Contour advection tracer method in smoothed particle hydrodynamics

by

Tiffany M. Fields

A Thesis Submitted to Saint Mary's University, Halifax, Nova Scotia in Partial Fulfillment of the Requirements for the Degree of Masters of Science in Astronomy (Department of Astronomy and Physics)

August 15, 2019, Halifax, Nova Scotia

Approved: Dr. Robert Thacker

Advisor

Approved: Dr. Ian Short

Committee Member

Approved: Dr. Luigi Gallo

Committee Member

Date: August 15, 2019.

Acknowledgements

Many thanks are necessary to give to my thesis supervisor, Dr. Rob Thacker, for helping me get through every step of this process. He has been an incredible mentor over the last few years and has been immensely supportive and helpful and I cannot thank him enough for his constant understanding. Without him, there would be no finished thesis at all. Thanks also to my committee members, Dr. Luigi Gallo and Dr. Ian Short, for their time, help, and support. I appreciate the entire department – faculty, staff, and students – for supporting me and including me throughout both my BSc and now my MSc. From making posters, working at the Burke-Gaffney Observatory, to participating in photoshoots and television interviews, I feel so lucky to have spent the last six years in this department and here in Halifax.

Sincere appreciation for the free services offered by the Counseling Centre here at SMU is also necessary. Mental health often takes a backseat to research and work, but it is extremely important and I hope that others can encourage themselves to use their services, or at least reach out to a friend, if they're having difficulties. There should be no shame in promoting your own mental well-being, and no one can do their best work if their mental health is not in order.

Final thanks, of course, to my supportive family, both in Halifax and in the US, friends, my partner, my dog(s), and all of the students that I've met along the way that have had such a substantial impact on my life.

Contents

List of Figures

List of Tables

Abstract

Contour advection tracer method in smoothed particle hydrodynamics

by Tiffany M. Fields

Studying the evolution of galaxies is important in understanding many other astrophysical phenomenon. However, this process required numerical work due to long timescales over which evolution occurs. To study the effects of chaotic mixing within a galactic disk, we implemented a contour advection method using tracers to follow the flow of gas within a disk. In this thesis, we examined effects of integration properties such as timestep normalization, artificial viscosity, temperature ceiling, and more to determine ways that our tracer method produced errors during evolution. We found that the primary issue with the tracer method was how tracers experienced shock regions, or regions of steep density contrasts. When tracers experienced too much shock heating, they overheated and were ejected from the disk, but when experiencing too little shock heating, they flowed through shocks without feeling the effects. Both of these instances led to incorrect following of the gas flow.

August 15, 2019

Chapter 1

Introduction

The evolution of galaxies is a complex yet fascinating branch of astronomy. Until the advent of fast digital computers, progress using analytical techniques was moderate and often limited to idealized or steady state solutions (Binney & Tremaine 1994). However, by dealing directly with dynamics and evolutions over millions to billions of years, numerical simulations have greatly aided in developing theories of galactic evolution (Baugh 2008) and are required to see the full picture of such evolution.

Despite the power of numerical simulations, understanding how galaxies form and evolve over time is a challenging and unsolved area of study in astrophysics. To study how galaxies form, evolve, and merge, one must consider not only the properties of the galaxies themselves, but also the environment in which they are located (Baugh 2006). The current accepted theory is that galaxies evolve hierarchically through gravitational instability (Press & Schechter 1974; White & Rees 1978; Lacey & Cole 1993; Baugh 2006), where over time mergers progressively increase the mass of halos. However, high-mass halos are more rare

than low-mass. This theoretical model is natural within cosmologies that include cold dark matter (CDM) and a cosmological constant Λ, and is called the ΛCDM model.

Dark matter halos eventually host galaxies as gas falls into and cools in the centers of these halos (Press & Schechter 1974; Lacey & Cole 1993; Springel et al. 2006). The halos, originally created by a rotation-free initial perturbation, acquire their angular momentum through gravitational interactions, specifically tidal torques (Peebles 1969; Vitvitska et al. 2002). The overall spin of the dark matter halos can be characterized by a dimensionless spin parameter λ_0 in the following way

$$\lambda_0 \equiv \frac{J|E|^{1/2}}{GM^{5/2}} \tag{1.1}$$

where J is the total angular momentum of the halo, E is the total energy, G is the gravitational constant, and M is the mass of the halo (Springel & White 1999). This rotational behaviour is also shared by the baryons, the collapse of which often leads to disk galaxies (Book et al. 2011). The contractions that are experienced by the baryons that cool is larger than the experienced by the dark matter, so the galaxies appear smaller and rotate coherently for disks.

The galaxies that form within the dark matter halos have many observable phenomena. The stellar component of galaxies can provide information about the evolution of the galaxy via the stellar formation rate and history, as well as the chemical composition and abundance of certain elements from the spectra. The total luminosity from the stellar and ISM components of a galaxy can be measured, while the total mass (including dark matter) can

be inferred by measuring the velocities of baryonic matter within the disk. These properties mentioned above all depend on the evolution of a given galaxy, which itself depends on the amount of dynamical mixing that is happening within it. Note that using observational parameters to relate to the properties of the dark matter halo is difficult, as dark matter is both collisionless and does not interact with electromagnetic radiation, though methods like gravitational lensing can be used to determine the size of an unknown dark matter halo within the line-of-sight of a distant, massive galaxy.

In terms of galactic dynamics and chaos, initial studies by Henon & Heiles (1964) described the motion of stars around the galactic centre, assuming motion in the x-y plane only and used a simple potential with only a four-dimensional phase space. What their results showed, however, is that there are certain sets of parameters that cause orbits to be chaotic suggesting that orbital motions in galaxies can be highly irregular. This relates to galaxy evolution directly, since the amount of turbulent mixing within a galaxy can affect the observable properties. Orbits of stars in a galaxy are affected by chaotic evolution, and large amounts of mixing can affect the star formation rate and thus the overall luminosity of a galaxy. The interstellar medium (ISM) is also affected by the amount of chaos, as the chemistry and temperature can be affected by the stellar formation rate (Baugh 2006; Federrath et al. 2008), and chaotic evolution can mixing stellar distributions, affecting the evolution of a disk and star formation history. While a great deal of research has been done on the evolution of galaxies, many unanswered questions still remain, not least of which is how well do simulated models reproduce variance in observed relationships.

Although chaos may not be directly measurable in any set of observations due to long

timescales involved, the impact of chaos in numerical simulations is measurable. Thus, studying the chaotic mixing of astronomical objects such as galaxies is highly relevant and important to predictions made from theoretical models motivated by numerical work.

1.1 Types of chaos: Hamiltonian and non-Hamiltonian chaos

Different types of chaotic systems exist and are relevant for different physical and astrophysical phenomenon. In general, chaos is a type of movement that lies between regular trajectories that can be integrated and a random state of noise Goldstein et al. (2001). More specifically, two initially similar but infinitesimally different configurations of phase space are said to be chaotic if the trajectories of those configurations vary widely over time. This behaviour is not random, but rather deterministic as the trajectories can be numerically integrated, though they are not predictable if measurement uncertainties are present as the behaviour depends critically on the initial conditions (Goldstein et al. 2001). These chaotic trajectories come from nonlinear and nonperiodic systems, and specific solutions to these chaotic systems change exponentially in response to minute changes in the initial properties (Eckmann & Ruelle 1985; Goldstein et al. 2001).

In systems with perturbations, the Kolmogorov-Arnold-Moser (KAM) theorem states that if the perturbation is small and the frequencies of motion in the integrable Hamiltonian are incommensurate, then the motion will be confined to a torus (Kolmogorov 1954; Arnold 1963; Moser 1962). As the perturbations become larger, the effect on the motion of the system becomes more pronounced. When the perturbation becomes significantly large, the

system may become chaotic. In these cases where the KAM theorem does not hold, chaos can occur (Goldstein et al. 2001).

Chaotic trajectories have motion which mixes, produces quasiperiodic orbits, and is extremely sensitive to initial conditions. Such systems encapsulate the "butterfly effect"[1] where a small, sometimes unknown, change in the initial conditions of a system can have a large-scale impact over time (Lorenz 1963; Goldstein et al. 2001). Two trajectories that are said to be chaotic will diverge exponentially in phase space, and a way to quantifiably measure that divergence is with the Lyapunov exponent, λ (Wolf et al. 1985; Goldstein et al. 2001). As given in Goldstein et al. (2001), the separation in phase space of two systems at a given time t can be written as

$$s(t) \sim s_0 e^{\lambda t} \tag{1.2}$$

where s_0 is the initial small separation between the two systems. When $\lambda > 0$, the system is chaotic and the timescale for growth is $\tau \sim 1/\lambda$. If λ is negative, then it gives the timescale for which the system approaches a regular attractor, and the system is not chaotic. Attractors are the path or point in which systems evolve toward, given sufficient time (Goldstein et al. 2001).

Two main types of chaotic systems exist: Hamiltonian and non-Hamiltonian systems. Hamiltonian systems are those which are conserved in phase space. In these systems, there are no attractors (Eckmann & Ruelle 1985; Goldstein et al. 2001).

However, in non-Hamiltonian chaos, energy can be input or dissipated from the system.

[1]This term comes from the thought experiment that just the small flap of butterfly wings can drastically change weather patterns in a distant location at a later time.

For systems that include drag, for example, the system may evolve towards a specific fixed point in phase space, which are called "fixed-point attractors." One example of a fixed-point attractor is the equilibrium position of a pendulum at rest. In higher dimensional systems, the attractors can have dispersed or disjointed structure and are called "strange attractors," one example of which is the Lorenz attractor. The Lorenz system was first described in Lorenz (1963) and arose from trying to model convection in the atmosphere. A system of ordinary differential equations was produced to model convection in the atmosphere, and many solutions to the Lorenz equations are chaotic. We focus on non-Hamiltonian aspects of galactic evolution in this thesis as we include dissipative effects.

1.2 Galactic stability in the presence of chaos

The implicit goal of this work is to explore the relationship between the local stability of a galaxy and how chaotic the hydrodynamic evolution of that galaxy is by mathematically describing the mixing of the interstellar medium (ISM) in simulations of disk galaxies.

The mixing of the ISM has consequences on turbulence and the formation of stars. Knowing how the ISM mixes during galaxy evolution can allow astronomers to better understand how a galaxy as a whole evolves over time, as well as how the process of star formation can be affected by ISM mixing. The overall amount of mixing can be estimated by the relative stretching of contours in the gas, which relates to the Lyapunov exponent, λ (Goldhirsch et al. 1987). Large values of λ mean rapid, chaotic mixing, while smaller values mean less rapid flows.

The stability of a disk galaxy is approximately described by the balance between the gravitational force that attracts objects to one another and the repulsive forces that prevent collapse, such as pressure or velocity dispersion. The Toomre Ratio, Q, provides a way to quantify the stability of a system (Toomre 1964). For a gas disk, Q can be expressed as

$$Q_{\rm gas} \equiv \frac{c_* \kappa}{\pi G \Sigma_g}, \tag{1.3}$$

where c_s is the speed of sound, which is related to the temperature; κ is the epicycle frequency, which determines the characteristic time of oscillation in the radial direction; G is the gravitational constant; and Σ_g is the surface density of the gas, which contributes to the local gravitational force (Toomre 1964). Additional work was done (Wang & Silk 1994; Rafikov 2001) to extend Q from its original form dealing with just a thin disk of gas to combining the contribution from both collisional and collisionless particles, namely both gas and stars. The updated form of Q including both gas and stars is

$$Q = \gamma \frac{\sigma_g \kappa}{\pi G \Sigma_g} \tag{1.4}$$

where

$$\gamma = \left(1 + \frac{\Sigma_* \sigma_g}{\Sigma_g \sigma_*}\right)^{-1}, \tag{1.5}$$

and σ_g and σ_* are the radial velocity dispersion of the gas and stars respectively and Σ_* is the surface density of the stars which contributes similar to the local gravitational force in equation (1.3) (Wang & Silk 1994; Rafikov 2001; Foyle 2007). Although researchers often

talk about Q being a global measurement, it is actually a local measurement and changes with radius. Therefore, Q can be considered to be a function of radius, $Q(r)$. When $Q(r) > 1$, the system is locally stable. When $Q(r) < 1$, the system is locally unstable and will likely collapse.

