show results from the same snapshot as the previous figure. The 'no cooling' results in the first column show the difference between the initial temperature and the final temperature in the lower part. The other runs show the difference between the result for that density/time, and the 'no cooling' result. The temperature change in the first column is entirely positive, as material is heated by its adiabatic collapse. We see from the time evolution that this heating happens before the first snapshot, as the dense tip of the cone is forming. Since cooling is explicitly ignored in this case, this is expected. In the low density case we see a more complex picture. The unperturbed regions, outside the over-dense wake, cool by a consistent amount, producing a uniform background temperature. However, the same heating, created by the collapse of the gas, is still present. The densest part of the wake is hotter, by several hundred Kelvin. This trend continues at later times, with the dense cone edge retaining a temperature higher than the background. It can also be seen, in the lower part of the later time plots, that the spherical part of the wake maintains its shape, with a slightly higher temperature than the background.

Moving to the medium density case, the cooling effect is much greater. The temperature of the background drops to close $T = 12$K by the first snapshot, and by the final time, has dropped to $T = 4$K. While the dense part of the wake is still slightly warmer than the background, the difference is now small. It is clear that the increased density produces a much greater loss of heat. As material forms the wake, it would previously have become pressure supported, heated by its collapse. In this case, the pressure support is almost entirely removed, with much of the heat generated by the collapse radiated away. The heating is overcome by the extreme cooling, and so material collapses to much greater densities. This combination is also seen in the high density case, where cooling is even more extreme. The background temperature has dropped to $T = 2.7$K by the end of the run. This is a hard limit placed on the simulation, set to current temperature of the cosmic microwave background. The dense wake shows only a tiny deviation from the background temperature. Effectively all of the heat generated by the collapse has bee radiated away. This shows the difference between the medium and high density cases. In the over-density plots, the alpha value was limited to provide the easiest comparison between runs. The temperature range is not limited in this way. While the temperature is lower in the high density case, the structure of the wake does not appear to differ, at least in overall shape.

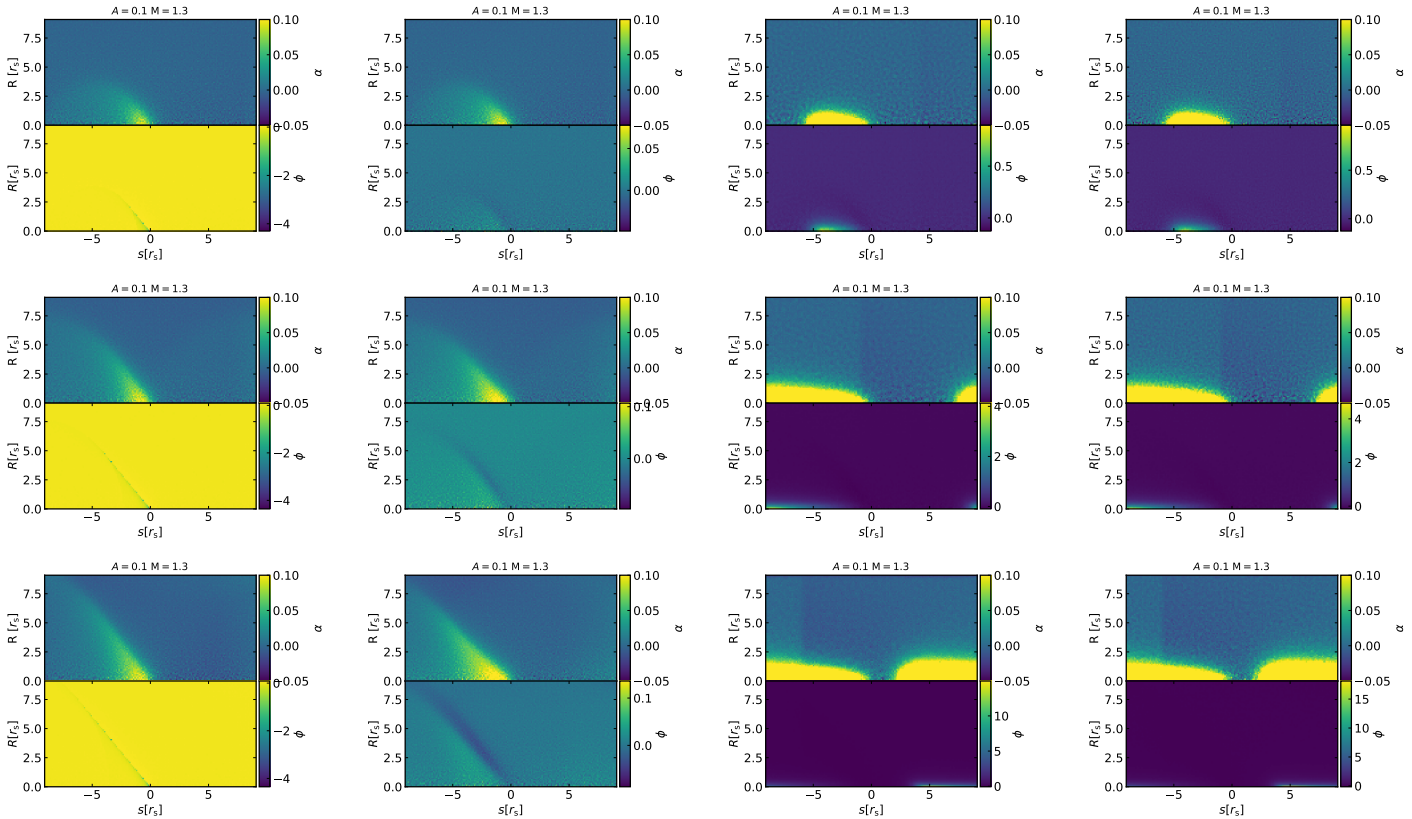

Figure 5.1 *Distribution of over-density α, for the solar metallicity runs. Time increases downwards, with snapshots at $t = 4t_c$ (top row), $t = 8t_c$ (middle row), and $t = 12t_c$ (bottom row). First column shows the benchmark no cooling/chemistry runs, second column the $n = 0.01\mathrm{cm}^{-3}$ results, third column $n = 1\mathrm{cm}^{-3}$, and fourth column $n = 10\mathrm{cm}^{-3}$.*

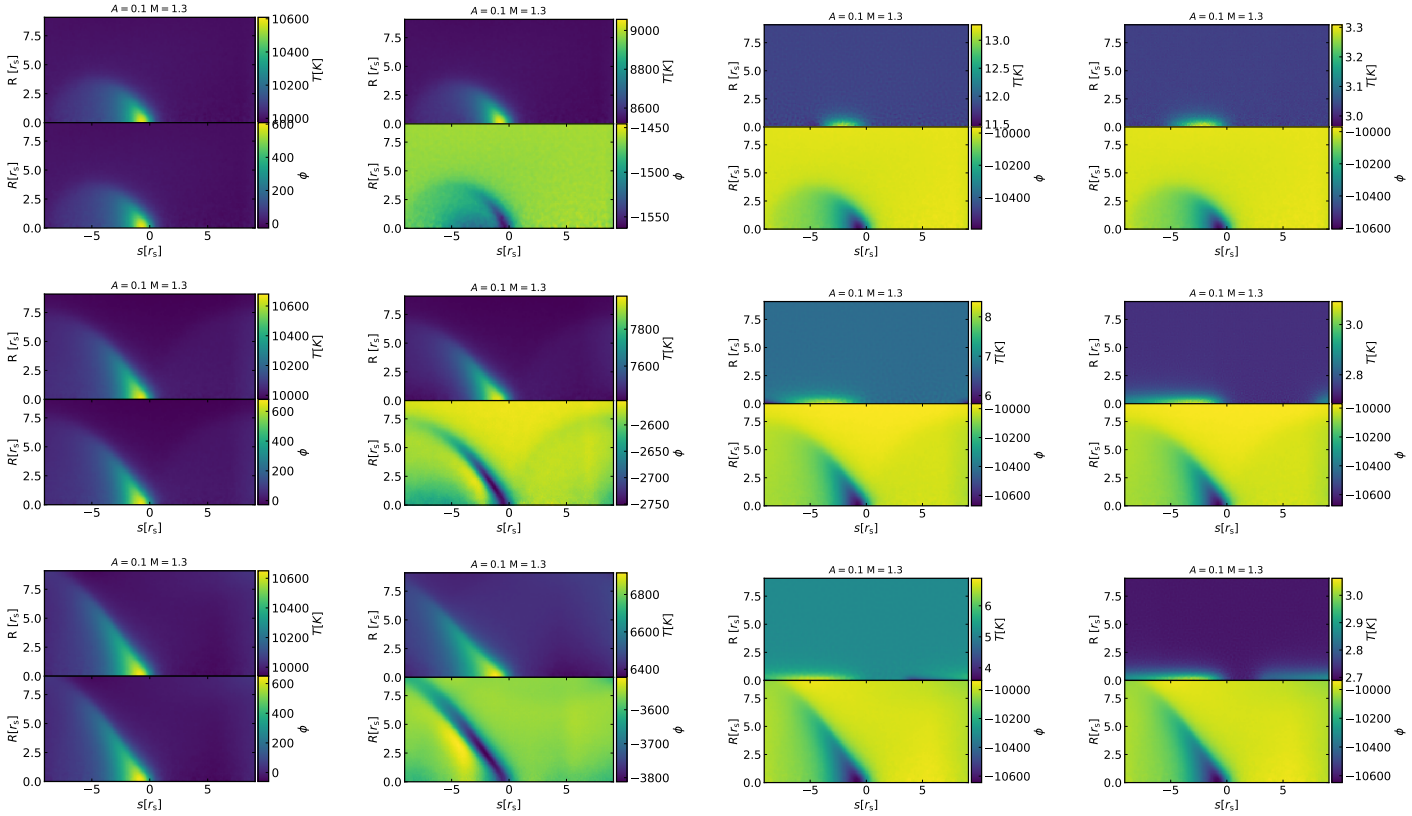

Figure 5.2 *Temperature distributions, in cylindrical coordinates, for the solar metallicity runs. Time increases downwards, with snapshots at $t = 4t_c$ (top row), $t = 8t_c$ (middle row), and $t = 12t_c$ (bottom row). First column shows the benchmark no cooling/chemistry runs, second column the $n = 0.01\mathrm{cm}^{-3}$ results, third column $n = 1\mathrm{cm}^{-3}$, and fourth column $n = 10\mathrm{cm}^{-3}$. The lower panels now show the difference between the run and the benchmark run $\phi = T_{\mathrm{cool}} - T_{\mathrm{nc}}$, at that time. The benchmark show the difference from the initial temperature.*