The way that λ changes for different disk galaxies made with varying initial conditions, and therefore differing values of Q, will be able to tell us if there is a relationship between the amount of mixing measured with λ and the level of stability that a galaxy has.

1.3 Numerical simulations

Following the large-scale temporal evolution of a galaxy cannot be done by simply observing a single galaxy in the sky as the time scales are too long. Consequently, astronomy relies upon inferring evolution from populations of different ages, which is a non-trivial issue and can be frought with problems such as the ecological fallacy where the nature of individual galaxies are derived by studying groups of galaxies. Studies of galaxy evolution often include using surveys such as 2-degree Field Galaxy Redshift Survey (2dFGRS) (Colless et al. 2001), the Sloan Digital Sky Survey (SDSS) (York et al. 2000), or the upcoming Large Synoptic Survey Telescope (LSST) (Ivezić et al. 2008). By using surveys, astronomers can study galaxies at various evolutionary phases in order to construct a picture of how galaxies may evolve overall. However, even with very large all-sky surveys, we cannot follow the evolution of a single galaxy to see how particular properties will determine the overall galactic properties. The processes involved in galactic evolution are complex, non-linear, and take

millions of years to unfold. To help understand the complicated relationship between galaxy properties and their evolution, numerical methods are used.

There are two major types of hydrodynamic approaches in astrophysical codes: Lagrangian and Eulerian. Lagrangian codes such as smoothed particle hydrodynamics (SPH) (Gingold & Monaghan 1977; Monaghan 1992) use discretized particles or volumes that have a defined trajectory (Genel et al. 2013), while Eulerian codes are discretized into static volume elements where physical quantities like mass and energy are conserved (Trac & Pen 2003). SPH codes have the benefit of having good conservation properties and being able to have adaptive spatial resolution (Naab & Ostriker 2017). However, they must be modified to include shocks, shear, and some implementations have difficulty modeling fluid mixing (Agertz et al. 2007; Naab & Ostriker 2017). Eulerian codes typically handle shocks better than Lagrangian codes, but they can be prone to numerical diffusion issues (Robertson et al. 2010; Naab & Ostriker 2017).

There has been much work done to attempt to improve numerical simulations to study the evolution of galaxies and an excellent review is provided by Somerville & Davé (2015).

1.4 Simulation code HYDRA

The simulation code Hydra (Couchman et al. 1995) combines a smoothed-particle hydrodynamics (SPH) method (Lucy 1977; Gingold & Monaghan 1977) and adaptive particle-particle, particle-mesh gravity calculations (Couchman 1991). We chose to use Hydra due to our familiarity with the simulation code, as it was used in Fields (2017). The version

used in this work is the serialized version, not the parallelized version seen in Thacker & Couchman (2006). The code HYDRA requires a set of initial conditions of particles, including the positions, velocities, masses, and types of particles specified.

1.4.1 SPH Solver

In smoothed-particle hydrodynamics (SPH) (Lucy 1977; Gingold & Monaghan 1977), each particle has its own smoothing length, which is defined such that the particle has a certain number of neighbors within the radius of the smoothing length. Typically, the number of neighbors is about 50 (Thacker et al. 2000). Within this radius around the particle, the particle-particle interactions for the hydrodynamic forces are calculated and carried by the particles.

The local density at the position of particle i, denoted by $\mathbf{r}_i$, is calculated with a sum over neighbouring particles,

$$\rho_i \equiv \rho(\mathbf{r}_i) = \sum_k m_k W(\mathbf{r}_i - \mathbf{r}_k, h), \tag{1.6}$$

where W is the smoothing kernel and h given here is the smoothing length and determines the size of the smoothing region (Monaghan 1992), with each particle having its own smoothing length according to the local particle density. Each particle k has a mass m_k, position $\mathbf{r}_k$, velocity $\mathbf{v}_k$, temperature T_k, and density ρ_k.

This result can be generalized so that any function or field $A(\mathbf{r})$ can be written as a

summation over N neighboring particles using the smoothing kernel W:

$$A(\mathbf{r}) = \sum_{k}^{N} m_k \frac{A_k}{\rho_k} W(\mathbf{r} - \mathbf{r}_k, h) \tag{1.7}$$

where the value of any A at $\mathbf{r}_k$ is denoted A_k (Monaghan 1992). Gradients of fields can be constructed in a similar manner.

1.4.2 Gravity Solver

The gravity solver is an adaptive particle-particle, particle-mesh (AP^3M) method developed by Couchman (1991). The gravitational force that each particle feels is a combination of a long-range force between the particle and the mesh (PM) and a short-range force between particles (PP). When the distribution becomes more dense due to gravitational forces in the non-adaptive P^3M, the number of particles N in the original grid can become very large. The PP calculation is of order $\mathbf{O}(N^2)$, so as N grows, this calculation slows significantly (Thacker & Couchman 2006).

With AP^3M, this slowdown is avoided by adding sub-meshes, or "refinement" meshes, in high-density regions. By using a refinement mesh on top of the original grid, computational time is drastically reduced as the distribution becomes more dense. Since the AP^3M method allows for refinement meshes on top of refinement meshes, the long-range gravitational force may be the sum of multiple PM calculations. The level of refinement meshes typically are 6-deep at most, as the computational time gain slows down after many additional refinement meshes are added (Thacker & Couchman 2006).

1.4.3 Solution Cycle

The solution cycle of HYDRA is reproduced here from Thacker & Couchman (2006) with modifications appropriate for following contours in this work:

1. In the first iteration, load the initial conditions and choose a contour to follow (see Fields (2017))
2. Assign mass to the Fourier mesh
3. Convolve with Green's functions using the Fast Fourier Transform (FFT) method to get potentials, and difference this to recover mesh forces in each dimension
4. Apply mesh force and accelerate particles
5. Decide where it is more computationally efficient to solve via the further use of Fourier methods as opposed to short-range forces, and, if so, place a new sub-mesh (refinement) there
6. Accumulate the gas forces (and state changes) as well as the short-range gravity for all positions not in sub-meshes
7. Repeat 2-6 on all sub-meshes until forces on all particles in simulation have been accumulated
8. Decide whether any tracer particles need to be added to the contour, and if so, add them
9. Update timestep and repeat.

1.4.4 Running HYDRA

The code Hydra requires a set of initial conditions, including a distribution of particles including the positions, velocities, types, and temperatures of those particles. Additionally, a parameter file is required to identify parameters such as the data file of the initial conditions, a timestep normalization parameter, and boolean values to turn cooling on or off and to evolve either a periodic or isolated box. During the simulation, the default simulation code saves data files at regular intervals for later evaluation. These outputs include the full state of the distribution of particles at the given time, and can be used to restart the simulation. The simulation code can be modified to print additional outputs, such as we implemented with the contour tracing method in Fields (2017), where the length of the contour at each iteration was saved in a continually-updated text file.

More information about system requirements, units used in the code, inputs, outputs, and how to run Hydra can be found in Couchman et al. (1996).

1.5 Context for this work

Following the flow of material is important in understanding the evolution of astronomical processes. In order to learn how turbulent environments can influence the interstellar medium or the intergalactic medium, following the flow of particles that make up those regions can tell us about how these environments are affected (Federrath et al. 2008; Mitchell et al. 2009; Genel et al. 2013). Further, tracer methods can be used to study how gas accretes onto dark matter halos from the intergalactic medium (Genel et al. 2013; Nelson et al.

2013). Tracers can be either scalar fields that follow the properties throughout an evolution, more common in grid-based codes, or they can be individual particles that follow the flow of some property or population. Normally in Lagrangian methods, no extra "tracer" method is needed to learn about the evolution of a system, because the discrete particles in Lagrangian simulations can be traced back in time to learn how they evolved. The reason that tracers are necessary in our work is that we want to follow how much of the gas mixes with itself by measuring the stretching between particles. By following a piecewise continuous "contour" of gas particles, we can measure the change in length of that contour over time. This contour connects the initially-chosen gas particles from a disk galaxy, and typically contains about two dozen gas particles. As the contour stretches, however, resolution is lost as the particles move away from each other. When adding tracers to the contour, we can keep a high-resolution to more accurately follow the flow of the gas particles and measure how much chaotic mixing is in the disk. After simulating a disk galaxy for millions of years, there can be from approximately 10^2 to 10^5 tracers added, depending on the amount of mixing in the disk (Fields 2017).

The method of measuring the amount of chaos utilized in this thesis comes from the work of atmospheric sciences, specifically from the work of Waugh & Plumb (1994) who employed a method called contour advection with surgery (CAS). Advection is the movement of matter and describes the movement of their contour over time. This method was based originally on a technique by Schoeberl & Bacmeister (1993) that added tracer particles continuously to a flow. In Waugh & Plumb (1994), they examined the small scale structures in material with a specific flow. In particular, they were interested in better understanding the stratospheric

polar vortex and the associated large-scale flow in order to detect the transport of material across the vortex edge in meteorological analyses. Here, we examine the change in the length of a contour to determine the amount of mixing happening on small scales. Previous to Fields (2017) and this work, there have not been uses of this contour tracking method in astrophysics to our knowledge. There have, however, been extensive studies of tracer fields in the astrophysics literature (e.g. Federrath et al. (2008), Price & Federrath (2010), Dubey et al. (2012), Genel et al. (2013)).

Previously, Genel et al. (2013) compared types of tracer particles in astrophysical fluid simulations using the AREPO code. Using AREPO, special methods were tested for following the flow in a Lagrangian manner in addition to its Eulerian, grid-based nature. They presented a method of tracking the flow using velocity field tracers, and another method of Monte Carlo tracers. The Monte Carlo tracers are attached to particular volume elements on the grid, and tracer particles are transferred between neighbouring cells according to the mass flux. Their results showed that velocity field tracers, those tracers that followed the flow of material from one cell to another based on velocity fields, were a poorer fit to the overall movement of the flow than the Monte Carlo tracers. However, in Lagrangian-based simulations designed to follow contours, Monte Carlo tracers are not appropriate as the calculation must be exact rather than statistical. We thus use tracers that are similar to the velocity field tracers that Genel et al. (2013) worked with in the Eulerian code AREPO. Despite the velocity field tracers being a poorer fit in Genel et al. (2013), it is important to emphasize that SPH is a different hydrodynamical technique. The SPH code Hydra involves particles instead of grids and thus should be more apt to use the velocity field tracer

method.

Additionally, recent work by Keller et al. (2019) and Genel et al. (2019) has highlighted the chaotic behaviour of galaxy formation and evolution simulations. Their work has shown that the conclusions made about galactic disks may be dependent on the exact models, simulation code, and feedback methods chosen in the work. This variance highlights the need for understanding chaotic evolution in astrophysical simulations.

1.6 Motivating issues

In Fields (2017), the contour surgery method was implemented and preliminary results showed that a power-law relationship between the stability of a disk measured by Q and the amount of chaotic mixing in the disk over evolutionary time measured by λ. That work was later discovered to have issues in the way that the simulations were run and in the tracer method implemented, and we had assumed an isothermal evolution of the tracer population. What was not anticipated while running the simulations in Fields (2017) was that the timestep normalization may affect the results of the simulation, so some simulations were restarted part of the way through (after a server had been restarted, for instance) with a different timestep normalization value in the parameter file.