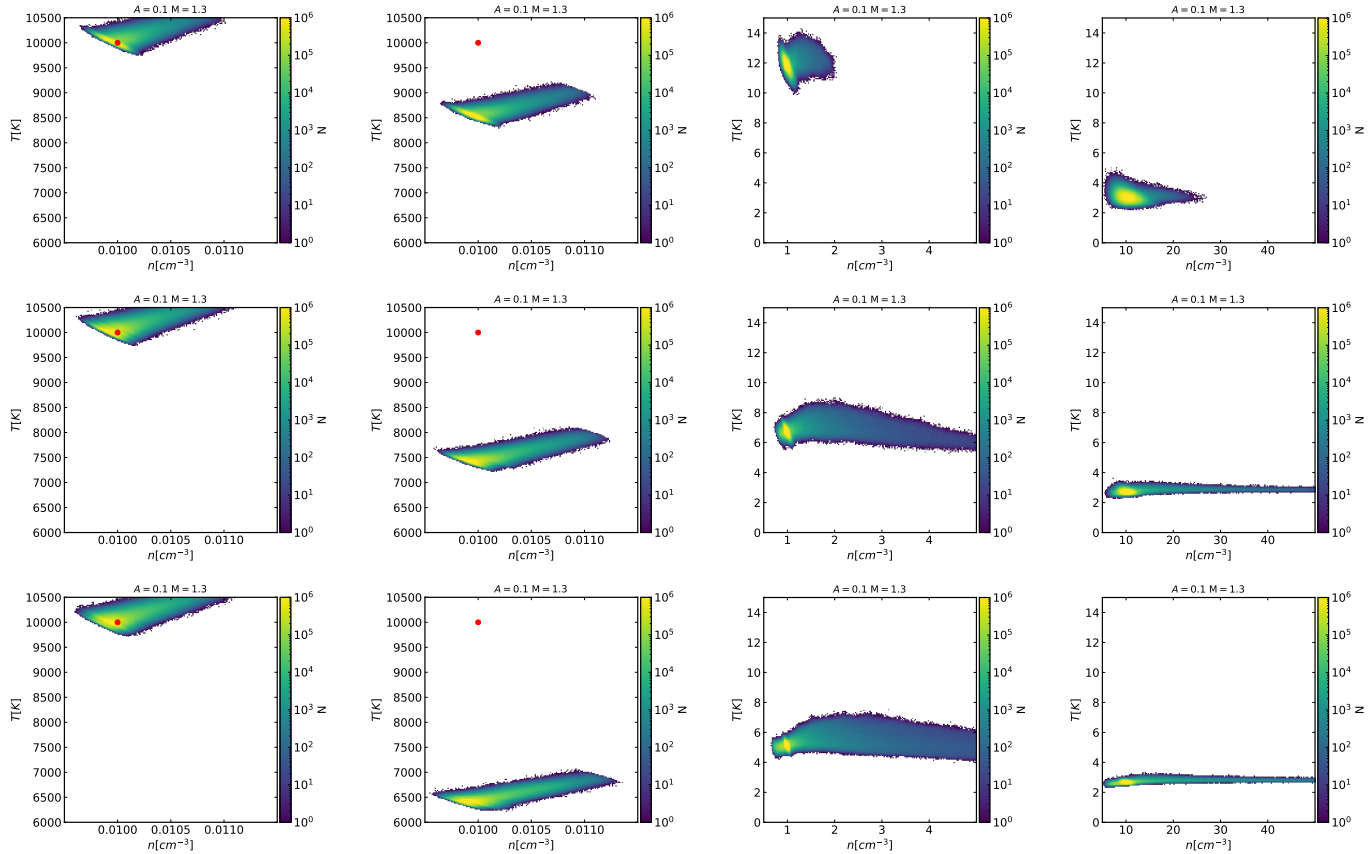

Figure 5.3 *Phase diagrams, showing number of particles with number density and temperature* (n,T)*, for solar metallicity. Time increases downwards, with snapshots at* $t = 4t_c$ *(top row),* $t = 8t_c$ *(middle row), and* $t = 12t_c$ *(bottom row). First column shows the benchmark no cooling/chemistry runs, second column the* $n = 0.01\mathrm{cm}^{-3}$ *results, third column* $n = 1\mathrm{cm}^{-3}$*, and fourth column* $n = 10\mathrm{cm}^{-3}$*.*

In both medium and high density cases, the wake has approximately the same extent. The form of the wake suggests that almost all pressure support has been removed, with the lateral extent of the wake likely caused by the conservation of momentum in the particle trajectories.

The detail of the differences, in over-density and temperature, between the two higher density cases, can be seen in the phase diagrams from the different runs. These are shown in Figure 5.3, alongside the phase-diagrams for the 'no cooling' and low density cases. The particles of the simulation are binned by number density n and temperature T. The colour bar indicates the number of particles in each bin, and the red dots show the initial background conditions, when they can be reasonably shown in the same axes. The lack of cooling is obvious in the first column, where the heating from the collapse of the gas is clear in the particles that lie around the linear tail. This extends from the initial conditions, with a density increasing proportionally to temperature. The spread is present in the initial conditions, and is simply caused by the glass like initial conditions not exactly replicating the desired uniformity. This tail is a clear representation of the adiabatic heating. In the low density case, the spread of particles is largely unchanged, with the whole ensemble decreasing in temperature at an approximately constant rate. The medium density case starts with the cluster of particles close to its initial density. In the first snapshot, the tail is less linear than the previous cases. It has a slight turnover, as we move to higher densities. This reflects the increased cooling that we expect at higher densities. In the lower density case, this cooling is outmatched by the heating from collapse.

As we move to the later snapshots in the medium density run, the tail becomes increasingly extended, even as the whole ensemble continues to cool. The turnover in the tail is clear, with no high density particles able to maintain any gains in temperature. We do not see these particles drop far below the temperature of the background particles. This is likely because the whole ensemble is already cooling so rapidly that the relatively small increases in density does not increase the cooling rate by a proportionally large amount. When I refer to the small density increase, this is still to an over-density of $\alpha = 5$. This is significant, when compared to the standard over-density that would be produced by dynamical friction, which should be $\alpha \ll 1$, assuming the scenario is within the linear regime of $A < 1$. While the high density case reaches higher densities, and lower temperatures, the bulk of over-densities it produces are still $\alpha \leq 5$, since the initial background density is higher. The spread in temperatures is smaller,

and the bulk temperature lower, showing the increased efficiency from cooling. By the final snapshot, the bulk of the gas has cooled to the imposed floor, but even before this the shape is flatter than the medium run. There is almost no signal of the increased temperature from collapse. The high density tail extend to greater over-density in the medium and high cases, but the number of particles beyond this point is not significant.

Overall, these solar metallicity results show that the addition of advanced cooling and chemical networks, can have a profound impact on the evolution of a gaseous, gravitationally induced, wake. The cooling of the gas produces greater over-densities, with the effect being most pronounced at the highest densities, where cooling is most efficient. In the low density case, while the gas does cool somewhat, the difference in the evolution of the wake is small. I will discuss the wider implications of these results in Section 5.5, but it should be noted that the setup used here is still highly idealised. Regions of uniform gas at these densities, and temperatures, are not to be expected in a cosmological setting. However, these results can still show how cooling can change the impact of dynamical friction.

5.4.2 Primordial Metallicity

I now show the equivalent results for the primordial metallicity runs. These runs use the same densities as their solar metallicity counterparts, but now with much smaller metal abundances. The figures are arranged in the same manner as before, with columns for each density, and the 'no cooling' benchmark. Time increases downwards. Looking first at Figure 5.4, we see that the wake is largely unchanged in the low and medium density cases. As there is no longer much metal line cooling, the medium density case cannot shed enough internal energy to overcome the pressure support that we see in the original adiabatic case, and the low density case. The medium density case does show some evidence of cooling however. There is an unexpected over-density forming a tail directly downwind of the perturber, which is not seen in the low density run. The cone is also slightly sharper, as seen in residual plot for the final snapshot. This snapshot also shows the over-dense tail clearly wrapping around. The high density run shows a similar extreme collapse to the solar run. The increased density provides enough cooling to overcome the pressure support, even without significant metal line cooling. In the first snapshot, however, there is also a peculiar break in the over-density, suggesting the presence of some kind of instability within the wake. This structure

has disappeared in the later snapshots. Otherwise, it would appear that the high density case can still cool efficiently, and so, once again, loses pressure support for the extended wake structure.

Moving to the temperature distributions, shown in Figure 5.5, we see the difference between the low and medium density runs more clearly. The low density case is still almost identical to the benchmark run, with the temperature varying from it by only a few Kelvin. The medium density case shows the bulk cooling of all particles, caused by the higher density. It also shows a cooler region behind the position of the perturber. This material has cooled more than the background, because it has passed through the densest part of the wake. In the solar metallicity runs, the cool region closely mapped to the dense region, but in this case there seems to be some disparity, with the cool region extending to parts that are not particularly over-dense. Material that was previously in the dense region has cooled enough to be distinct from the background. This is the material that will go on to form the over-dense tail directly behind the perturber. Its lower temperature seems to allow it to collapse further, even though it is not currently particularly dense. Something similar has likely happened in the high density run, where the low density break is also cooler than the surrounding high density wake. As density increases, more molecular hydrogen will form, and this will also contribute to the increased cooling in the high density regions. The molecular hydrogen fraction is discussed in greater detail in Section 5.4.4.

The phase diagrams for these runs can be used to better understand these new structures. Shown in Figure 5.6, these demonstrate that the low density run is essentially unaffected by the additional physics. The medium density case, however, has a wing of particles that decreases in temperature, as the density increases. These are the particles that form the over-dense tail directly behind the perturber. They have cooled as they pass through the dense front of the cone, but some of them are now in the lower density part behind the cone. They do not gain temperature again, so remain in this separate wing of the phase structure. The high density case has a similar evolution to the solar counterpart, just with significantly less cooling. The high density tail does not increase in temperature, as it does in the benchmark, since the particles are losing heat through the cooling processes, which are once again more efficient at higher density. Since the cone structure does not form at all, the wing observed in the medium case does not form. There are no particles that move from a high density cone front, to the lower density region behind the front.

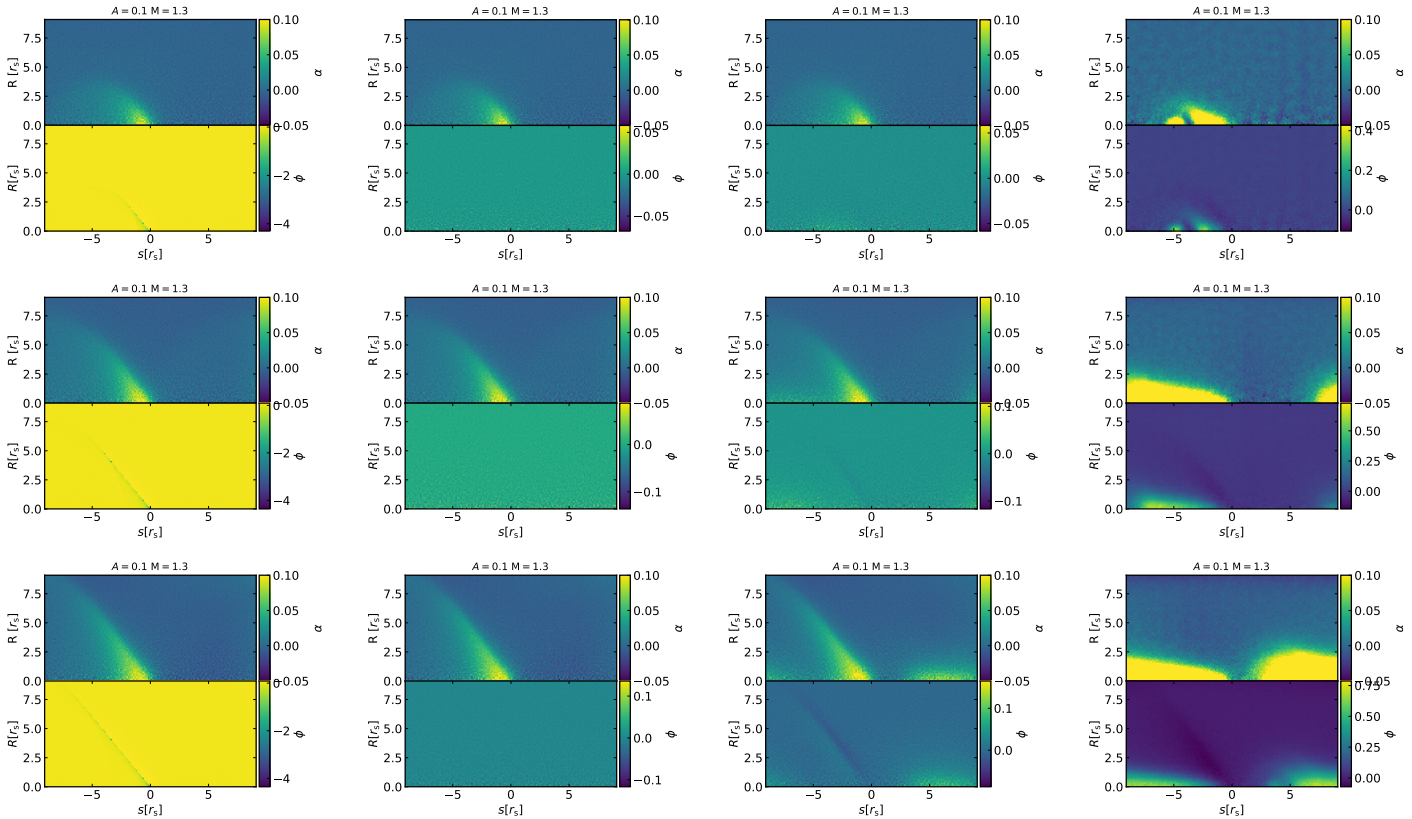

Figure 5.4 *Distribution of over-density α, for the primordial metallicity runs. Time increases downwards, with snapshots at $t = 4t_c$ (top row), $t = 8t_c$ (middle row), and $t = 12t_c$ (bottom row). First column shows the benchmark no cooling/chemistry runs, second column the $n = 0.01\mathrm{cm}^{-3}$ results, third column $n = 1\mathrm{cm}^{-3}$, and fourth column $n = 10\mathrm{cm}^{-3}$.*

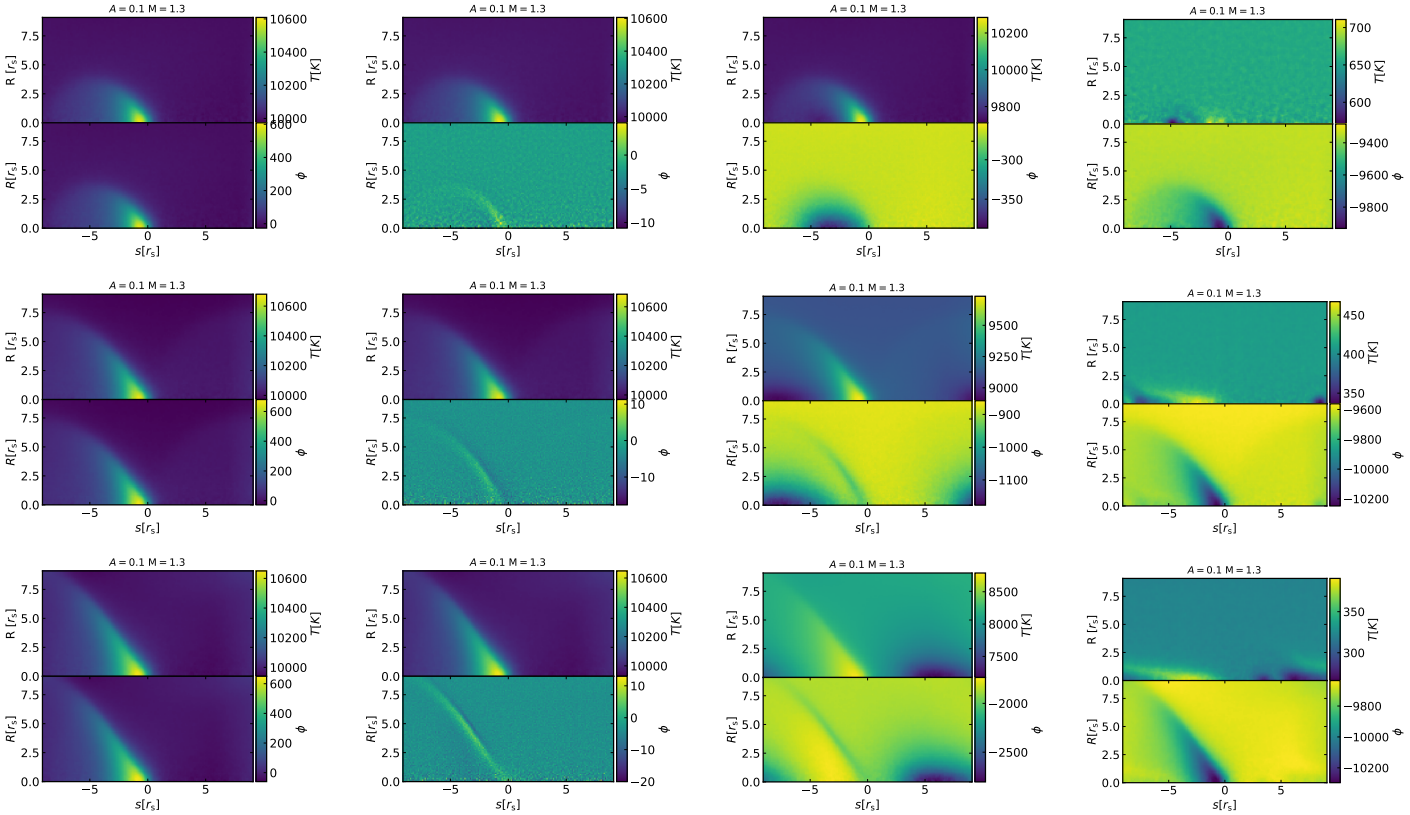

Figure 5.5 *Temperature distributions, in cylindrical coordinates, for the primordial metallicity runs. Time increases downwards, with snapshots at $t = 4t_c$ (top row), $t = 8t_c$ (middle row), and $t = 12t_c$ (bottom row). First column shows the benchmark no cooling/chemistry runs, second column the $n = 0.01\text{cm}^{-3}$ results, third column $n = 1\text{cm}^{-3}$, and fourth column $n = 10\text{cm}^{-3}$.*

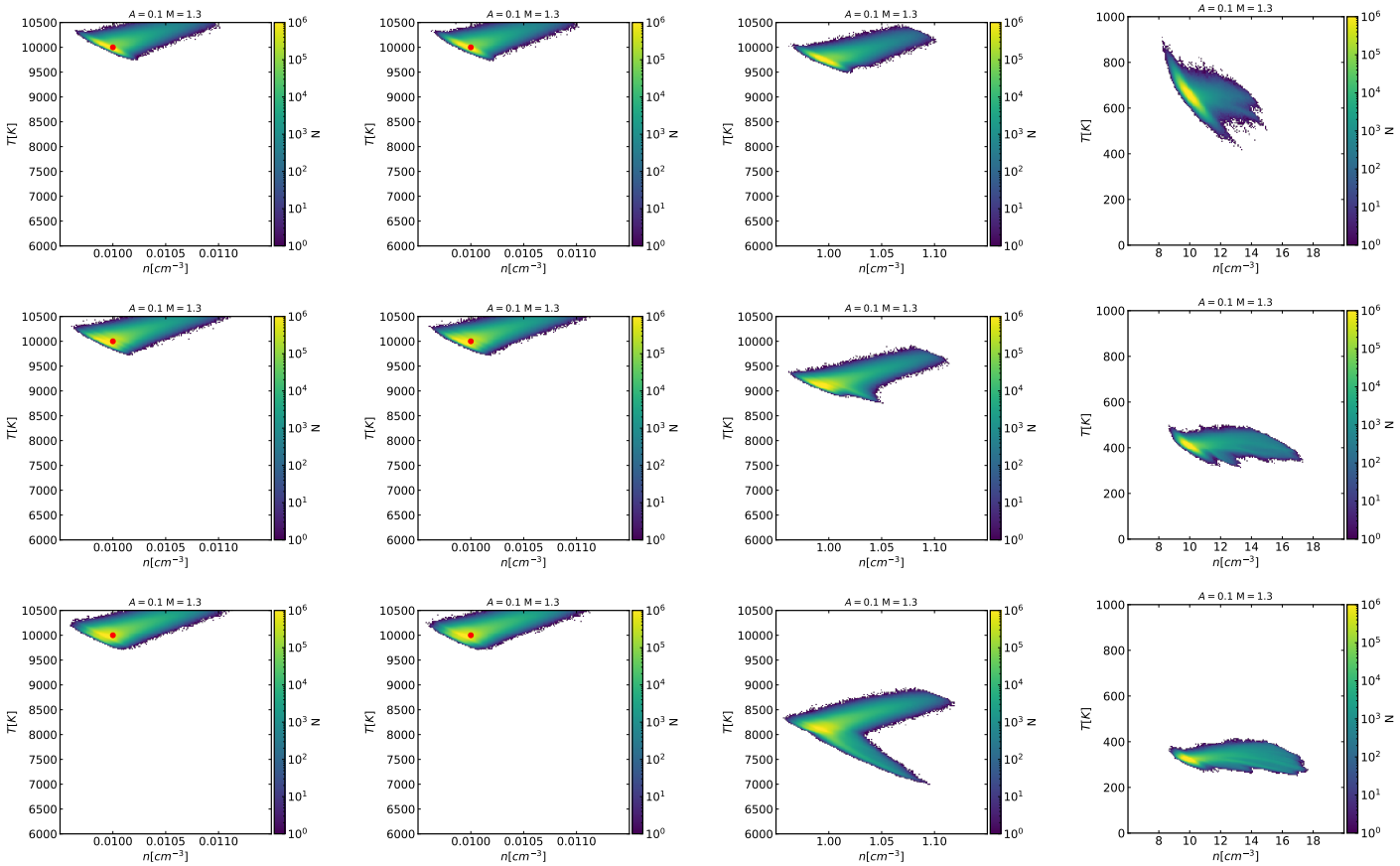

Figure 5.6 *Phase diagrams, showing number of particles with number density and temperature* (n,T)*, for solar metallicity. Time increases downwards, with snapshots at* $t = 4t_c$ *(top row),* $t = 8t_c$ *(middle row), and* $t = 12t_c$ *(bottom row). First column shows the benchmark no cooling/chemistry runs, second column the* $n = 0.01\mathrm{cm}^{-3}$ *results, third column* $n = 1\mathrm{cm}^{-3}$*, and fourth column* $n = 10\mathrm{cm}^{-3}$*.*

Taking this all together, the lack of significant metal content shifts the density at which cooling dominates the formation of the wake. In these primordial cases, only the very high density of $n = 10\text{cm}^{-3}$ can cool efficiently, while in the solar metallicity case, $n = 1\text{cm}^{-3}$ gas can also cool rapidly, leading to a much more over-dense wake. Even in the cases where the formation of the wake is not dominated by cooling, the additional physics has some effect on the detail of the wake, in general producing more over-dense structures than are seen in the idealised analytic prediction.

5.4.3 Drag Force

To see the impact of these new features, we can consider the net drag force, on the perturber, from the over-dense wake. This is calculated by direct Newtonian force summation. The summation is performed between all particles in the box and the perturber. In Figure 5.7, I show this force, in its dimensionless form. The plot on the left shows the results for the solar metallicity runs, with the primordial metallicity results on the right. The black line gives the analytic prediction, with the lower panels showing the residual between the numerical results, and this analytic solution, such that $\phi = (F_{\text{num}} - F_{\text{ana}})/F_{\text{ana}}$. The blue dashed line shows the drag force from the 'no cooling' benchmark case, the orange the results from the low density run, green the medium density, and red the high density. In both cases we see a smooth increase in force, until about $t = 6t_c$. After this point in the solar case, the medium and high density runs, which have essentially identical forces, show a sharp turnover, with the force dropping to zero, before rebounding. The same is found in the high density run for the primordial case. This is the time when the wrapping around of the perturbation shows its effect. In the higher density cases, a large over-density wraps around.

Since the force effectively comes from any over-density against the background, the wrapping around of the dense wake creates a forward force that counteracts the drag. In the lower density cases, the effect is much less pronounced, but still present. This turnover makes analysis at the later times difficult, so they have been excluded from the plot. To check for any force variation from changes to the background wrapping around, I calculate the force using only those particles that are behind the perturber. This will also exclude any structures that protrude ahead of the perturber, but from the over-density plots in Figure 5.1, it is clear that such structures are very limited, in all cases. To remove the force from

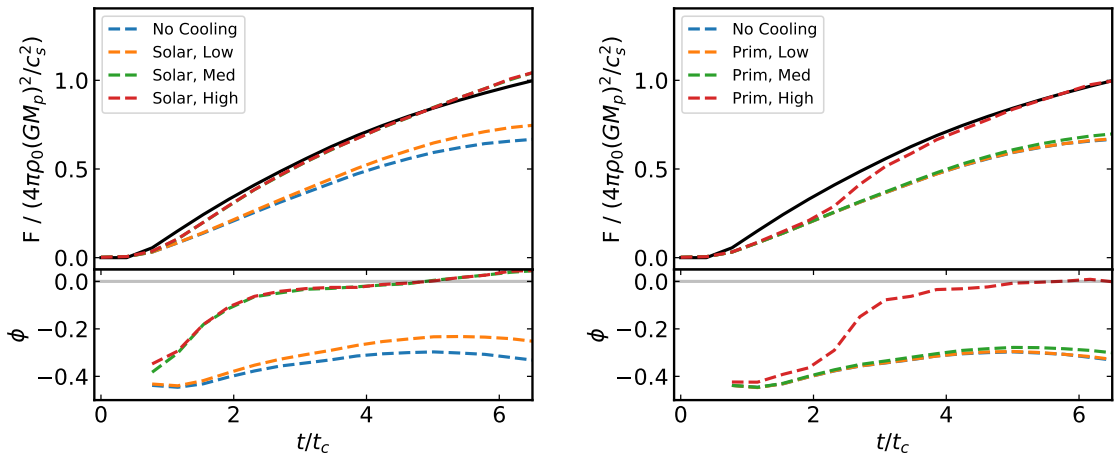

Figure 5.7 *Upper panel: Net dimensionless drag force from all particles, for the benchmark case (blue), and the low (orange), medium (green), and high (red) density cases. Black line shows analytic prediction. Lower panel: Residual between the numerical and analytic forces* $\phi = (F_{\rm num} - F_{\rm ana})/F_{\rm ana}$.

the background density, I subtract the force calculated for the initial particle distribution behind the perturber, from the force calculated at each snapshot. For the times shown in Figure 5.7, the force is unchanged from the previous result.