In this thesis, we determine what affect the changed timestep normalization and other properties have on the tracer method, as well as explaining the precise tracer methodology, and the determination of how the contour length changes over time. The work of Fields (2017) was an excellent proof-of-concept, and it contained a successfully-implemented tracer

contour surgery method; however, it was not fully tested for durability and accuracy due to time and resource constraints. In this work, we dive into some of the problems with the Fields (2017) method and the affects of other properties of the simulation code HYDRA.

This thesis is organized as follows: in Chapter 2, the results of using an adiabatic spherical collapse model are presented, Chapter 3 discusses the tracer method and some challenges with using tracers in addition to showing how differing simulation properties can affect the tracer behaviour. Chapter 4 looks at individual tracers and more systemic difficulties, Chapter 5 gives the Lyapunov results when looking at particular models, and a discussion and summary is presented in Chapter 6.

Chapter 2

Improved contour method and alternate smoothing approach: spherical collapse test

In this chapter, we focus on the evolution of a controlled and moderately well-resolved spherical collapse of gas particles with a $\rho \propto 1/r$ density profile, where r denotes the radial distance from the centre and ρ the density. This approach allows us to test the behaviour of the tracer particles in a solution where the variation in physical quantities, such as density and pressure, have well-resolved gradients.

In SPH, properties of the fluid such as density are calculated at the particle positions by using a smoothing kernel that weights contributions at a given point from neighboring

particles. The equation of motion that governs this evolution for a particle i is

$$\frac{d\vec{v_i}}{dt} = -\sum_{j=1, r_{ij}<2h_{ij}}^{N} m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2}\right) \nabla_i [W(\vec{r_i} - \vec{r_j}, h_i) + W(\vec{r_i} - \vec{r_j}, h_j)]/2 \tag{2.1}$$

where $\vec{v}$ is the velocity vector of the particle, $\vec{r}$ is the position vector, m is the mass, P is the pressure, ρ is the density, h is the smoothing length, and W is the smoothing kernel. The smoothing kernel given by W depends on the distance from the particle and the neighbor particle b as well as the smoothing length h. The kernel used in HYDRA is the B2-spline, given as follows:

$$W(x) = \frac{1}{4\pi} \begin{cases} 4 - 6x^2 + 3x^2, & 0 \leq x < 1; \\ (2-x)^3, & 1 \leq x \leq 2; \\ 0, & x > 2 \end{cases} \tag{2.2}$$

where

$$x = \frac{|\vec{r_1} - \vec{r_2}|}{h} < 2$$

and is described in further detail in Thomas & Couchman (1992). Following this kernel-smoothing methodology, subroutines to smooth either the velocities or the accelerations were developed and implemented in the contour tracing code discussed in Fields (2017). The tracers were not entirely removed from the long-range gravitational calculations, though the goal was to have the tracers follow the flow and not affect the flow themselves. In this chapter, we study the effects of changing simulation and tracer properties on the accuracy of the tracer flow compared to the flow of the gas particles in the spherical collapse test.

2.1 Velocity smoothing for tracer particle insertion

When adding tracers to the contour via the surgery method described in Fields (2017), the tracers needed to have initial properties such as mass, position, and velocities assigned to them. Two methods here were used and tested to set the initial velocities. The first method was a pairwise average and the second was a smoothed weighted average calculation which we implemented in a subroutine `velsmooth`. For example, the pairwise velocity was a straightforward calculation of the average of the 3-dimensional velocities of the particles in the contour on either side of the new tracer,

$$v^i_{\text{tracer}} = \frac{v^i_{\text{left}} + v^i_{\text{right}}}{2}, \tag{2.3}$$

where v^i_{left} and v^i_{right} would denote the components of velocities of the particles (either gas particles or tracer particles) on either side of the new tracer being placed in the contour, and i goes from 1 to 3 for the three components of the velocity vector, v_x, v_y, and v_z.

The second method was an algorithm that used a smoothing kernel weighting of the neighbors of the newly-created tracer particle to interpolate the velocity at the position of the tracer and thus assign that smoothed velocity. Following Monaghan (1992),

$$v^i(\vec{r}) = \sum_{b=1}^{n_{\text{part}}} m_b \frac{v^i_b}{\rho_b} W(\vec{r} - \vec{r}_b, h), \tag{2.4}$$

where $v^i(\vec{r})$ corresponds to either the x-, y-, or z-component of velocity vector $\vec{v}$ at the position $\vec{r}$ of the particle whose property is being calculated. The summation is over all

other particles, denoted b with mass m_b, density ρ_b, position $\vec{r}_b$, and velocity v_b^i in dimension i and this sum includes the smoothing kernel W.

To emphasize, the velocity of the tracers is only affected by either the pairwise calculation or by `velsmooth` during the iteration that they are added to the contour or in other words, their initial velocity. In later iterations, the velocity of the tracer particles are calculated by using the acceleration.

2.2 Acceleration smoothing for the tracer position integration

In addition to testing how velocities were initially set, we also implemented an alternative way of calculating the tracer acceleration in a subroutine called `asmooth`. This algorithm replaced the gravity and hydrodynamic calculations for the tracer particles only, so they did not go through the usual pairwise accumulation of these force evaluations. The smoothed acceleration was calculated via the same equation as 2.4 except modified for $a^i(\vec{r})$ components, thus

$$a^i(\vec{r}) = \sum_{b=1}^{n_{\text{part}}} m_b \frac{a_b^i}{\rho_b} W(\vec{r} - \vec{r_b}, h). \tag{2.5}$$

The above equation can be compared to the pairwise approach found in eq. (2.1) where the properties of neighboring particles are used to determine the acceleration of the particle. It is worth emphasizing that smoothing the acceleration is a continual process done at every step while the velocity smoothing is just used to set the initial velocity of the tracer. Once set, the velocity is updated every step from the acceleration. In this method, since the

tracers are not part of the pairwise calculation, we would not call them "active" tracers, but rather "passive" tracers. The difference between the "active" and "passive" terminology is whether the pairwise sum of components in the hydrodynamic and gravitational forces is active or not. In either case, the tracers are not seen by the other particles in the simulation during the hydrodynamic and gravitational calculations.

Initially, we ensured that the accelerations calculated for new tracers did indeed match the accelerations of surrounding particles by placing a new tracer into a grid of uniformly-moving particles and correctly getting the same acceleration for the new tracer particle. However, it is necessary to see how the method performs over the course of a simulation and different hydrodynamic states.

In this chapter, the two methods were compared for effectiveness and to test how well the tracer behaviour follows the neighboring fluid.

2.3 Test Case: Spherical Adiabatic Collapse

To evaluate consistency of the tracers with the surrounding fluid, we used an adiabatic collapse of an initially isothermal spherical gas cloud which follows the work of Evrard (1988) and Thacker et al. (2000); and the properties of this test are thoroughly discussed in Thacker et al. (2000). This test contains only gas particles, no collisionless star or dark matter particles, and these gas particles follow an adiabatic equation of state. This model has a $\rho \propto 1/r$ density profile produced by scaling a uniform grid. During the evolution, the particles collapse inward with no rotational velocity. In this model for the parameters

we have chosen, individual gas particles have a mass of approximately 58 million $M_\odot$, with approximately 17,000 gas particles, giving a total mass of $10^{12}M_\odot$, broadly consistent with representing a large galaxy or small group of galaxies.

The initial cloud collapses and an accretion shock front propagates outward, while a thermalization causes a slight bounce in the evolution which eventually becomes virialized. To ensure our spherical collapse test reproduced the evolution found in Thacker et al. (2000), Fig. 5 from their paper is recreated using our data and can be seen here in Fig. 2.1. In this figure, we show the kinetic, thermal, potential, and total energy over time for both a version of the spherical collapse run with and without tracer particles. Therefore, our spherical collapse test both produces expected results, and adding tracers to the simulation does not significantly change either the total energy or the evolution of the components of the total energy in the simulation in this model, as we expected.

In our tests, a sphere of tracer particles is inserted into the initial adiabatic sphere of gas particles to measure how well the tracers follow the flow and the shock front over the evolutionary time of the collapse. A sample distribution of the tracers and gas particles can be seen in Fig. 2.2. Here, a single z-plane is shown to best illustrate how the tracers are inserted between all neighboring gas particles.

Fig. 2.3 shows the typical positions of tracer and gas particles during evolution. This particular example is from Run 1013, as seen in Table 2.1, §2.3.1. The initial sphere gas particles are cut from a uniform grid of particles where the number of particles per side is $L = 32$. In this model, the initial sphere of gas particles is duplicated and offset as in Fig. 2.2 and set to be tracer particles with a mass of $0.001m_{\rm gas}$ where $m_{\rm gas}$ is the mass of

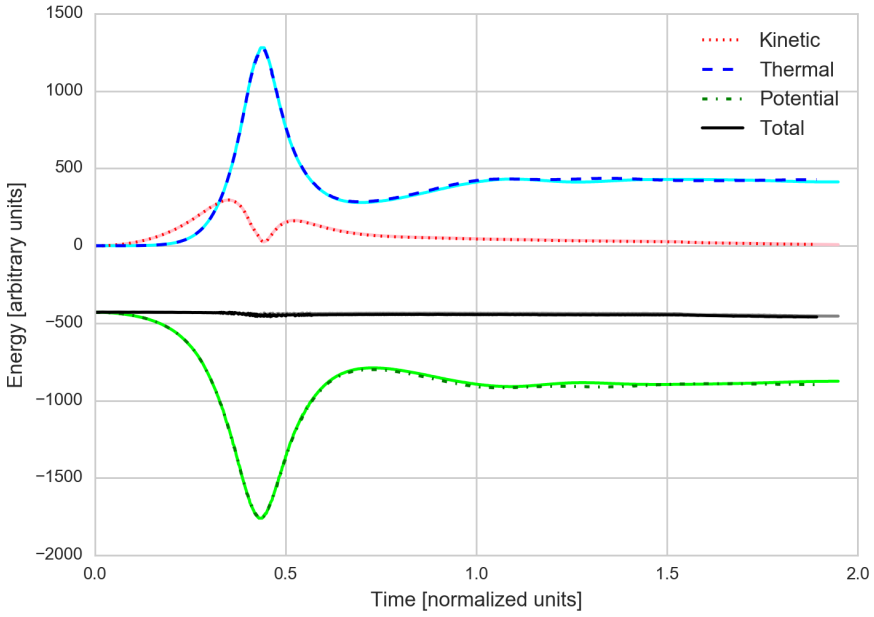

Figure 2.1: The total (black), thermal (blue), kinetic (red), and potential (green) energies are shown for the spherical collapse simulation over time where both energy and time are given in normalized units. The solid, lighter lines correspond to a run without tracer particles, and the lines shown in the legend (with varying line styles) are from a run with tracer particles. It is clear these two sets are in good agreement, and this figure reproduces what is seen in Fig. 5 of Thacker et al. (2000).