We see a trend in the force with increasing density. As we move from the low density to high density runs, the net drag force increases. In the solar metallicity case, the low density force is only slightly higher than the benchmark run, while the medium and high density runs shows much greater forces. In the primordial metallicity case, the same pattern is seen, except the medium density case is much closer to the low density and benchmark runs. Both of these observations fit well with the distributions of material that we see in the over-density plots. It is noticeable that the higher density cases come close to matching the analytic solution. This is largely coincidental, as we already know that the numerical results do no replicate their analytic counterparts. It is of more use to compare the chemistry runs to the run with no cooling, as this will disentangle the effects of adding cooling. This comparison will be performed later. The 'no cooling' run uses assumptions that best match the analytic solution, and so should be the closest to that result.

The analytic solution is largely included as a guide, but it does bring us to a peculiar result from these runs. The benchmark run, shown in blue, produces a force that is significantly below both the analytic solution, as previously found,

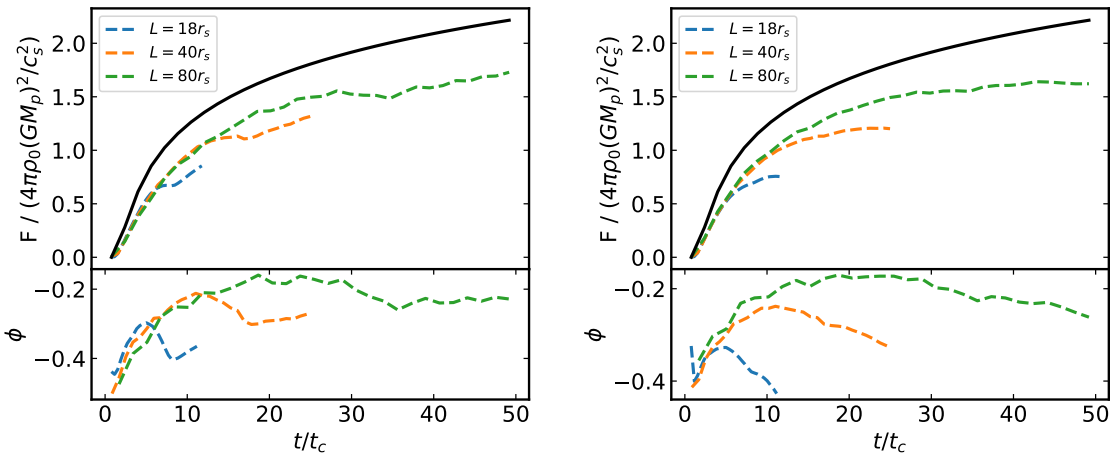

Figure 5.8 *The left plot shows the results using all particles in the wake, while the right plot show the force from only those particles behind the perturber. Upper panel: Net dimensionless drag force from particles behind the perturber, for the benchmark case with $L = 18r_s$ (blue), $L = 40r_s$ (orange), and $L = 80r_s$. Lower panel: Residual is between the numerical and analytic results $\phi = (F_{\rm num} - F_{\rm ana})/F_{\rm ana}$.*

and the equivalent result from Chapter 2. The force for $A = 0.1$ and $\mathcal{M} = 1.3$ was previously found to have an offset of approximately 10%, when compared to the analytic prediction, but it now sits at about 40% (see Section 2.4.1 for previous results). This is very odd, since one of the previous tests included varying the specifics of the initial conditions, to see if the result was genuinely scale free. While the test did not specifically cover the regime used here, they did show that previous offset did not change with different combinations of initial conditions. Why this would change for the new case is not clear. As mentioned before, the resolution of these runs is higher than the work presented in the previous DF chapter. For simple comparison, both use the same number of particles, but the original runs have a box that has edges of length $L = 80r_s$, wheres these runs use $L = 18r_s$. This means that the average particle separation, in terms of gravitational softening scale, is about four times smaller in these new chemistry runs.

In Figure 5.8, I show the force from two setups designed to check how the ratio of box size to softening scale might effect the recovered force. The blue line shows the standard setup for the chemistry runs, the green line the effective ratio of the previous runs, and orange a ratio in between the other results. The change is achieved by decreasing the softening scale. The perturber mass is decreased alongside the softening scale, to keep the same A value. The simulation is run for the same physical time, so the number of crossing times increases as the

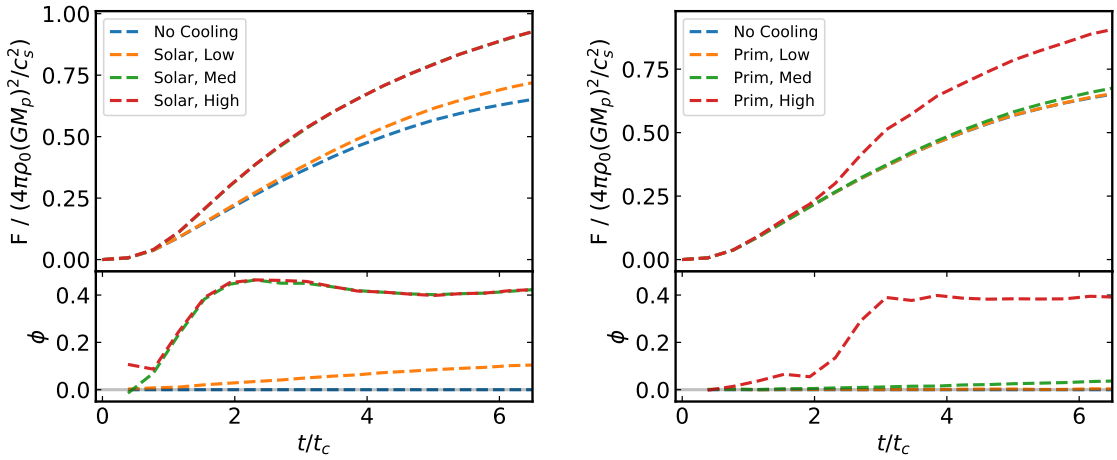

Figure 5.9 *Upper panel: Net dimensionless drag force from particles behind the perturber, for the benchmark case (blue), and the low (orange), medium (green), and high (red) density cases. Lower panel: Residual is now between the runs with cooling, compared to the one without* $\phi = (F_{\rm cool} - F_{\rm nc})/F_{\rm nc}$.

softening scale decreases. The results that use all particles (left) all follow the same trajectory, until the point where the wake wraps around through the box boundary. This is at a different number of crossing times for each case, but the trajectory after this point is very similar, with the force decreasing. This effect is somewhat missing in the right plot, where only the particles behind the perturber are used, but the signal can still be seen. This is likely caused by rarefaction of the wrapped around background, as some material has been retained in the over-dense wake. The peak in this distribution seems to increase with the number of crossing times that can realistically be reached. I noted a similar potential convergence in Chapter 2, although it is unclear if the numerical result will converge on a force that is below the analytic prediction, or if it will eventually match it. My previous result showed convergence was not achieved, even at $t > 100t_c$. Other numerical results (Kim & Kim, 2009) found a match at $t > 300t_c$, but do not discuss the results before that time, and this lead time for convergence was not found by other numerical works (Bernal & Sánchez-Salcedo, 2013). Either way, it would seem that the increased discrepancy is simply caused by a combination of the relatively early time, and the later wrapping around of material. This is interesting, but since the focus of this work is on the change in the numerical result when additional physics is added, a comparison between the numerical results alone is adequate, for our purposes.

I now compare the results from the new chemistry runs to their benchmark,

shown in Figure 5.9. Here, we clearly see that the increased over-density in the wake, caused by the addition of cooling and the 9-species chemical network, has a strong effect on the net force. In the solar case, as noted before, the medium and high density runs show effectively the same force evolution, producing a force that is almost half as great again, when compared to the benchmark case. The low density run shows a smaller increase, rising to a value around 10% higher than the benchmark. This clearly corresponds well to the observations made about the over-density in the wake. Both the medium and high density cases cool so rapidly that their wakes lose all pressure support. The wakes collapse to much greater over-densities, with more material in the inner wake, close to the perturber. It is interesting that small additional collapse allowed by the slower cooling in the low density case, is still enough to provide a notable increase in force. Taking the primordial metallicity case, in comparison, we see that the high density result follows an almost identical path. Even in the absence of significant metal content, material at this density and initial temperature is able to cool rapidly, presumably through a combination of atomic and molecular cooling. The low and medium density forces, at this metallicity, show very little impact from the cooling processes. In these cases, cooling is too inefficient, without more metals, to counteract the heating from the collapsing material.

5.4.4 Molecular Hydrogen

As has been discussed previously, linking dynamical friction to the formation of molecular hydrogen (H_2) could have strong implications for the evolution of early galaxies. In summary, molecular hydrogen is an important component in the evolution of galaxies. It is found in star forming regions, and is thought to be a key driver of star formation itself. At early times, when the metal fraction is extremely low, before the formation of the first stars, it provides a possible channel to cool gas. This could be a mechanism by which the first stars form (Bromm, 2013), as the cooling reduces pressure support, and regions of gas can collapse under self-gravity. However, without the presence of metals, the formation of molecular hydrogen can only proceed via two-body, or three-body, interactions. The formation rate is therefore dependent on the square of the number density of the gas, or the cube, in the three-body case. The cooling itself will also increase with density. It is possible that DF from, for instance, dark matter substructure, could trigger the initial increase in density, and so drive the formation of more molecular hydrogen. The advanced chemical networks used

here provide information on the fraction of the gas that is made up of molecular hydrogen. I use the evolution of this fraction to speculate on the possibility of DF stimulating the formation of molecular hydrogen.

In Figure 5.10, I show the phase diagrams for the solar metallicity runs. Where before the colour bar showed the number of particles in each bin, it now shows the mean fraction of molecular hydrogen for the particles in that bin. Starting with the low density $n = 0.01\text{cm}^{-3}$ run, the gas is initialised with a very small molecular hydrogen fraction. Even at the final temperature reached by this run, the temperature is to high to allow much H_2 to survive. Collisions disassociate hydrogen molecules, and the small increase in density caused by the perturber is not enough to counteract this. At medium density $n = 1\text{cm}^{-3}$, in the first snapshot, we see that the tip of the material at lowest density shows a small increase in molecular hydrogen fraction, but this increase is washed out by the bulk changes in the fluid state. Even as it cools, it largely retains its molecular hydrogen fraction, but does not show any promotion of H_2 formation, even in the high density tail. A very similar pattern is seen in the high density $n = 10\text{cm}^{-3}$ case, but with a higher equilibrium H_2 fraction, caused by the higher overall density. These results suggest that at solar metallicity, presence of a perturber does not change the formation of H_2. With metals present in the gas, molecular hydrogen formation can proceed on the surface of dust grains, so the increases in density will have less impact.

The primordial results, on the other hand, show a more complex picture. Shown in Figure 5.11, the results for the low density case show the same trend as the solar results, but with at an even smaller H_2 fraction. With even less cooling, and no metals to promote formation via dust, very little H_2 can survive. The medium density case shows strong evidence for the stimulation of H_2 formation by gravitational perturbation. In the final snapshot, we see that the high density, low temperature wing has an increasing fraction of molecular hydrogen. The increase in density, combined with the decrease in temperature, has allowed H_2 to form, and survive. The increased H_2 fraction will also have increased the cooling rate, allowing further collapse. This structure is distinct from the bulk cooling of the background medium, and the standard linear structure produced in the benchmark run. When we compare the medium density to the high density run, the same strong effect is not present. There is some gradient in the fraction, in the first snapshot, but the relative change is less pronounced. This gradient is largely gone at the later times.