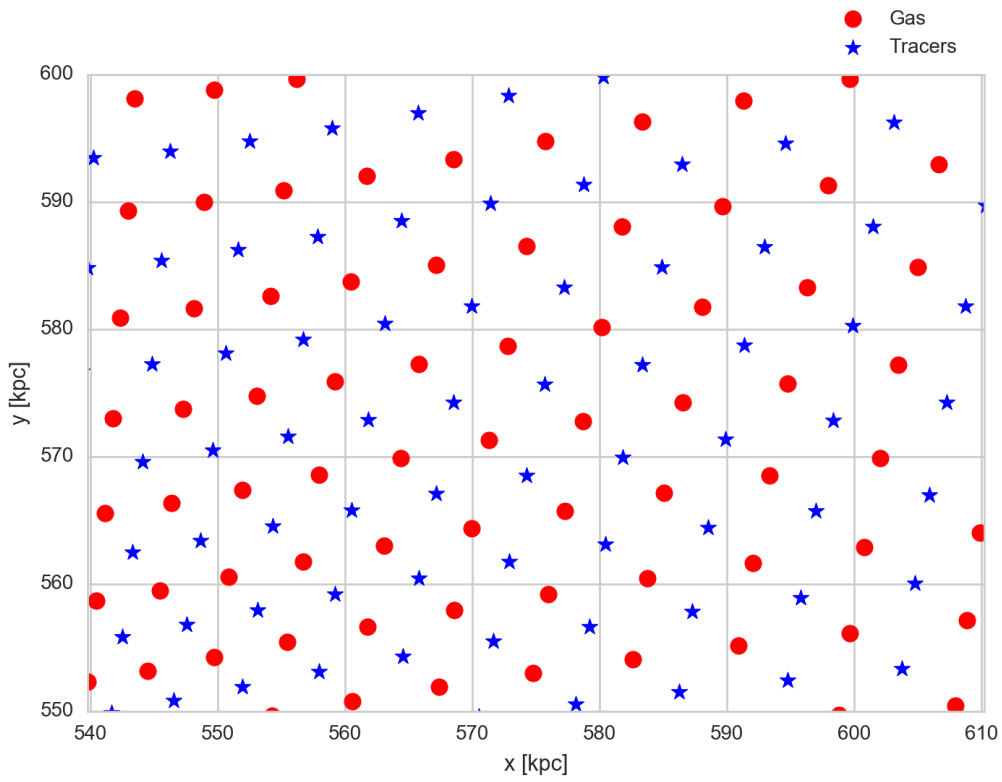

Figure 2.2: Illustration of the tracers located between gas particles in the spherical collapse models. Here, a single z plane shows the x and y positions of gas (red circles) and tracer (blue star) particles.

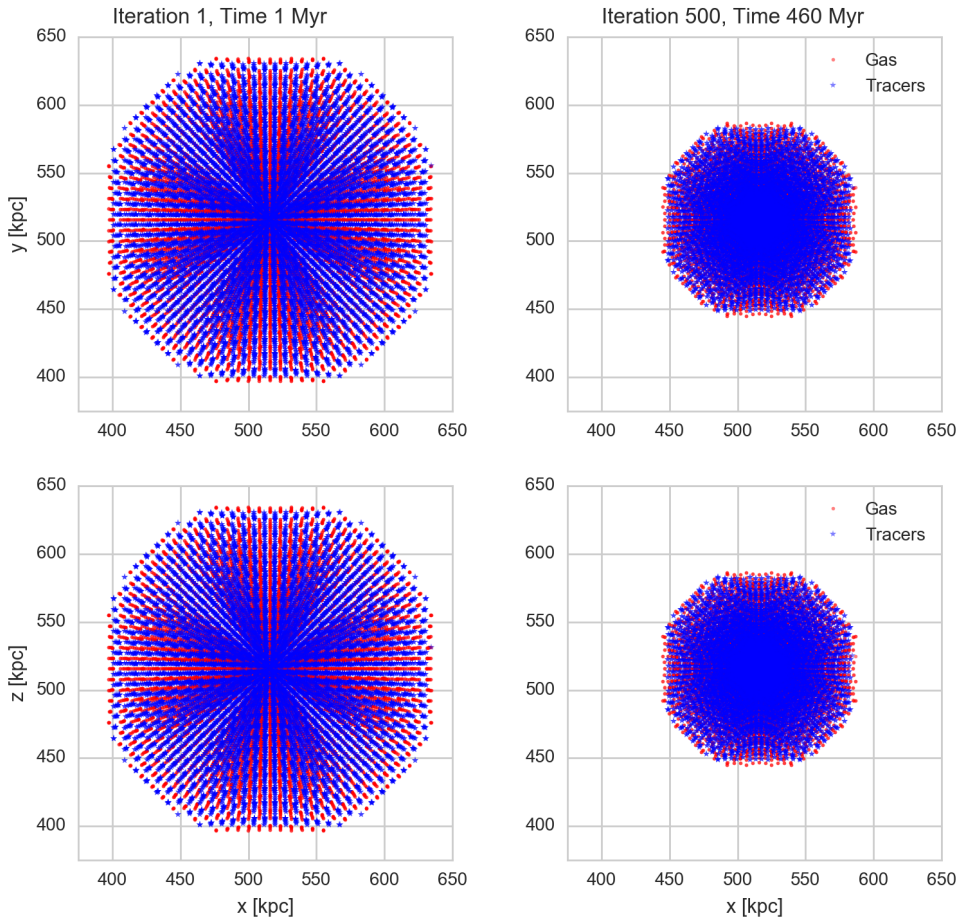

Figure 2.3: This figure shows the evolution of the spherical collapse model. The positions of both the gas (red dots) and tracer particles (blue stars) after the first (left panels) and 500^{th} (right panels) iterations. The two top panels show particle positions in the x-y plane and the bottom two panels show particle positions in the x-z plane. This particular figure shows the data from Run 1013. This spherical collapse test had passive tracer particles with $1/1000^{\text{th}}$ the mass of a single gas particle inserted between every pair of gas particles, and included the acceleration smoothing algorithm 'asmooth'.

single gas particle in the distribution. Often, tracer particles in simulations are massless, but we use a small, non-zero mass to prevent numerical divergence or divide-by-zero errors. Thus, there are two overlapping spheres of $n_{\rm part} = 34015$ particles that evolve together – a sphere of gas particles and a sphere of tracer particles. The goal is to see how well the two co-located spheres mimic each other to test whether the tracer particles adequately follow the flow of the gas particles over time.

In the left panels of Fig. 2.3, the x-y (top panels) and x-z (bottom panels) planes are shown for the initial sphere of combined gas and tracer particles. The right panels of Fig. 2.3 show the x-y and x-z planes at 460 Myr into the evolution of the spherical collapse, after the shock front has moved through the sphere from the outer edge to the center and the particles have started to bounce back. As can be seen, the overall structure of the sphere seems to be visually well-matched between both the gas and the tracer particles, which is indicative of the tracer particles adequately following the flow of the gas particles.

2.3.1 Effect of velocity and acceleration smoothing on the spherical collapse model

These tests include the smoothing algorithms for velocity and acceleration described previously, `velsmooth` and `asmooth`, we test these types of smoothing in a controlled and moderate-resolution model to examine if the tracer population reproduces the density field of the gas particles. Table 2.1 shows the runs used to compare the effectiveness of the two smoothing algorithms with tracers along with a "control" test that includes tracers but no additional smoothing algorithm for either velocity or acceleration, therefore these are active

Table 2.1: Testing the effect of the two smoothing algorithms, 'velsmooth' and 'asmooth' against a run with neither. These runs have a full sphere of tracer particles with $n_{\rm part} = 34015$, allow for variable tracer temperatures, have a minimum temperature of $e_{\rm min} = 1000$ K, and a tracer mass of $0.001 \times m_{\rm gas}$. The 'Velocity Calculation' and 'Acceleration Calculation' columns give how the properties were calculated.

Run ID	Velocity Calculation	Acceleration Calculation	Final Iteration	Time [Myr]
1010	`velsmooth`	Pairwise	301	460
1013	Pairwise	`asmooth`	498	460
1014	Pairwise	Pairwise	301	460

tracers whose initial velocities are calculated using the pairwise velocity calculation shown in eq. (2.3).

In Table 2.1, the Run ID is the 4-digit code used to refer to unique simulations. These tests allowed the tracer temperatures to vary as the simulation evolved, rather than fixing the tracer temperatures at some minimum temperature, as was seen in the work of Fields (2017). The temperature of particles in Hydra is related to the internal energy, and the minimum temperature in these simulations is set to $e_{\rm min} = 1000$ K. We use these runs to compare and contrast how the different smoothing algorithms impact the tracer evolution.

In Run 1010 and Run 1014 where the acceleration calculation is done pairwise, the tracers in our simulation are considered "active" because the forces and accelerations are evaluated in the gravity and SPH calculation. Run 1013 uses `asmooth` to calculate the acceleration of the particles at each timestep and thus the tracer forces are calculated outside of the gravity and SPH loop, so the tracers in this simulation are considered "passive."

The radial positions over time of the tracer and the original gas particles in Run 1010

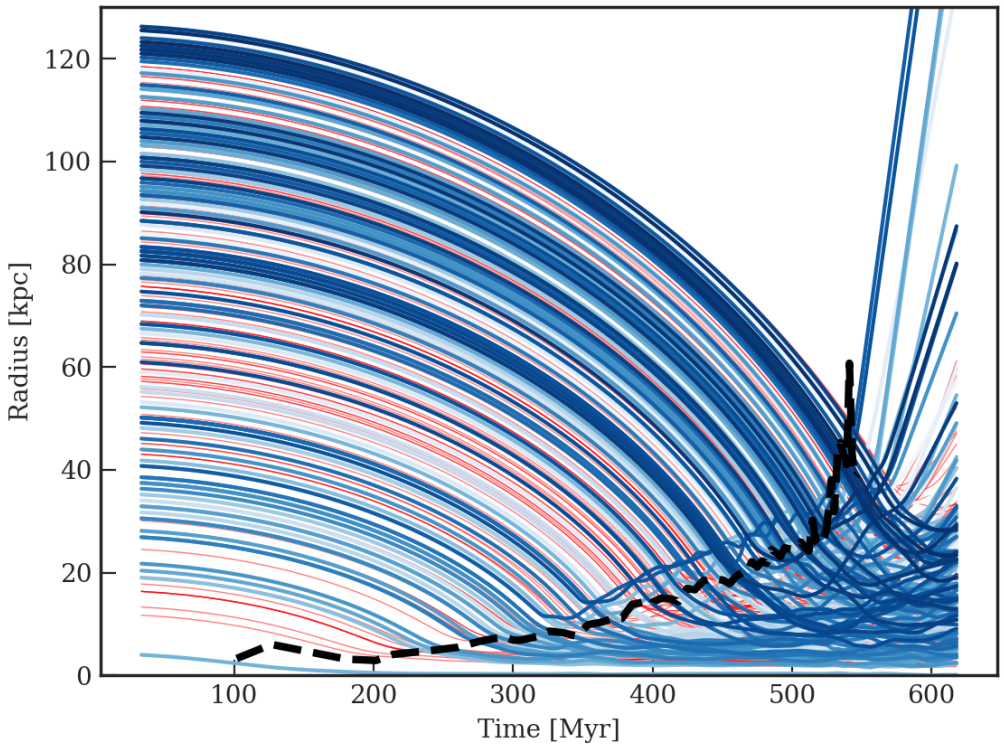

Figure 2.4: The radial positions of the tracer (blue lines) and gas particles (red lines) and the shock positions (black dashed line) over time for Run 1010.

were compared in Fig. 2.4 to see how well the tracers followed the flow of the gas particles in this test. By following the radial positions over time, we could also see the position of the shock front change in time, to compare how well the tracers moved with the gas particles both interior to and exterior to the front. This figure shows the radial positions over time for the gas and tracer particles for Run 1010 from Table 2.1. The red lines in the figure are the positions over time for individual gas particles, while the varying blue-colored lines show the position over time for individual tracer particles. The black dashed line shows the

position of the shock over time, which is calculated from the steepest slope at each time step of the radial velocity profile. The radial velocity is calculated in the following way

$$v_r = \frac{|\vec{v} \cdot \vec{r}|}{|\vec{r}|} \tag{2.6}$$

where $\vec{r}$ is the position vector and $\vec{v}$ is the velocity vector. Each of the x-, y-, and z-components of position and velocity are saved for all tracked particles at every time step in the simulation. It can be seen that the shock front first moves outward ($\sim$ 100-200 Myr) before moving very slightly inward and then propagating outward again.

The number of tracers and gas particles with radial positions less than the position of the shock front are plotted over time in Fig. 2.5 for Run 1010. The blue stars represent the number of tracer particles at a given time, and the red dots represent the number of gas particles inside the shock front over time. The agreement between these two distributions is excellent and thus shows well that the tracers follow the flow of the gas particles in Run 1010. This shows that in this controlled spherical collapse regime with velocity smoothing and with pairwise acceleration calculations, the tracers follow the gas particles well.