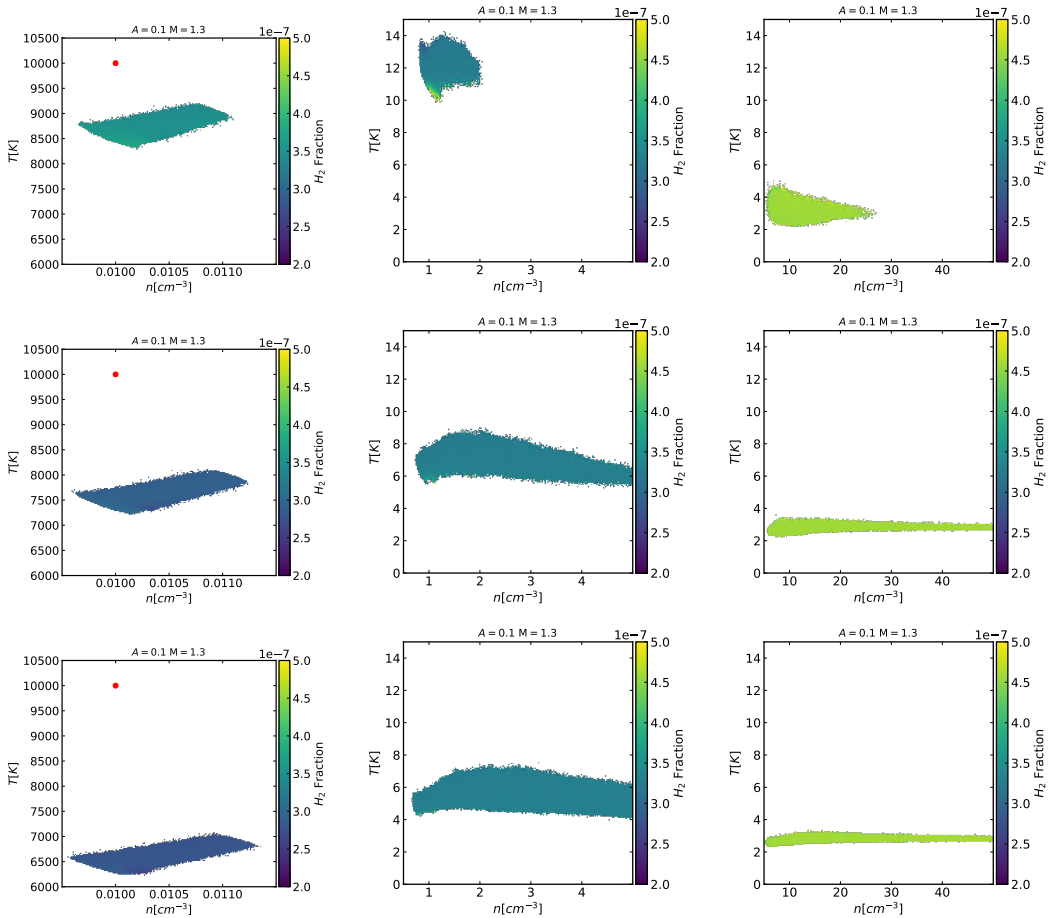

Figure 5.10 *Phase diagrams of the solar metallicity runs, with the colour bar showing the mean molecular hydrogen fraction, for the particles in that* (n,T) *bin. First column shows* $n = 0.01\mathrm{cm}^{-3}$*, second* $n = 1\mathrm{cm}^{-3}$*, and third* $n = 10\mathrm{cm}^{-3}$*.*

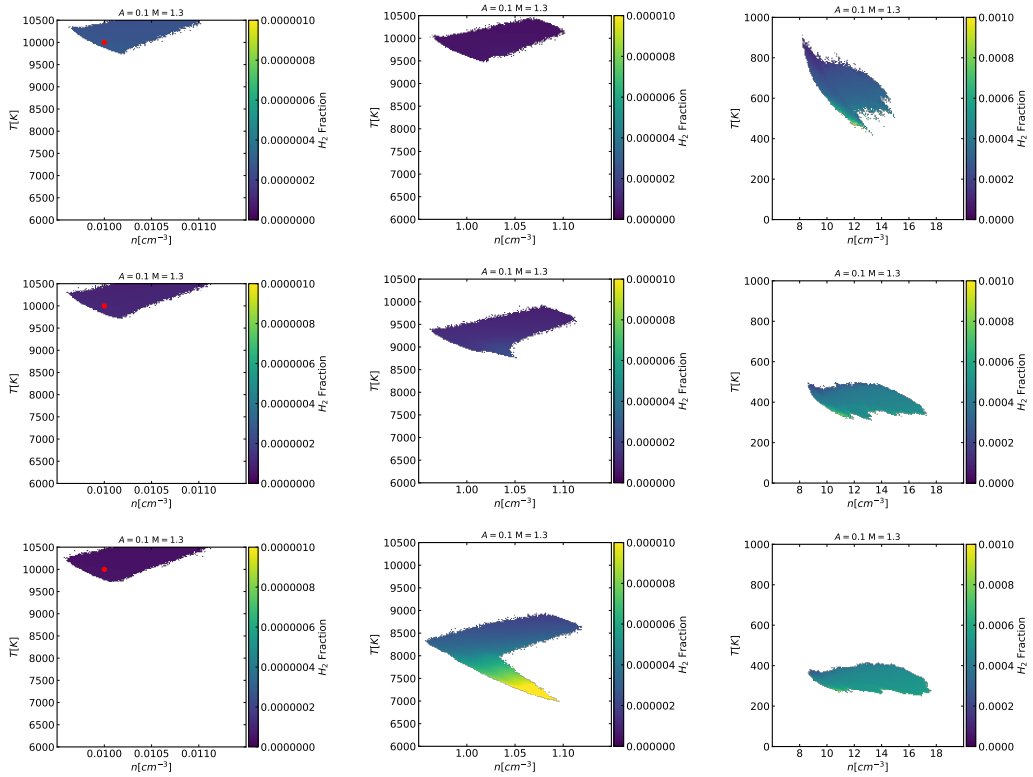

Figure 5.11 *Phase diagrams of the primordial metallicity runs, with the colour bar showing the mean molecular hydrogen fraction, for the particles in that* (n, T) *bin. First column shows* $n = 0.01\mathrm{cm}^{-3}$*, second* $n = 1\mathrm{cm}^{-3}$*, and third* $n = 10\mathrm{cm}^{-3}$*.*

The overall fraction is also much higher, driven by the higher initial density. From these results, it is clear that dynamical friction can drive the formation of molecular hydrogen, in the absence of metals. The scenarios in which than can happen are still determined by the combination of density and temperature of the gas, and there appears to be a narrow set of circumstances where the effect is most pronounced.

When we compare the fraction formed in these solar metallicity runs, to those found in the primordial runs, we see that the fraction is several orders of magnitude lower for the solar runs. Even at high density, where we see the highest molecular fractions, the solar runs produce $f_{\rm mol} = 4.5 \times 10^{-7}$, while the equivalent primordial runs show $f_{\rm mol} = 6 \times 10^{-4}$. This suggests that the very low temperatures and high densities reached in the solar run is somehow either disrupting the formation of molecular hydrogen, or destroying it. Alternatively, the high metallicity is not producing more molecular hydrogen, as would be expected, but is instead somehow suppressing its formation. Exactly what is causing this unexpected difference is not clear at this moment, and requires further study. It must, however, be somehow dependent on the detail of the collapse dynamics in the different cases.

5.5 Discussion and Summary

I have shown the impact of cooling processes on the development of a gravitationally induced over-dense wake. For solar metallicities, scenarios with densities $n \geq 1\text{cm}^{-3}$ show a complete loss of pressure support over timescales of tens of mega-years. The resultant wake is much more dense, and produces a much greater drag force, larger by as much as 50% of the force from a wake with no cooling. At this metallicity, across all densities, the formation of molecular hydrogen is dominated by the background conditions, and not driven by the perturbation of the density. This is likely caused by the high metal content, which provides a channel for molecular hydrogen formation on the surface of dust grains. The primordial metallicity results show less extreme cooling, as is expected when there are less metals. Only at $n = 10\text{cm}^{-3}$ do these results produce a significantly different wake structure. This case produces a force that is roughly 30% above the previous value. In the low and high density cases, molecular hydrogen levels are largely constant across the box, showing the abundance is once again driven by the background conditions. In the medium density case, where $n = 1\text{cm}^{-3}$,

we see strong evidence for DF driven molecular hydrogen production. The high density, low temperature wing that builds from the bulk of the particles, show an increasing H_2 fraction. This suggests that, under the right circumstances, DF can stimulate H_2 production, and so potentially trigger, or help to trigger, very early star formation.

It should be noted that the setups used in this chapter, while including more physical processes than my other DF work, are still highly idealised. Encountering a 1kpc^3 region of uniform temperature and density gas, at any of these densities, is highly unlikely. Observations of the ISM and CGM show they have much more complex, turbulent, structures, with gas existing in multiple phases throughout a given region. These results do, however, show the significant difference that cooling can have on the gravitationally induces wake. The structure and net drag force are both strongly effected by the reduction or even loss of pressure support. These changes would likely still manifest in a more complex environment, and the potential for increasing the production of molecular hydrogen in a primordial environment still has wide reaching implications. If DF, from the previously discussed dark matter substructure, can trigger even relatively small increases in H_2 formation, it could dramatically influence our understanding of the formation of the first galaxies.

The scenarios described here are motivated by changing as little as possible between runs, to isolate the cause of any differences in the gravitationally induced wakes. However, this leads to initial conditions that are initialised far from thermal equilibrium. The medium and high density runs in the solar metallicity case, and the high density run on the primordial case, all cool rapidly from the start of the simulation. The cooling time scales are much shorter than the time scales associated with the formation of the wake structure. This leads to an evolution of that is entirely dominated by the extreme cooling of the gas. A different set of initial conditions, using the same density, but where the initial temperature is chosen such that the background density is in thermal equilibrium, would add new information on the formation of the wake in more physically likely conditions. For the solar metallicity case, this would have the medium density case start at $T \approx 10\text{K}$, and the high density case at $T = 2.7\text{K}$. The latter temperature is the value of a temperature floor in the code that keeps the temperature at or above the current observed temperature of the CMB. The high density primordial case would be initialised at $T \approx 300\text{K}$. These scenarios would allow us to observe the development of the wake, in the presence of strong

cooling mechanisms, without the extreme changes introduced by starting so far from thermal equilibrium. The changes in background temperature also change the sound speed of the gas. This could complicate elements of the comparison, as the physical times are no longer the same between setups. However, if the Mach number of the initial bulk flow is kept constant, direct comparison can still be achieved, thanks to the scale free nature of the original problem. The cooling processes are not scale free, since they depend on density, but it should still be possible to compare the different scenarios by using the sound speed crossing time to normalise the different physical times. These hypothetical runs require a careful consideration of the different time scales, in order to produce accurate comparisons to the work presented in this chapter.

Chapter 6

Conclusion and Future Work

6.1 Dynamical Friction

I have shown, through a set of idealised gravo-hydrodynamic simulations of gaseous dynamical friction, that state-of-the-art hydro solvers, used for a variety of astrophysical simulations, systematically under-produce the expected gravitational drag force. This retarding force, acting in opposition to the direction of travel of a perturbing mass, is generated by the over-dense wake, which is produced by the gravitational perturbation of the gaseous medium. The mismatch is found for supersonic perturbers, with Mach number $1 \leq \mathcal{M} < 2$. The detail of the structure of the wake does not match that predicted by linear perturbation theory, with the largest differences found close to the perturber. Most notably, the sharp edge of the Mach cone is not recovered well, and parts of the wake extend forward of the position of the perturber. The difference was found for different levels of linearity, defined by the $A = GM_p/c_s^2 r_s$ parameter, with the force mismatch present down to $A = 0.01$. To understand how these conditions correspond to those found in cosmological simulations, I show the distribution of dark matter sub-halos in the coordinates that define the scale free idealised DF runs. A large fraction of the sub-halos identified in the IllustrisTNG-300 simulation box exist in conditions that produce a mismatch in my idealised setups. This suggests that these structures are not experiencing enough DF, in these simulations, as opposed to their physical counterparts, which has important implications for the merger histories of galaxies. I proposed a standard gravo-hydrodynamics test, based on this idealised setup, that could be applied to the

many simulations codes used in the field.