In Fig. 2.6, the density and radial velocity profiles for each of the three runs in Table 2.1 are shown in panels (a), (b), and (c) for Run 1010, 1013, and 1014 respectively. These figures are shown for each run at the same evolutionary time, 460 Myr, as given in Table 2.1. The profiles for the density and radial velocity distributions are created by binning particles radially from the center to the edge of the distribution into 30 evenly-populated bins. The blue stars are representative of the bins of tracer particles at each radius, and the red circles

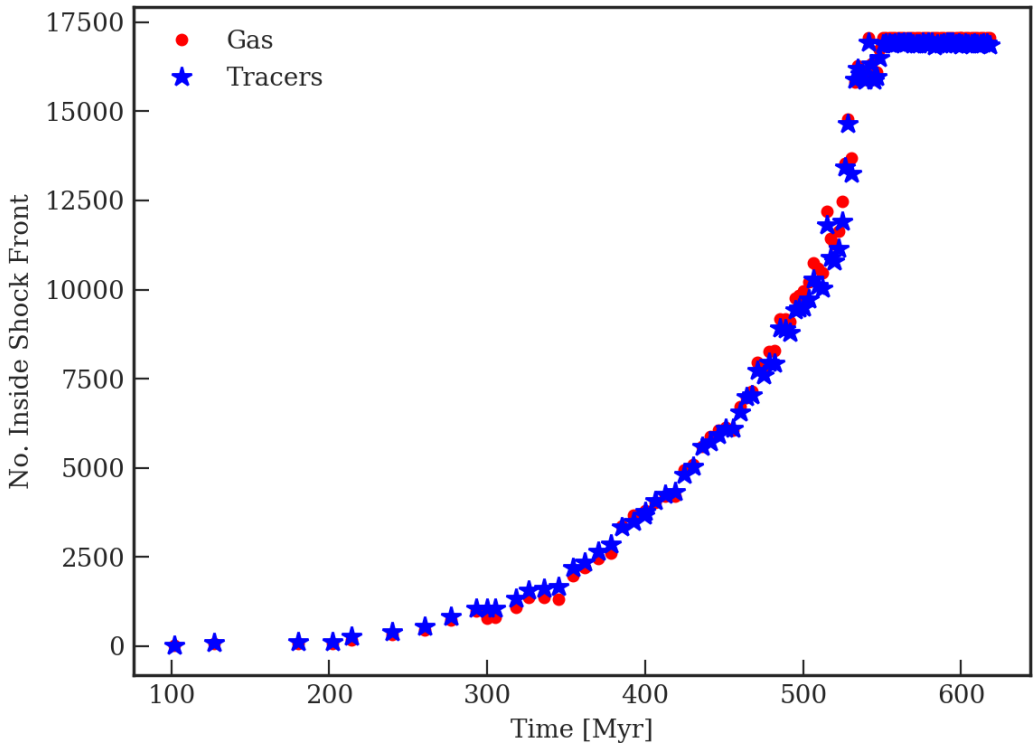

Figure 2.5: The number of tracers inside the shock front for Run 1010 over time. This figure shows the number of tracer (blue stars) or gas particles (red dots) with radial positions less than the shock front position at each time step.

represent the bins of gas particles. The gray shaded region shows where the resolution is limited, and is equal to five times the softening length, where the softening length is equal to 2 kpc for all runs.

The density is plotted by counting the number of particles in each bin (a constant) and then dividing by the volume of the radial shell, using the following formula

$$V_{\text{shell}} = \frac{4\pi}{3}(r^3_{\text{outer}} - r^3_{\text{inner}}) \tag{2.7}$$

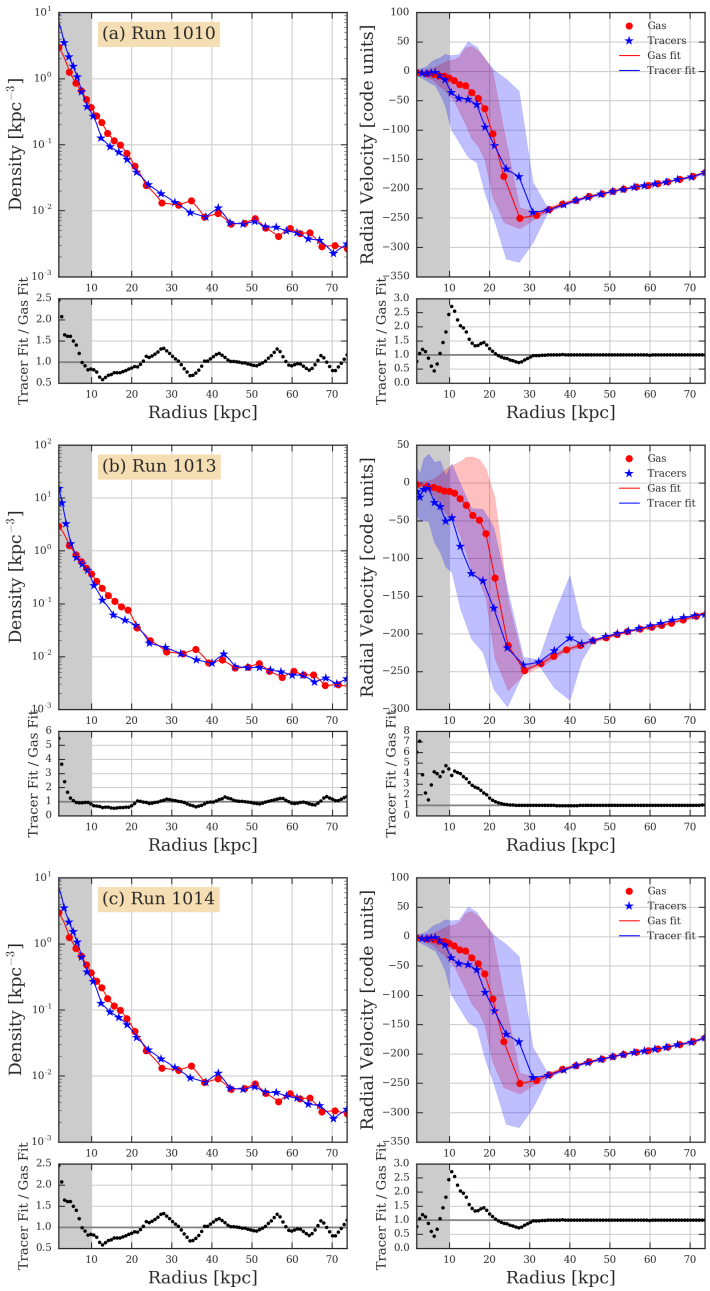

Figure 2.6: The density and radial velocity profiles for Runs 1010, 1013, and 1014 are shown above for both the gas and tracer particles in panels (a), (b), and (c) respectively. The ratio between the gas and the tracer linear interpolation fits are shown in the residuals at the bottom of each panel. The red circles represent gas particles, while the blue stars represent tracers, and the red and blue shaded regions represent one standard deviation above and below the mean value for each population. The gray shaded region is the region below which the resolution is limited.

where r_{outer} is the outer radius of the bin and r_{inner} is the inner radius of the bin. Since the masses of the gas particles and tracer particles are different, the densities shown in the figure are normalized by the mass of the given particle so that the two distributions can be appropriately compared. The radial velocity v_r is calculated using eq. (2.6) for all of the particles in a given bin, then the average of the distribution is plotted with the point. The blue and red shaded regions along the points show one standard deviation above and below the mean value.

For both the density and v_r distributions, a linearly-interpolated fit is given. This fit is a simple linear interpolation between every point for both the gas and tracer populations, and is used to create the residuals plots. At the bottom of each panel for both the density ρ and radial velocity v_r distributions are the residuals between the gas and tracer populations. The residuals are calculated by dividing the fit of the tracer population by the fit of the gas particle population, and do not account for bin variance. A solid gray line in shown in the residuals plots to show what the residuals should be if the two fits were exactly the same. As a note, in this figure and the figures following this one, the distributions are shown at the latest time of all simulations of a given set. In this section, the simulations were limited to 460 Myr (Table 2.1) due to simulation data limitations, and future comparisons are limited to the time given in each upcoming tables in each section.

It is easiest to determine the similarities and differences between the distributions of gas and tracer particles by looking at the residuals in each panel of both density and radial velocity. Panel (c) for Run 1014 of Fig. 2.6 could be considered the "control" run in this set, since the tracers are active (no `asmooth` use) and the velocities of the tracers are determined

via a pairwise calculation rather than with `velsmooth`. While the residuals in panel (c) are within the range 0.5 to 3.0 for both the density and v_r profiles, comparison of the bin variances for velocities shows that the means are within acceptable limits. The larger variances are associated with post-shock regions, those that are yet to feel the shock and are very tightly correlated. Importantly, the residuals are largest at the edge of the central resolution limit which is to be expected. When examining the residuals from Run 1010, panel (a) of Fig. 2.6, the residuals are just about the same as seen in Run 1014, panel (c). Therefore, the addition of `velsmooth` used in Run 1010 produces almost identical results to the control run for the spherical collapse test shown here. Using this information, it can be concluded that including `velsmooth` in future simulations is reasonable, and does not significantly impact how the tracers follow the flow of the gas particles.

In Run 1013, panel (b) which use passive tracers and the acceleration smoothing method, however, it is evident that the radial velocity distributions of the gas and tracer particles are less well-matched, especially in a few bins where the average gas particle v_r is substantially higher than the tracer particle average v_r at the same radial position. For Run 1013, the radial velocity of the tracers does not match the radial velocity of the gas particles around the shock front ($\sim$ 10–20 kpc). We thus conclude that the acceleration smoothing algorithm is less accurate in terms of following the flow of gas particles around shock fronts as compared to active tracers.

To summarize, the addition of the velocity smoothing algorithm `velsmooth` has comparatively little effect on the overall accuracy of the tracers in following the flows. However, the acceleration smoothing algorithm `asmooth` does not shock particles sufficiently strongly

and appears inferior compared to the pairwise calculation, directly calculating the local acceleration.

2.3.2 Effect of tracer mass on the spherical collapse model for active tracers

Since the tracers are not seen by other hydrodynamic particles in theory their mass should have little impact on the overall evolution. However, we did not entirely remove them from the long-range gravitational calculation, so we test this variation here. Table 2.2 shows the simulations used to determine the effect of tracer mass in the spherical collapse model. All of the runs used in this section are the same except for the tracer mass. They all have a temperature floor of $e_{\rm min} = 1000$ K, allow for a variable tracer temperature, and contain a full sphere of tracer particles in the initial conditions that mimic the distribution of the normal gas particles, as seen in Fig. 2.2. Also included is the velocity smoothing subroutine `velsmooth`. The sphere has 34171 total particles with $L = 32$. The six simulations are

Table 2.2: Runs used to determined if tracer mass has an effect on the results of the spherical collapse model simulations. All runs in this table are the same except for the tracer mass. They all have $n_{\rm part} = 34171$, include 'velsmooth', and allow a variable tracer temperature.

Run ID	Tracer Mass	Final Iteration	Time [Myr]
4000	$1{\times}10^{-6}$ $m_{\rm gas}$	298	450
4001	0.001 $m_{\rm gas}$	296	450
4010	0.01 $m_{\rm gas}$	307	450
4050	0.05 $m_{\rm gas}$	309	450
4100	0.1 $m_{\rm gas}$	295	450
4950	0.95 $m_{\rm gas}$	553	450

all compared at the same evolutionary time, 450 Myr. The differences in the tracer mass, compared to the mass of a gas particle, are shown in Table 2.2. The tracer masses range from one millionth the mass of a gas particle in Run 4000 to 95% the mass of a single gas particle in Run 4950.

In Fig. 2.7, the binned density and radial velocity versus radius are given for all runs in Table 2.2 in panels (a) through (f). These plots are created the same way as Fig. 2.6 in the previous section. In the figure, good agreement is seen between the densities and v_r of the tracer and gas particles at all radii, except for in panel (f) with Run 4950, where the mass of the tracer particles is 95% that of a gas particle. Panel (f) shows Run 4950 and it is clear when examining both the density and v_r distributions that this run is at a different stage of evolution than the others, even though it is plotted at the same evolutionary time, 450 Myr. The high mass of the tracer particles in this run significantly impacts the overall mass of the sphere, whereas the mass is only increased by at most $\sim$10% in the other five mass tests. As would be anticipated, the plot shows it is unreasonable for tracer particles to be so massive, as the tracers should have negligible impact on the overall evolution of the system.

Throughout panels (a) through (e), the residuals are fairly similar across all radii and the shaded regions representing one standard deviation in the v_r profiles for the gas and tracer populations are similar throughout the runs. This shows that the tracer mass (ranging from one millionth the mass of a gas particle in Run 4000, panel (a), to 10% of the mass of a gas particle in Run 4100, panel (e)) does not significantly change the flow of the tracer particles compared to the gas particles for this spherical collapse model. This is not unexpected given

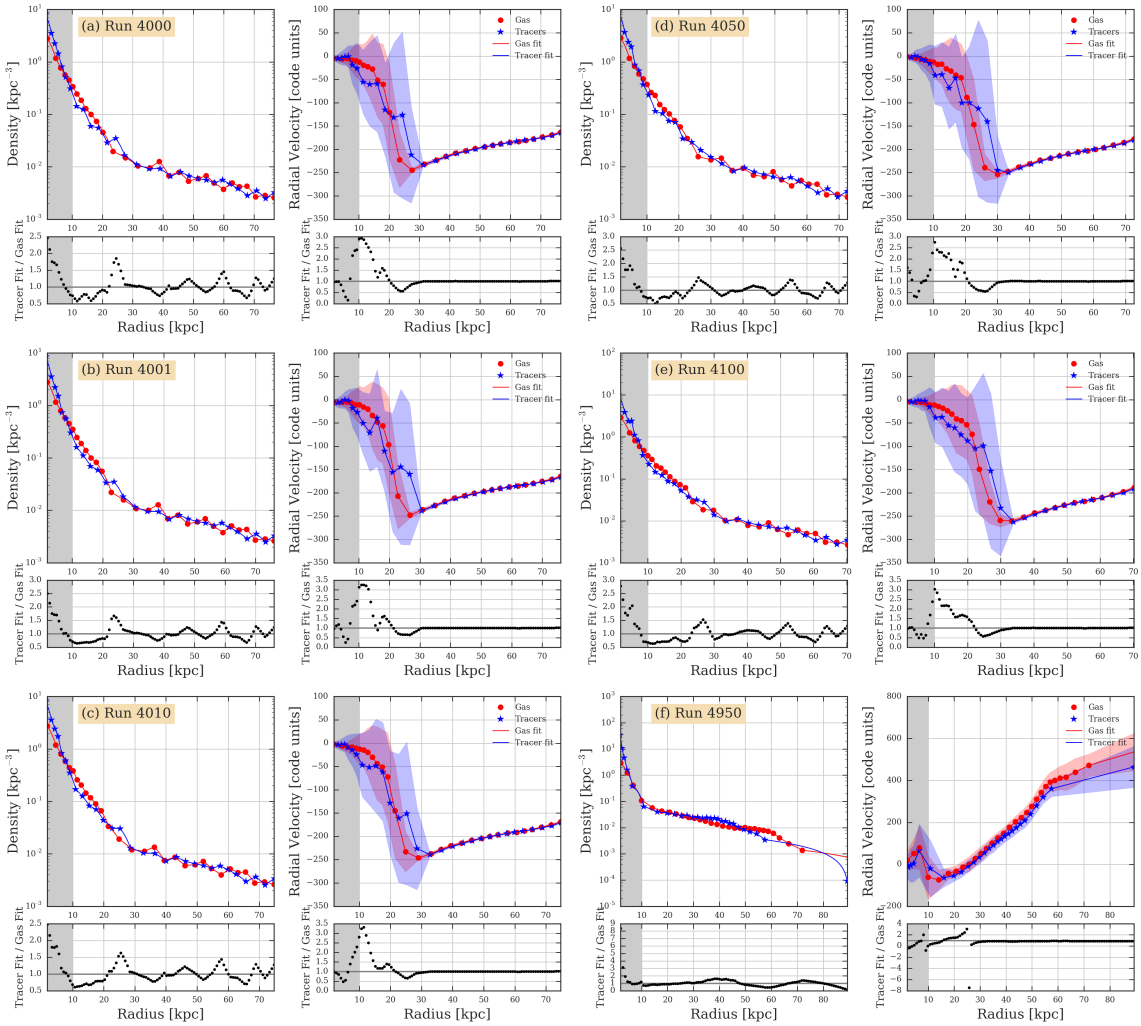

Figure 2.7: The density and radial velocity profiles for Runs 4000, 4001, 4010, 4050, 4100, and 4950 are shown above for both the gas and tracer particles in panels (a), (b), (c), (d), (e), and (f) respectively along with a fit to the points (solid lines). The ratio between the gas and the tracer linear interpolation fits are shown in the residuals at the bottom of each panel. The red circles represent gas particles, while the blue stars represent tracers, and the red and blue shaded regions represent one standard deviation above and below the mean value for each population. The gray shaded region is the region below which the resolution is limited.

bin variances are usually larger than 10% levels.

2.3.3 Effect of timestep normalization of tracers on the spherical collapse model

Timestep normalization in the simulation determines how many iterations it takes to do a full timestep. This is a prefactor ahead of the Courant-Friedrichs-Lewy condition (Courant et al. 1967) in the code, which is the timestep and stability criterion used in HYDRA. The variable used to set the timestep normalization in the parameter file is `dtnorm`. A shorter timestep is anticipated to produce a more accurate integration at the cost of more computation and potentially more numerical diffusion, a situation where the particles are more spread out than they would be physically.

Table 2.3: Tests used to determined if the length of the timestep of the iterations has an effect on the results of the spherical collapse model simulations. In each of the runs below, the are 34172 particles, about half of which are tracers. In all cases, the tracer mass is 0.001 $m_{\rm gas}$. The only thing changing between the Run IDs is the timestep normalization 'dtnorm'. The table shows the Run ID used to identify each run, the value of 'dtnorm', the number of the final iteration and the final time in Myr used in the comparison.

Run ID	`dtnorm`	Final Iteration	Time [Myr]
4001	1.0	80	298
4015	0.5	146	300
4016	0.25	300	298

Table 2.3 shows the three runs compared in this section and the differing timestep normalization values in the particular run. The mass of the tracers in all of the runs is 0.001 $m_{\rm gas}$, the minimum temperature $e_{\rm min}$ = 1000 K, the tracers are allowed to have variable

temperature, each run contains 34172 particles with $L = 32$, and the distribution of tracer particles in the initial conditions is a sphere similar to the distribution of gas particles, as seen in Fig. 2.2. Run 4001 has a `dtnorm` of 1.0, Run 4015 has `dtnorm` of 0.5, and Run 4016 has `dtnorm` of 0.25. The three simulations are each presented at roughly 300 Myr, somewhat earlier than in the other tests but still comparatively far into the evolution.

Fig. 2.8 shows the distributions of density and radial velocity for the three simulations with varying `dtnorm` from Table 2.3. The figure properties are the same as Figs. 2.6 and 2.7 from the previous sections. In Run 4001 (Fig. 2.8, panel (a)), the shock front is on the outside edge of the unresolved region, as can be seen in the sharp difference in radial velocity at about 10 kpc from the center. The tracer and gas radial velocities agree very well outside the shock radius, however. The density distributions for both the gas and tracer particles are also very similar with low residuals. Fig. 2.8, panel (b) shows the density and radial velocity distributions for Run 4015, which show good agreement between the gas and tracer particle distribution as well. Run 4016 is shown in Fig. 2.8, panel (c).

The three panels all show low residuals and good agreement between the gas and tracer particle distributions, showing that the effect of timestep normalization is very small on the results of this spherical collapse test. This confirms that for adiabatic tests like this, the timstepping is accurate in the default `dtnorm`=1.0 setting.

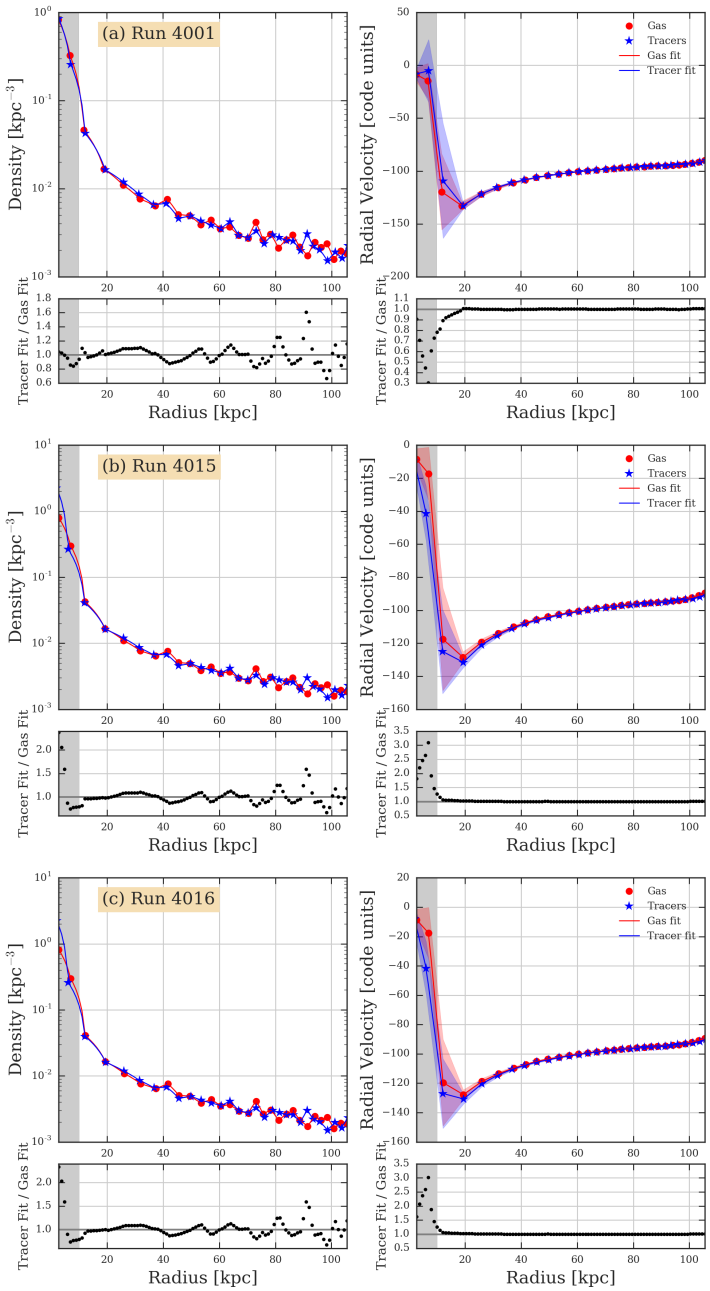

Figure 2.8: The density and radial velocity profiles for Runs 4001, 4015, and 4016 are shown above for both the gas and tracer particles in panels (a), (b), and (c) respectively along with the fits to the data (solid lines). The ratio between the gas and the tracer linear interpolation fits are shown in the residuals at the bottom of each panel. The red circles represent gas particles, while the blue stars represent tracers, and the red and blue shaded regions represent one standard deviation above and below the mean value for each population. The gray shaded region is the region below which the resolution is limited.