These highly idealised runs were followed by runs that include additional cooling physics, with the associated calculation of molecular chemistry. The new runs use much of the same idealised setup, with the core difference being the addition of the new physics. They show the profound impact that cooling, from atoms, metals, and molecules, has on the evolution of a gravitational perturbation of a uniform gaseous medium. At high metallicity, cooling from metal line emission removes much or all of the pressure support for the structure of the gravitationally induced wake. Much greater densities are reached, and the drag force is greatly increased. This effect is greater at higher densities, where cooling is more efficient. At primordial metallicity, metal line cooling is largely gone, but cooling from atoms and molecules is still present. In these runs, we saw that only the highest density case could cool enough to lose all of its pressure support, showing that it is still possible for this to happen, even without metals. At $n = 1\text{cm}^{-3}$, in the medium density case, I also show that parts of the gravitationally induced wake can stimulate the production of molecular hydrogen, which could potentially provide an avenue for DF induced cooling to produce the first stars.

6.1.1 Self Gravity

The idealised runs used in this work, and those that include the cooling processes, ignore the self gravity of the gaseous medium. The only gravitational force that the gas experiences comes from the massive perturber. This is one of the assumptions made in calculating the analytic solution. If the self-gravity of the gas were included in the modelling of dynamical friction, the over-densities created by the gravitational perturbation of the massive object would continue to grow under their own gravity. A wake that was previously stable may become unstable to its own gravity, producing higher density regions in the wake. This would lead to greater drag forces on the perturber. Since the numerical results show that the numerical wake does not reach the expected high densities in the cone front, the growth of these densities due to self-gravity would also be suppressed, assuming the collapse under self-gravity becomes non-linear. Smaller initial over-densities produce less collapse, and so further undershoot the physically correct wakes. The numerical wake would therefore produce force that is proportionally lower than the idealised scenario produces without self-gravity. This could point to an even greater deficit in the DF force felt by structures in numerical simulations,

further distorting the rates of processes, such as mergers.

6.1.2 DF in Cosmological Context

The ability of modern cosmological simulation codes to accurately capture the effects of both collisionless and gaseous classes of DF, in their full cosmological context, has not been studied in detail. Capturing the effects of DF in simulations requires modelling the hydrodynamic response, as well as the purely gravitational effects. As has been discussed, the nature of the hydrodynamic problem makes it difficult for traditional hydro solvers to model accurately. The extended gaseous structure is highly unstructured. Based on the results presented in this work, it is clear that the key next steps include investigating DF in its full cosmological context. This could be done by running a set of cosmological zoom boxes, focused on a suitably massive DM halo. The simulations can end at z=10, before the onset of significant star formation or reionisation, to avoid the complicating the results with additional heating and feedback. The effects of low mass DM substructure will be most pronounced in this epoch, when gas fractions are higher than at later times. This setup would allow one to study the CGM, in the vicinity of substructure within a halo, in great detail. It would be possible to examine the density perturbations and heating, as well as the promotion of molecular hydrogen formation by this perturbation. These results could be used to make predictions about the evolution of proto-galaxies. The CDM model predicts the build-up of structure through the collapse of regions of increasing size, and DF driven mergers. On top of this, there is the potential contribution to the chemo-thermo evolution of the gas of the host. The detail of this crucial process, in the full cosmological context and looking for the possible signature of DM substructure, is not yet well understood.

6.2 RD Solver

I have presented my implementation of the truly multi-dimensional residual distribution family of hydro solvers. These solvers produce the numerical solution to the Euler equations, to second order accuracy in both time and space. They utilise a mesh of triangles, the Delaunay triangulation, of an arbitrary distribution of vertices. I have shown that with little targeted optimisation, it can compete

with other modern hydro solvers, for problems directly related to those found in astrophysical scenarios. This included extensive use of standard hydrodynamics tests, such as the multi-dimensional flows in the Kelvin-Helmholtz instability, and the taxing Sedov blast, for which the RD solver produced accurate results. I demonstrated the different advantages of the various RD schemes, and noted that the solver could resolve structures at remarkably low resolution. The LDA scheme shows its superior ability to resolve smooth flows, while N scheme produces accurate and stable results for problems involving extreme shocks. This improved shock handling is demonstrated by the removal of spurious oscillations in the shock profile, when going from the LDA to the N scheme, in the Sod shock tube test. The differences in capturing smooth flows is seen most clearly in the KH test, where the LDA scheme is able to resolve considerably more structure than its N scheme counterpart. The potential for blending these schemes, through the use of a blending coefficient, provides an excellent balance between the two, allowing the possibility of highly accurate results for scenarios that include a wide variety of physical conditions.

I covered the extensions I have implemented to the basic scheme, including the introduction of the adaptive time-stepping mechanism, and the conversion to full 3D. I showed, in detail, the complications of applying the time-stepping optimisation, discussing the implications for conservation of mass and energy. Although I found a small loss in conservation, the difference remained very small for most setups, and the mechanism offered an excellent boost to performance. I have demonstrated the abilities of the various RD schemes using tests with up to 3D. Finally, I discussed the possible ways that gravity can be included in the solution, and showed preliminary results from the first gravity based test, using the setup proposed from my dynamical friction work. Once again, even at low resolution, much of the expected structure in the wake was recovered, demonstrating the accuracy of the solver.

6.2.1 Optimisation and Extension

The results and discussion, presented in this work, are based on my current implementation of the RD approach. As with many numerical methods, there are many options for further improving this new family of methods. Much of what I have discussed here are the results from the RD solver 'straight out the box', with little specific tailoring or optimisation. These results are strong, and

produce a solver that is already able to compete with others used in the field, but there are always going to be further improvements available. For instance, many of the results are produced on entirely random vertex distributions. Replacing these with distributions designed to match the expected flows could produce even better results, such as starting the Sedov blast with more vertices close to the centre, so that the initial blast shape is more regular. As it stands, the solver produces excellent results, but the underlying methods have even more to offer. There are a number of computational improvements and optimisations that are clear next steps. Improvements to the modularity of the code, making even greater use of the object orientated nature of C++ to make the code more readable and more efficient, should be implemented next. For instance, the function that currently calculates the residual is unnecessarily large, and can be broken up. This will reduce code repetition, and make future adaptation easier. Memory usage could also be slimmed down by adding more explicit memory allocation and de-allocation routines, among other simple changes, such as reducing the number of tracked variables.

As it currently stands, I have reached the practical limits of the tests that I can run with the current implementation. Further testing and analysis requires more detailed results, which currently require very long compute times to achieve. Some work in extension and optimisation is still required to produce a solver that can compete with the current state-of-the-art gravo-hydrodynamics codes. Distributed memory parallelism, using MPI, is an obvious potential avenue for further development, as most large scale simulation work now requires high performance computing capabilities, which are mostly run on distributed memory systems. To speed this process, it would be optimal to take advantage of the bookkeeping provided by an established multi-physics code, such as GIZMO, as a numerical backbone for future development of the solver. I will discuss this further in Section 6.2.2.

Alongside this work, there is still a lot to learn about the possibilities of alternative distributions schemes. Those that I have covered here are the most widely used, but a number of other formulations exist. The possibilities for different blending schemes, from different bending coefficient calculators to blending different distributions schemes, present a host of opportunities. The mass matrix, present in the second order formulations of some schemes, presents another element to investigate further. A number of mass matrices have been developed for these solvers, and those not discussed here could shape the implementation

in positive ways. Most importantly, they could provide a mechanism to reduce numerical diffusion. Building on the discussion of gravity, one could develop a simple mechanism by which new source and sink terms can be easily added to method. This would open the door to including a multitude of processes, from radiative cooling, to star formation itself, to star formation feedback, and more. The work I have presented, in the two chapters covering the RD solver, represents a comprehensive framework onto which I can implement any number of additional features.

6.2.2 Moving Mesh

The implementation, in its current form, is built around a static unstructured mesh of triangles. The residual distribution approach is fundamentally built around such an unstructured mesh, but can be applied to structured meshes, such as that built from a Cartesian grid of points. While other solvers, which utilise dimensional splitting, can be transformed to work with unstructured meshes, the RD approach does so innately. This makes it ideal for conversion into a moving mesh method. With a moving mesh, the vertices of the mesh move with the flow, making structured meshes impractical. Unstructured meshes, often using the dual of the Delaunay triangulation, the Voronoi tessellation, can be used instead. Since the mesh is assumed to be unstructured with the RD approach, the conversion of the current formulation to its moving mesh counterpart only requires that the Eulerian fluid equations be reformulated into their arbitrary Lagrangian Eulerian (ALE) formulation (Michler et al., 2003). With the moving mesh providing natural resolution refinement in regions of high density, and the truly multi-dimensional modelling of the fluid flows that the RD approach achieves, the combination could provide a powerful new tool for running astrophysical simulations.

As mentioned before, one could implement the RD hydro solver as a new module for an established code, such as GIZMO, allowing the RD solver to handle the hydrodynamics of a given simulation, while the rest of the physics is modelled using the advanced processes, already present in the other code. A state-of-the-art multi-physics code can provide key bookkeeping mechanisms, such as tree construction, that will allow the new formulation to function efficiently. The advanced gravity calculations, utilised by such codes, could be used as an optimised and more accurate gravity model for the RD solver. Once this is done,

the effects of various additional physical processes, can be applied. Using the example of GIZMO, advanced models for a wide range of processes and feedback mechanisms are already implemented, and so their contribution to the state of the gas can be easily applied. These can be included as source and sink terms, much like gravity.

Appendix A

RD Implementation

In this appendix, I will describe how I have implemented the underlying numerical method, into a functioning program to solve the evolution of any set of initial conditions (ICs). This will cover the 2D form of the solver, which finds the numerical approximation for the solution to the 2D Eulerian inviscid fluid equations. This solution is found in a domain discretised by a set of vertices, around which an unstructured Delaunay mesh is constructed. The code is written in C++, with extensive use of the languages object-oriented nature allowing for significant modularity. The core of the implementation is built around two classes: one for the vertices of the mesh, and one for the triangular elements of the mesh. These are supplemented by a number of bookkeeping and linear algebra functions. The code is publicly available at via GitHub (https://github.com/bpfm/rdsolver), but is not packaged for easy distribution. There are only very limited installation and user instructions available at this time.

A.1 Vertex

The `VERTEX` class stores the position of the vertex, and the fluid state at that position, including both primitive and conserved variable forms. It can be found in the `vertex2D.h` file. To handle the second order in time extensions, each vertex also has information about the intermediate state used by the RK2 time stepping mechanism. This class also consists of various member functions that update the state and intermediate state, when passed the appropriate updates from the

triangles for with they are vertices. In the current setup, the vertices do not hold information about which triangles these are, or how many triangles with which they are associated. Finally, this class has a member function for checking that the fluid state is viable, namely that it does not have negative density or pressure. This is an error handling function for nonphysical solutions. In the event that such a state is reached, the code will exit. The set of vertices, around which the mesh will be constructed, are held in a std::vector `RAND_POINTS` structure, which allows for simple insertion and deletion of new vertices.

The class contains four key member functions that handle the updating of the fluid state. These are divided in two , with two functions that keep track of the contributions the update for a given time step, and two that update the state once all contributions have been made. There are two of each because one is designed for creating the intermediate state needed by the RK2 time-stepping regime, and one is required for constructing the final state at the end of the update. The `update_du_half` function is passed updates by the `pass_update_half` member function of the `TRIANGLE` class. It keeps a running total of the updates passed to it during a time-step. Once updates have been passed by all the triangles in the mesh, the `update_u_half` member function updates the initial state with the total change, generating the intermediate state. Assuming a second order scheme is being used, then the updates from the intermediate to final states are passed from the triangles to the `update_du` function, which again keep a running total for a given time-step. After all elements have does this, the update is performed by the `update_u_variables` function. If a first order scheme is being used, then the update passed from each element to `update_du` is forced to zero. This means that the update from initial to final state is entirely found by the update to the intermediate state, which is exactly equivalent to the formulation of the first order update. Once the fluid state has been updated at a given vertex, the update trackers for that vertex are reset to zero, ready for the next time-step.

A.2 Triangle

The second class, which handles the elements of the mesh, is the `TRIANGLE` class. The key information that this class is initialised with, are the pointers to the three vertices that define it. These are pointers to entries in the vector of `VERTEX` objects. Alongside the basic functions for importing the positions and states of the component vertices, the `TRIANGLE` class contains the member

functions for calculating and distributing the element residual. The element residual is calculated by member function `calculate_first_half`, which imports the current state, constructs the Roe parameter vector and subsequent K-matrix and element residual. This residual is distributed by the `pass_update_half` member function, which calculates the update to the fluid state at each vertex of the element based on the time step Δt passed to the function, and the chosen distribution scheme. The distribution of the update is handled by calling the `update_du_half` function, described in the previous subsection, for each vertex of the triangle. Again assuming a second order solver is being used, the total residual is calculated by `calculate_second_half`. This performs the same task as the first function, but now for the total residual, rather than the element residual. Once again this is used to calculate the update distributed to the vertices, this time calculating the change from the intermediate state to the final state. The update is passed to the vertices by calling `update_du` for each vertex. As mentioned above, the first order schemes simply force the update passed to `update_du` to be zero.

A.3 Additional Functions

Alongside these fundamental structures are a number of supplementary functions that perform the standard bookkeeping tasks such as I/O, scenario setup, some linear algebra functions, and the actual running of the program. These are included in the `io.cpp`, `setup2D.cpp`, `inverse.cpp`, and `main.cpp` files. With the main code is a sub-module, in `cgal_periodic2D.cpp`, that can construct the required triangulation for the main RD solver code. The desired scenario, and various constants such as domain size and CFL factor, are stored in the parameter file `constants.h`, and included at compile time in the main program.

A.3.1 Input/Output Functions

The code includes basic functions for reading in sets of vertices, and the corresponding Delaunay triangulation, in ASCII format. When the vertices are read in, the setup function is called to set the appropriate fluid state, and then to setup the triangles of the mesh. Once the evolution of the fluid is started, snapshot files that contain all the information about the fluid state at each vertex, at that

time, are created. These are produced by the `write_snap` function, which creates an ASCII snapshot file, which are then used to analyse the results.

A.3.2 Setup

The setup functions are called when a mesh of triangles and vertices are read in from the input file. Based on the conditions set out in the parameter file (see below), the primitive variables are set for each vertex, and the functions that convert these to their conserved counterparts are called. Once all states have been setup, the mesh of triangles is read in, and the triangle setup function is called for each element. As the vertices associated to each element are set, the corresponding geometric properties of that element are calculated, namely the area, the contribution to the dual area of the associated vertices, and the inward pointing normals of each edge.

A.3.3 Matrix Inversion

Part of the construction of both the LDA and N schemes is the inversion of the sum of negative inflow matrices. The inversion of a square matrix is only possible if the matrix is not singular, which is when the determinant is zero. Numerically calculating the inverse of a square matrix of arbitrary size is not straight forward, and is a computationally intense task. There are a number of well established linear algebra C++ libraries which can handle this calculation. I use OpenBLAS library (https://www.openblas.net/), which provides a highly optimised implementation of the BLAS linear algebra libraries, with an extensive manual, and many examples. The OpenBLAS functions are called from the `mat_inv` function, which, in turn, is called by the residual calculation member functions of the `TRIANGLE` class.

A.3.4 Evolution Bookkeeping

The `main` function, as usual, ties everything else together. It handles the vectors that contain the triangles (`RAND_MESH`) and vertices (`RAND_POINTS`), and calling the input and setup functions to build the ICs for the current scenario. Once this is done, `main` runs through the evolution of the fluid, looping over time steps until

the final time is reached. Within this loop are nested loops that call the residual cacluation functions for the first and second order updates for every element, and for updating the states based on these residuals.

A.3.5 Delaunay Triangulation

In its current state, the code includes a module that can construct 2D and 3D Delaunay periodic triangulations for arbitrary distributions of vertices, a Cartesian grid of vertices, and an offset grid of vertices. In the triangulation directory, there are a number of options for constructing the various setups. The key code is built around functions provided by the CGAL library, in `cgal_periodic2D.cpp`. This program sets up the desired distribution of vertices, creates the CGAL mesh constructor, and pass the set of vertices to the constructor. This produces a list of triangles, where each triangle is defined by three integers. These integers are the indices that refer to vertices in the original set. All of this information is output to an ASCII file, `Delaunay2D.txt`, which is used is read in when the main code is run. This program will be eventually integrated into the main code, to allow for on the fly mesh refinement, and eventually for the possibility of a moving mesh.

A.3.6 Global Parameters

A number of global parameters and scenario definitions are declared in the `constants.h` header file. This includes defining the test case that is being used, the number of dimensions (see Chapter 4 for 3D extension), the number of snapshots files, the type of boundaries, the choice of distribution scheme, and the temporal accuracy order. Depending on the choice of test case, global parameters for box size, CFL coefficient, adiabatic gas constant and total time of simulation, are also defined. At the moment, these are included at compile time, but in future I plan to allow for run time setting of some of these variables.

A.4 3D Solver Implementation

The 3D solver implementation follows closely from the 2D equivalent. The `VERTEX` and `TRIANGLE` classes have 3D counterparts, with the addition of the

extra dimension in coordinates, fluid state etc. Other than the updates to the fundamental bookkeeping functions, such setter and getter functions, and the various conversion functions, the main change is to the exact form of the inflow matrix calculation, which is converted to the 3D form given above. The conversion of most other parts of the calculation is simply achieved by looping over five elements that correspond to the underlying fluid equations, where before there were only four, and including the contribution from four vertices, rather than three. The structure of the code was designed with this eventual extension in mind, so the change in code structure is minimal. For instance, the same inversion function can be passed either the 4×4 or 5×5 inflow matrix.

A.4.1 Geometry

A number of geometric extensions must be performed to construct the analogous setup, particularly in the definition of required time step, and inward facing normal. The former is addressed in Section 4.3.3. The main geometric change in the calculation is moving from the normal to an edge to the normal to a face. Calculating this normal, and guaranteeing that it is the inward facing normal for the host tetrahedron, is more complicated than the equivalent in 2D. It is sufficient, in 2D, to require that the vertices are ordered counter-clockwise, and then calculate the normal with vertices ordered in this way to find the inward facing normal. In 3D, an equivalent ordering is not obvious, so instead, I calculate the normal based on the arbitrary order of vertices produced by the triangulation. Taking a tetrahedron with vertices $\mathbf{v} = (v_0, v_1, v_2, v_3)$, the triangular face opposite v_0 has normal can be found from the cross product of two of its edges

$$\mathbf{n}_0 = \mathbf{r}_{12} \times \mathbf{r}_{13} \tag{A.1}$$

where $\mathbf{r}_{12}$ and $\mathbf{r}_{13}$ are the edges between v_1 and v_2, and v_1 and v_3, respectively. I then check that it points inwards, with respect to the host, by finding the angle θ between the normal and one of the edges between the face and the opposite vertex. This can be done by using the dot product of the two vectors

$$\cos(\theta) = \frac{\mathbf{n}_0 \cdot \mathbf{r}_{01}}{|\mathbf{n}_0||\mathbf{r}_{01}|}. \tag{A.2}$$

If this angle is obtuse $\theta > \pi/2$, then the normal is pointing in the wrong direction, and is simply flipped. This process is repeated for all faces, and guarantees the

desired orientations. Other than normals, the dual area is replaced by the dual volume, which is trivial to calculate.

A.4.2 Triangulation

The shift to 3D requires a mechanism to produce arbitrary Delaunay triangulations of vertices in three dimensions. The definitions of a Delaunay triangulation essentially remains the same (see Section 3.3), except now the empty circumdisk becomes an empty circumsphere. The same strategies can be applied to construct such a triangulation (I still refer to it as a triangulation, even though the space is now divide into tetrahedrons). As before, I utilise the `CGAL` triangulation library to construct the underlying mesh, within which I compute the residual distribution solution.

It is obvious that 3D simulations will be significantly more computationally costly than the same resolution simulation in 2D. Other than the typical scaling problem encountered by all 3D simulations, whereby the number of cells/particles increases with N^3, methods built around simplex meshes also have an increased ratio of vertices to simplices when in 3D. To visualise this, one can consider a Cartesian grid of vertices. A square of four vertices describe one square cell, or two triangular cells. The ratio of vertices to simplices is therefore $1 : 2$. In 3D, a set of eight vertices describe a cube cell, but contains six tetrahedrons, giving a ratio of $1 : 6$. The exact ratio only applies for the Cartesian grid, but on average the ratio is about three times larger in 3D. Therefore not only does go from N^2 to N^3, we must also contend with more volume elements per vertex, increasing the computational cost further.

Bibliography

Aarseth S. J., 1963, MNRAS, 126, 223

Abel T., Anninos P., Zhang Y., Norman M. L., 1997, New Astron., 2, 181

Abel T., Bryan G. L., Norman M. L., 2002, Science, 295, 93

Abgrall R., 2006, Computers & Fluids, 35, 641

Abgrall R., 2012, Comm. Comput. Phys., 11, 1043

Abgrall R., Marpeau F., 2007, J. Sci. Comput., 30, 131

Abgrall R., Roe P., 2003, J. Sci. Comput., 19

Abgrall R., Santis D. D., 2015, J. Comput. Phys., 283, 329

Adhikari S., Dalal N., Clampitt J., 2016, J. Cosmology Astropart. Phys., 2016, 022

Agertz O., et al., 2007, MNRAS, 380, 963

Anninos P., Norman M. L., 1996, ApJ, 460, 556

Arth A., Donnert J., Steinwandel U., Boss L., Halbesma T., Putz M., Hubber D., Dolag K., 2019, WVTICs – SPH initial conditions for everyone (`arXiv:1907.11250`)

Barnes J., Hut P., 1986, Nature, 324, 446

Beckman J., Carretero C., Vazdekis A., 2008, Chinese J. Astron. Astrophys., 8, 77

Beckmann R. S., Slyz A., Devriendt J., 2018, MNRAS, 478, 995

Begum A., Chengalur J. N., Karachentsev I. D., Sharina M. E., Kaisin S. S., 2008, MNRAS, 386, 1667

Berger M. J., Oliger J., 1984, J. Comput. Phys., 53, 484

Bernal C. G., Sánchez-Salcedo F. J., 2013, ApJ, 775, 72

Bertschinger E., 1998, ARA&A, 36, 599

Binney J., Tremaine S., 1987, Galactic dynamics. Princeton University Press

Bode P., Ostriker J. P., Turok N., 2001, ApJ, 556, 93

Bodenheimer P., Laughlin G., Rozyczka M., Plewa T., Yorke H., Yorke H., 2006, Numerical Methods in Astrophysics: An Introduction. Series in Astronomy and Astrophysics, CRC Press, https://books.google.co.uk/books?id=qWbLBQAAQBAJ

Bottrell C., Torrey P., Simard L., Ellison S. L., 2017, MNRAS, p. stx017

Boylan-Kolchin M., Ma C.-P., Quataert E., 2008, MNRAS, 383, 93

Bromley B. C., Kenyon S. J., 2016, ApJ, 826, 64

Bromm V., 2013, Rep. Prog. Phys., 76, 112901

Bromm V., Yoshida N., Hernquist L., McKee C. F., 2009, Nature, 459, 49

Bryan G. L., et al., 2014, ApJS, 211, 19

Caraeni D., Fuchs L., 2002, Theo. Comput. Fluid Dyn., 15, 373

Chandrasekhar S., 1943, ApJ, 97, 255

Chandrasekhar S., 1961, Hydrodynamic and hydromagnetic stability. Courier Corporation

Cheng S., Dey T., Shewchuk J., 2016, Delaunay Mesh Generation. Chapman & Hall/CRC Computer and Information Science Series, CRC Press, https://books.google.co.uk/books?id=oJ3SBQAAQBAJ

Chitre A., Jog C. J., 2002, A&A, 388, 407

Clark P. C., Bonnell I. A., Zinnecker H., Bate M. R., 2005, MNRAS, 359, 809

Cohen G., Joly P., Roberts J. E., Tordjman N., 2001, SIAM J. Num. Ana., 38, 2047

Colless M., et al., 2001, MNRAS, 328, 1039

Crowther P. A., 2001, Ap&SS, pp 215–230

Csik A., Ricchiuto M., Deconinck H., 2002, J. Comput. Phys., 179, 286

DESI 2016, The DESI Experiment Part I: Science,Targeting, and Survey Design (arXiv:1611.00036)