2.4 Summary

Overall, there were only minor differences, acceptable within bin variances, between the gas and tracer particle density, position, and radial velocity profiles over time throughout a wide variety of tests. This shows that the spherical collapse test is a well-controlled test that can be followed by a tracer population. We found that smoothing the acceleration field was in general less accurate for the tracer population than evaluating acceleration locally using a pairwise approach, especially when capturing the shocks. Meanwhile, we found that setting the initial velocity via a local smoothing algorithm had comparatively little improvement versus a two-particle average, likely because the collapse of this model was smooth overall and thus nearby particles and those neighboring particles used in the smoothing algorithm had similar velocities.

Chapter 3

Disk simulation results and changes to tracer methods: modified artificial viscosity, temperature, and additional factors

In Ch. 2 we showed that most tracer implementations reproduced a spherical collapse in the tracer population, thus following the flow of the gas particles to good precision. The small differences between the gas and tracer populations in Ch. 2 are comparable to the differences found when comparing different types of simulation codes, such as shown in Frenk et al. (1999). Hence, we continue to test our tracer method on more challenging simulations.

In this chapter, we extend simulation models to follow contours of gas particles in the

evolution of disk galaxies with the goal to eventually compare global properties such as stability to local properties like the amount of chaotic mixing, as in Fields (2017). The goal in earlier work was to calculate the Lyapunov exponent λ of each model by measuring the change in length of a contour throughout its evolution. To do this, contour placement algorithms and a "surgery" method were implemented in the simulation code. The primary goal of this chapter is to revisit the disk simulations first considered in Fields (2017) with a view to overcome the isothermal limitations adopted in that work.

3.1 Tracer injection method in the galactic disk model

In the previous chapter, the model used to test the effectiveness of the tracer particles was an adiabatic spherical collapse which was well-resolved throughout its evolution. Here, we use disk galaxy models extensively studied in Foyle (2007) in a less-well-resolved regime where strong density contrasts develop over time. This model is much more dynamically complicated than the spherical collapse as there are strong density contrasts and disk asymmetries throughout its evolution.

To follow the evolution of gas particles over the duration of the evolution of a disk galaxy, a subset of the total gas particles were followed and their properties were output at each timestep of the simulation. The method of inserting tracer particles into a tracked contour is described in full detail in Fields (2017) and will be briefly described here. The subset of gas particles was chosen in the initial conditions to be a circular contour of particles in the middle of the equatorial z-plane at a radius that is comparable to the radius of the Sun

from the center of the Milky Way, or about 2.5 times the scale length h of the disk (Fields 2017). The scale length gives a characteristic size of the disk, not to be confused with the smoothing length found in eq. (2.1). An example of the overall distribution of gas particles in the disk and the initially-chosen contour of gas particles can be seen in Fig. 3.1. This figure shows the distribution of only the gas particles (red) along with the initially-chosen contour of gas particles (black line) after the first iteration of a simulation of Model 63, a model which was described in greater detail in Foyle (2007) and Fields (2017). Model 63 had a total of 148,001 particles in its simulation, with 48,036 dark particles in a halo, 59,965 star particles in the disk and bulge, and 40,000 gas particles in the disk of the galaxy, though neither the dark or star particles are shown in the figure for clarity. It is a very unstable model, with $Q = 0.751$, and its low stability parameter made it an excellent test for our tracer methods. This figure does not show the full extent of the initial conditions, particularly the halo, but is zoomed-in to visualize the location of the contour relative to the disk.

As the disk evolved in time, the initial contour of particles stretched and sheared and lost resolution in total length. To avoid being limited when measuring the change in length of the contour, a "surgery" method was implemented where tracer particles were injected into the disk along the contour whenever two adjacent particles became too spread in distance. The goal for these tracer particles was to only follow the flow of the gas particles in the disk, therefore enabling us to accurately infer the total length of the contour over time, even as it gets stretched by the disk evolution. The tracers needed to have as little of an effect on the physical calculations of the rest of the particles as possible.

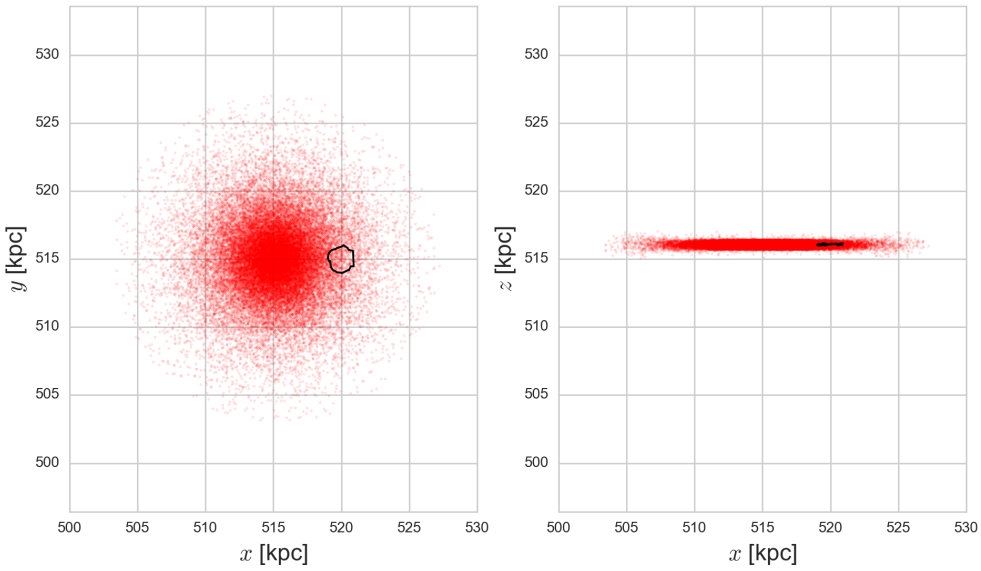

Figure 3.1: The initial contour for a simulation with the Model 63 initial conditions is shown here. The red dots are gas particles, and the black line is the contour of gas particles that is chosen in the first iteration of the simulation. For clarity, the dark and star particles of this model are not shown.

These tracers and the algorithm to inject them into the disk and follow them through the contour was developed and implemented in Fields (2017), but due to time constraints was not fully tested for accuracy and effectiveness. Concerns about poor capturing of shocks were overcome by using an isothermal equation of state. This thesis aims to further test the tracer particles, particularly improving their behaviour in high density contrasts and by understanding how their properties determine the overall contour length.

Ensuring that the tracers follow the flow accurately was the main goal of this work, as any single error in the tracer population significantly affects the measurement of the length of the contour as a whole, thus the Lyapunov exponent (λ) and the amount of chaotic mixing could not be calculated. This is a particularly strong constraint on evolution.

3.2 Initial conditions

The disk models used here are from the initial conditions created by Foyle (2007) and are described in further detail there and in Fields (2017). The models were chosen originally based on their stability parameter Q, as we desired to compare the amount of chaotic mixing (a local property, measured by the Lyapunov exponent) with the global stability Q. This stability parameter Q is the Toomre stability criterion (Toomre 1964) and gives a measure of the balance between gravitational forces and the forces that oppose gravity, such as pressure. For gaseous disks, Q balances pressure given by the speed of sound and the surface density, which is an extension of the gravitational force (see Ch. 1). Values of Q less than 1 are seen as unstable, and values of Q greater than 1 are stable to collapse. Models were chosen in

Fields (2017) such that a range of Q values were available.

Table 3.1: The parameters of the 9 models initially-selected from Foyle (2007) for which full simulations were run in Fields (2017). Only a subset of these models are further discussed in this work, and the parameters of these models are more thoroughly discussed in both Foyle (2007) and Fields (2017). Model 63 is bolded as it is the model considered for most tests in this thesis.

Model	λ_0	m_d	m_b	V_{200} [km/s]	c	h [kpc]	Q
63	**0.02**	**0.050**	**0.000**	**180.0**	**5**	**1.90**	**0.751**
64	0.02	0.050	0.010	180.0	5	1.63	0.787
75	0.02	0.050	0.000	180.0	15	1.20	0.805
118	0.03	0.050	0.010	180.0	5	3.03	0.893
136	0.03	0.100	0.020	180.0	5	1.92	0.762
147	0.03	0.100	0.000	180.0	15	1.47	0.767
165	0.08	0.025	0.000	180.0	15	6.11	2.91
183	0.08	0.050	0.000	180.0	15	5.82	1.78
202	0.08	0.100	0.020	180.0	15	5.00	1.20

Table 3.1 shows the nine models from Foyle (2007) from which simulations were run and were described in Fields (2017). This table gives the model number, the dimensionless spin parameter that sets the spin of the dark matter halo λ_0 (eq. 1.1), the fractional mass of the disk component m_d, the fractional mass of bulge component m_b, the circular velocity at the virial radius V_{200}, the concentration parameter of the halo c, the scale length of the distribution h, and the stability parameter Q. In this work, we mainly focus on Model 63, a very unstable disk with $Q = 0.751$.

3.3 General overview of problems encountered with the tracers

In Fields (2017), simulations conducted with the code HYDRA were analyzed initially without considering any detailed analysis of errors within the tracers and their associated algorithm. However, it became evident that the evolution of the tracers was not as expected when the tracer temperatures were not fixed at $e_{\min}$, but rather allowed to vary. During the evolution of the modeled disk galaxies in Fields (2017), some tracers were ejected above or below the disk, causing a chain reaction of more tracers being added, as the length of the contour grew uncontrollably and incorrectly. This can be seen in Figure 3.2. This figure shows the contour of Model 63 from Fields (2017) at 400 Myr into the evolution of the disk. The tracer particles are shown as blue stars and the gas particles are plotted as red dots in the x-y and x-z planes. Here, hundreds of tracers are found above and below the disk, as evident in the x-z plane. Therefore, final measurements of the length of the contour over time and thus the Lyapunov exponents in Fields (2017) were incorrect for some models. This thesis attempts to identify and solve the issues with the advection of the tracer particles over time.

By inspection of the temperatures, velocities, and other properties of the tracer particles as they moved through the disk, we determined that overheating of the tracer particles caused their velocities to become too large and thus they were ejected from the disk. Some tracers were heated to greater than 10^5 degrees Kelvin and were unable to properly cool because they were in low-density regions while being overheated, and the rate of cooling is proportional to the square of the density (White & Rees 1978; Foyle 2007). These large

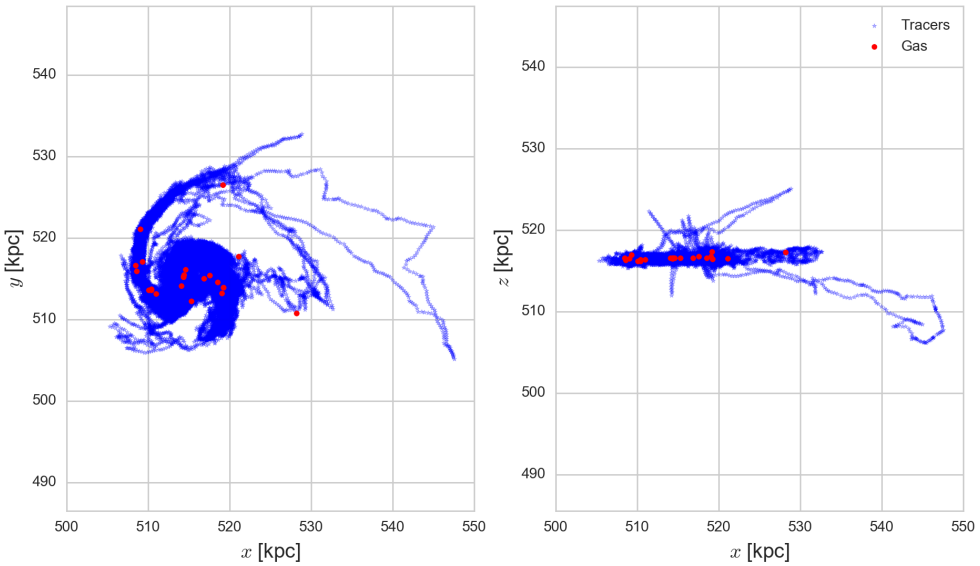

Figure 3.2: The contour from the simulation of Model 63 from Fields (2017) at 400 Myr into the evolution of the disk. The tracer particles are shown as blue stars and the gas particles in the contour are shown as red dots. It is clear in the x-z plane that the tracers are ejected far from the plane of the disk, thus causing the measurement of the change in length of the contour to be unusable.

temperature increases occurred for tracers in low-density regions around shock fronts, like the one that occurs around 250 Myr in the evolution of Model 63, as shown in Figure 3.3. Due to radiative cooling, these tracer particles should not have been heated to greater than 10^4 K, so we were motivated to attempt to pinpoint the cause of the problems and then attempt to resolve the issues.