Daddi E., et al., 2010, ApJ, 714, L118

Deconinck H., Ricchiuto M., 2007, Encyclopedia of Computational Mechanics, p. 1

Deconinck H., Roe P., Struijs R., 1993, Computers & Fluids, 22, 215

Dobes J., Deconinck H., 2008, J. Comput. App. Math., 215, 378

Donnert J. M. F., Beck A. M., Dolag K., Röttgering H. J. A., 2017, MNRAS, 471, 4587

Doroshkevich A., Tucker D. L., Allam S., Way M. J., 2004, å, 418, 7

Dosopoulou F., Antonini F., 2017, ApJ, 840, 31

Duffell P. C., 2016, ApJS, 226, 2

Duffell P. C., MacFadyen A. I., 2011, ApJS, 197, 15

Efstathiou G., Davis M., White S. D. M., Frenk C. S., 1985, ApJS, 57, 241

El-Zant A. A., Kim W.-T., Kamionkowski M., 2004, MNRAS, 354, 169

Emsellem E., et al., 2011, MNRAS, 414, 888

Erwin P., 2019, MNRAS, 489, 3553

Fan X., et al., 2006, AJ, 132, 117

Feng Y., Matteo T. D., Croft R., Tenneti A., Bird S., Battaglia N., Wilkins S., 2015, ApJ, 808, L17

Ferguson A. M. N., Irwin M. J., Ibata R. A., Lewis G. F., Tanvir N. R., 2002, AJ, 124, 1452

Ferland G. J., et al., 2013, Rev. Mex. Astron. Astrofis., 49, 137

Filippenko A. V., 2005, in Humphreys R., Stanek K., eds, ASP Conf. Ser.Vol. 332, The Fate of the Most Massive Stars. p. 34 (`arXiv:astro-ph/0412029`)

Fryxell B., et al., 2000, ApJS, 131, 273

Fujii M., Funato Y., Makino J., 2006, PASJ, 58, 743

Gaburov E., Nitadori K., 2011, MNRAS, 414, 129

Galli D., Palla F., 1998, A&A, 335, 403

Geha M., Blanton M. R., Yan R., Tinker J. L., 2012, ApJ, 757, 85

Gingold R. A., Monaghan J. J., 1977, MNRAS, 181, 375

Glimm J., 1965, Comm. Pure App. Math., 18, 697

Godunov S. K., Bohachevsky I., 1959, Matematicheskii Sbornik, 47(89), 271

Harten A., 1983, J. Comput. Phys., 49, 357

Hernquist L., Katz N., 1989, ApJS, 70, 419

Hillebrandt W., Niemeyer J. C., 2000, ARA&A, 38, 191

Hinshaw G., et al., 2007, ApJS, 170, 288

Hockney R., Eastwood J., 1988, Computer Simulation Using Particles. CRC Press, https://books.google.co.uk/books?id=nTOFkmnCQuIC

Hollenbach D., McKee C. F., 1979, ApJS, 41, 555

Holmes M., 2007, Introduction to Numerical Methods in Differential Equations. Vol. 52, Springer, doi:10.1007/978-0-387-68121-4

Hopkins P. F., 2013, MNRAS, 428, 2840

Hopkins P. F., 2015, MNRAS, 450, 53

Hopkins P. F., Quataert E., Murray N., 2012, MNRAS, 421, 3522

Hopkins P. F., Kereš D., Oñorbe J., Faucher-Giguère C.-A., Quataert E., Murray N., Bullock J. S., 2014, MNRAS, 445, 581

Hubbard M., Baines M., 1997, J. Comput. Phys., 138, 419

Hubble E., 1929, Proc. Nat. Acad. Sci., 15, 168

Johnson H. L., 1966, ARA&A, 4, 193

Just A., Kegel W. H., 1990, A&A, 232, 447

Just A., Khan F. M., Berczik P., Ernst A., Spurzem R., 2010, MNRAS, 411, 653

Katz N., Weinberg D. H., Hernquist L., 1996, ApJS, 105, 19

Khochfar S., Silk J., 2009, MNRAS, 397, 506

Khochfar S., Silk J., 2010, MNRAS, 410, L42

Kim H., Kim W., 2007, ApJ, 665, 432

Kim H., Kim W.-T., 2009, ApJ, 703, 1278

Kim W.-T., El-Zant A. A., Kamionkowski M., 2005, ApJ, 632, 157

Kuffmeier M., HaugbÃÿlle T., Nordlund â., 2017, ApJ, 846, 7

Landau L., Lifšic E., Lifshitz E., Pitaevskii L., Sykes J., Kearsley M., 1980, Statistical Physics: Theory of the Condensed State. Course of theoretical physics, Elsevier Science

Lanson N., Vila J. P., 2008, SIAM J. Num. Ana., 46, 1912

Lax P. D., 1957, Comm. Pure App. Math., 10, 537

LeVeque R. J., 2002, Finite Volume Methods for Hyperbolic Problems. Cambridge Texts in Applied Mathematics, Cambridge University Press, doi:10.1017/CBO9780511791253

Leer B. V., 1984, SIAM J. Sci. Stat. Comput., 5

Lucy L. B., 1977, AJ, 82, 1013

Ma X., et al., 2018, MNRAS, 478, 1694

Martig M., et al., 2013, MNRAS, 432, 1914âĂŞ1927

Martinsson T. P. K., Verheijen M. A. W., Westfall K. B., Bershady M. A., Schechtman-Rook A., Andersen D. R., Swaters R. A., 2013, A&A, 557, A130

Mathews W. G., Brighenti F., Buote D. A., Lewis A. D., 2003, ApJ, 596, 159

McDowell M. R. C., 1961, The Observatory, 81, 240

McGaugh S. S., de Blok W. J. G., 1997, ApJ, 481, 689

McNally C. P., Lyra W., Passy J.-C., 2012, ApJS, 201, 18

Michler C., De Sterck H., Deconinck H., 2003, Computers & Fluids, 32, 59

Mistani P. A., et al., 2015, MNRAS, 455, 2323

Mo H., van den Bosch F. C., White S., 2010, Galaxy Formation and Evolution. Cambridge University Press

Monaghan J. J., 1992, ARA&A, 30, 543

Moore B., Ghigna S., Governato F., Lake G., Quinn T., Stadel J., Tozzi P., 1999, ApJ, 524, L19

Muto T., Takeuchi T., Ida S., 2011, ApJ, 737, 37

Namouni F., 2010, MNRAS, 401, 319

Nelson D., et al., 2018, The IllustrisTNG Simulations: Public Data Release (`arXiv:1812.05609`)

Nishiyama K., Nakai N., 2001, PASJ, 53, 713

Noh W. F., 1987, J. Comput. Phys., 72, 78

O'Shea B. W., Norman M. L., 2007, ApJ, 654, 66

Ogiya G., Burkert A., 2016, MNRAS, 457, 2164

Omukai K., 2000, ApJ, 534, 809

Omukai K., Tsuribe T., Schneider R., Ferrara A., 2005, ApJ, 626, 627

Ostriker E. C., 1999, ApJ, 513, 252

Paardekooper S. J., 2017, MNRAS, 469, 4306

Paillere H., Deconinck H., Roe P., 1995, 12th Computational Fluid Dynamics Conference, p. 592

Palma P. D., Pascazio G., Rossiello G., Napolitano M., 2005, J. Comput. Phys., 208, 1

Parodi B. R., Binggeli B., 2003, A&A, 398, 501

Peacock J. A., et al., 2001, Nature, 410, 169

Peebles P. J. E., 1980, The large-scale structure of the universe. Princeton University Press

Perets H. B., Biham O., 2006, MNRAS, 365, 801

Planck Collaboration et al., 2016, A&A, 594, A1

Plummer H. C., 1911, MNRAS, 71, 460

Potter D., Stadel J., Teyssier R., 2017, Comput. Astr. Comso., 4, 2

Racca G. D., et al., 2016, in MacEwen H. A., Fazio G. G., Lystrup M., Batalha N., Siegler N., Tong E. C., eds, Vol. 9904, Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave. SPIE, pp 235–257, doi:10.1117/12.2230762, https://doi.org/10.1117/12.2230762

Rafikov R. R., 2017, ApJ, 837, 163

Ricchiuto M., Abgrall R., 2010, J. Comput. Phys., 229, 5653

Rider W. J., 2000, J. Comput. Phys., 162, 395

Riess A. G., et al., 1999, AJ, 118, 2675

Robaina A. R., Bell E. F., van der Wel A., Somerville R. S., Skelton R. E., McIntosh D. H., Meisenheimer K., Wolf C., 2010, ApJ, 719, 844

Roberts M. S., Hogg D. E., Bregman J. N., Forman W. R., Jones C., 1991, ApJS, 75, 751

Robertson B. E., Kravtsov A. V., Gnedin N. Y., Abel T., Rudd D. H., 2010, MNRAS, 401, 2463

Roe P., 1981, J. Comput. Phys., 43, 357

Rossiello G., Palma P. D., Pascazio G., Napolitano M., 2009, Computers & Fluids, 38, 1384

Sánchez-Salcedo F. J., Brandenburg A., 1999, ApJ, 522, L35

Sánchez-Salcedo F. J., Brandenburg A., 2001, MNRAS, 322, 67

Schaye J., et al., 2015, MNRAS, 446, 521

Sedov L. I., 1959, Similarity and Dimensional Methods in Mechanics. New York Academic Pres

Simon-Diaz S., Stasinska G., 2008, MNRAS, 389, 1009

Smith B., Sigurdsson S., Abel T., 2008, MNRAS, 385, 1443

Smith B. D., et al., 2016, MNRAS, 466, 2217

Sod G. A., 1978, J. Comput. Phys., 27, 1

Sohn S.-I., 2005, Computers and Mathematics with Applications, 50, 231

Somerville R. S., Davé R., 2015, ARA&A, 53, 51

Spergel D. N., et al., 2003, ApJ, 148, 175

Springel V., 2005, MNRAS, 364, 1105

Springel V., 2010, MNRAS, 401, 791

Springel V., Hernquist L., 2003, MNRAS, 339, 289

Stone J. M., Gardiner T. A., Teuben P., Hawley J. F., Simon J. B., 2008, ApJS, 178, 137

Struijs R., Deconinck H., Roe P. L., 1991, Fluctuation splitting schemes for the 2D Euler equations. Kluwer Academic Publishers

Sweby P. K., 1984, SIAM J. Num. Ana., 21, 995

Tacconi L. J., et al., 2010, Nature, 463, 781

Tagawa H., Saitoh T. R., Kocsis B., 2018, Phys. Rev. Lett., 120, 261101

Taylor J., 2005, Classical Mechanics. University Science Books, `https://books.google.co.uk/books?id=P1kCtNr-pJsC`

Teyssandier J., Terquem C., Papaloizou J. C. B., 2012, MNRAS, 428, 658

Teyssier R., 2002, A&A, 385, 337

Thun D., Kuiper R., Schmidt F., Kley W., 2016, A&A, 589, A10

Tittley E. R., Pearce F. R., Couchman H. M. P., 2001, ApJ, 561, 69

Toro E. F., Spruce M., Speares W., 1994, Shock Waves, 4, 25

Toyouchi D., Hosokawa T., Sugimura K., Kuiper R., 2020, Gaseous dynamical friction under radiative feedback: do intermediate-mass black holes speed up or down? (`arXiv:2002.08017`)

Villedieu N., Quintino T., Ricchiuto M., Deconinck H., 2011, J. Comput. Phys., 230, 4301

Vogelsberger M., et al., 2014a, MNRAS, 444, 1518

Vogelsberger M., et al., 2014b, Nature, 509, 177

White S. D. M., 1994, Formation and Evolution of Galaxies: Les Houches Lectures (arXiv:astro-ph/9410043)

White M., 2014, MNRAS, 439, 3630

White S. D. M., Rees M. J., 1978, MNRAS, 183, 341

Woltjer L., 1972, ARA&A, 10, 129

Woosley S., Janka T., 2005, Nature, 1, 147

Yoachim P., Dalcanton J. J., 2006, AJ, 131, 226

Zamora-Aviles M., Vazquez-Semadeni E., Colin P., 2012, ApJ, 751, 77

Zel'Dovich Y. B., 1970, A&A, 500, 13

Zhao H., 2004, MNRAS, 351, 891

van Leer B., 1979, J. Comput. Phys., 32, 101

van Leer B., 2006, Comm. Comput. Phys., 1, 192

van den Bergh S., 2009, ApJ, 702, 1502

van den Bosch F. C., Tormen G., Giocoli C., 2005, MNRAS, 359, 1029