As seen in Fig. 3.3, there was a loop of particles at about 250 Myr into the evolution of Model 63 that was propelled forward by a shock. This loop is indicated by the orange oval in the figure. The shock front continued to propagate in the positive x-direction and in the bottom panel at 275 Myr, it can be seen how particles start to stream out of the disk

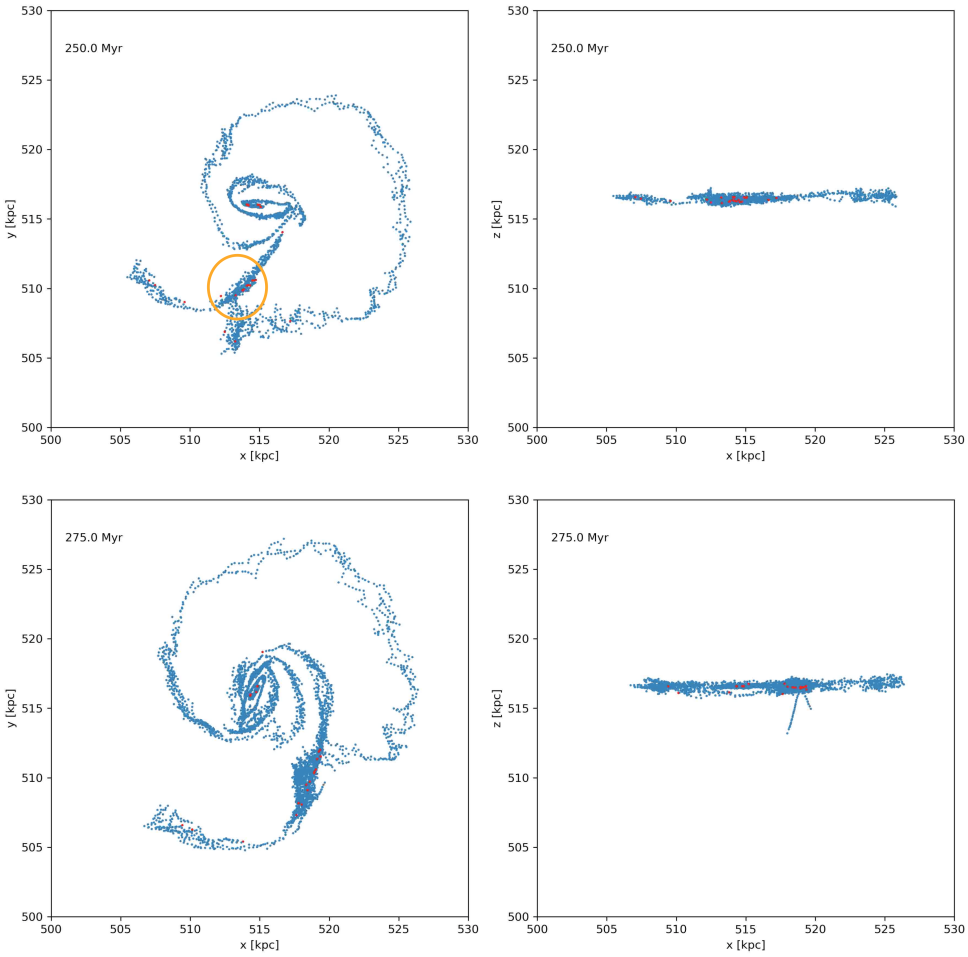

Figure 3.3: The contour evolution is shown at 250 Myr (top) and 275 Myr (bottom) for Run 1163, which is a copy of the Model 63 evolutionary run, similar to that found in Fields (2017). The x-y and x-z planes are shown, where the red dots are the original gas particles from which the contour was chosen and the blue dots are the tracer particles that have been added to the contour. The orange oval in the x-y plane at 250 Myr highlights the area of the shock front, where the tracers particles start to experience errors, leading to them being ejected from the disk, seen in the 275 Myr snapshot.

in the x-z plane. The problem here is how the tracers collide with tracers and gas particles already in the stream rather than about the specific string of tracers in this model. When the tracer particles are ejected from the disk in the z-direction as seen in Fig. 3.3, it became impossible to use the length of the contour growth as a proxy for the amount of mixing taking place on small scales, which was the ultimate goal of using the tracers.

We can classify tracer integration issues as having two types of errors, which we will call Type I and Type II errors. **Type I** errors are those in which tracer particles are ejected from the plane of the disk, such as can be seen in in the x-z plane in Fig. 3.2. This type of error is most obvious to see when visually inspecting the evolution of the contour over simulation time and results from the tracers experiencing too much shock heating. **Type II** errors are the opposite of Type I, in that they occur when the tracers do not experience enough shock heating. In this type of error, the tracers pass right through a shock region without being affected and therefore do not adequately follow the flow. The Type II error does allow integration to continue without an obvious visual problem, but it does so with an error in the way that the contour follows the flow.

In the following sections, algorithmic changes to our simulation method, outlining attempts to prevent tracers from becoming over-heated, are described. We consider changes from a number of different algorithm perspectives, including modifying the artificial viscosity felt by tracers, the mass of the tracer particles, the method which tracers are initially assigned velocities, the way acceleration is calculated for tracers, the timestep normalization for the simulation, how the minimum or maximum temperatures affect the tracers, and the effect of limiting temperature increases based on percentage increase or the local density. In

each section, the method of change, analysis, and results are presented in lieu of having a final results section at the end of the chapter.

When evaluating the nine simulations found in Fields (2017), Model 63 was one that stood out due to its extreme shock front and the numbers of ejected tracer particles seen late in the simulation. Because of the difficulty presented by this method, we focused our work in this chapter on Model 63 from Fields (2017). Other models are more likely to be more stable to tracer evolution due to their higher Q values.

3.4 Artificial viscosity changes for tracer particles

As noted in the previous section, the amount of shock heating affected the accuracy of the tracers following the flow of the gas particles, so we first looked at changes in the artificial viscosity of the tracers. In SPH, an artificial viscosity is added in order to capture shocks in real physical situations to convert kinetic energy into thermal energy for gas particles in a converging flow (Monaghan 1992; Thomas & Couchman 1992; Thacker et al. 2000; Fardal et al. 2019). The artificial viscosity is used to dissipate convergent motions and is added to the pressure term in the force calculation (Thacker et al. 2000). It is given as $q_{ij}\Pi_{ij}$ where q_{ij} is the multiplicative prefactor that we change for the tracers and Π_{ij} is the amount of artificial viscosity applied to the particle. If q_{ij}=0.9 for example, then the artificial viscosity experienced by the tracers in that simulation is 0.9 times the artificial viscosity applied to the gas particles in the same simulation. Following Thacker et al. (2000), which built upon the work of Gingold & Monaghan (1977); Thomas & Couchman (1992); Couchman et al.

(1996), Π_{ij} is given as

$$\Pi_{ij} = \frac{-\alpha\mu_{ij}\bar{c}_{ij} + \beta\mu_{ij}^2}{\bar{\rho}_{ij}} \tag{3.1}$$

where c is the speed of sound, α and β are coefficients that are typically given values $\alpha = 1$, $\beta = 2$, the bar denotes the average of the quantity for two particles i and j, and μ_{ij} is given as the following

$$\mu_{ij} = \begin{cases} \bar{h}_{ij}\boldsymbol{v}_{ij} \cdot \boldsymbol{r}_{ij}/(r_{ij}^2 + \nu^2), & \boldsymbol{v}_{ij} \cdot \boldsymbol{r}_{ij} < 0; \\ 0, & \boldsymbol{v}_{ij} \cdot \boldsymbol{r}_{ij} \geq 0 \end{cases} \tag{3.2}$$

where $\nu^2 = 0.01\bar{h}_{ij}$ is included to prevent numerical divergences, $\boldsymbol{r}_{ij}$ is the separation in position and $\boldsymbol{v}_{ij}$ is the relative velocity of the particles. Given the tracer overheating and ejections, it was hypothesized that the tracers needed a different treatment of their artificial viscosity. This change in the artificial viscosity was hoped to prevent the tracers from being ejected out of the system and so was implemented with the prefactor q_{ij} mentioned above. With a lower viscosity, the tracers should more easily flow through the fluid and thus be less likely to be ejected due to reduced heating when encountering a shock front or extreme density changes.

Table 3.2 shows the variations in artificial viscosity prefactor q_{ij} that were tested. All runs in the table have a normal timestep normalization (`dtnorm`=1.0) and include the velocity smoothing algorithm `velsmooth` to calculate the velocity of a tracer when it is initially added to the contour. The acceleration of the tracers is calculated by a pairwise calculation, and these simulations do not force the tracers to have a particular temperature, such as $e_{\min}$ in Fields (2017), but rather allow the tracers to vary in temperature following a more

Table 3.2: Run IDs for the variations of Model 63 with varying artificial viscosity for the tracers. Here, q_{ij} is a multiplicative factor by which the tracers feel q_{ij} times the artificial viscosity of the gas particles. Also given in the remaining columns are the timestep normalization ('dtnorm', §3.9), method of velocity and acceleration calculations (§3.6, 3.7), the tracer mass ($m_{\rm tracer}$, §3.5), and minimum and maximum allowed temperatures for gas and tracer particles (§3.8).

Run ID	Tracer q_{ij}	`dtnorm`	Velocity Calculation	Acceleration Calculation	$m_{\rm tracer}$ [$M_\odot$]	$e_{\rm min}$ [K]	$e_{\rm max}$ [K]
1163	1.0	1.0	`velsmooth`	Pairwise	100	1000	None
9163	0.9	1.0	`velsmooth`	Pairwise	100	1000	None
7163	0.75	1.0	`velsmooth`	Pairwise	100	1000	None
1063	0.5	1.0	`velsmooth`	Pairwise	100	1000	None

realistic approach. The tracers in these simulations have $m_{\rm tracer} = 100 M_\odot$ which is approximately equal to $0.0005 m_{\rm gas}$ where $m_{\rm gas}$ is the mass of a single gas particle. There is no temperature ceiling in these simulations, but there is a temperature floor of $e_{\rm min} = 1000$ K for the gas and tracer particles.

Fig. 3.4 shows the contours of the runs from Table 3.2 at 270 Myr in the x-y and x-z planes. Here, the first thing to notice is that Run 1163 (panel (a)) with the highest q_{ij} multiplicative factor for tracers has the earliest tracer ejection seen in the x-z plane. The tracers in this simulation experiencing too much shock heating, and are ejected early. Unfortunately all other simulations in this section have particles ejected from the disk, too, but at later evolutionary times. Run 1063 (panel (d)) has the lowest q_{ij} for tracers, and at 270 Myr, there are not yet any tracers being ejected from the disk, but the tracers have moved through the shock region too much (not enough shock heating) and there are extra filaments all around the contour, as in a Type II error. This highlights the challenges of establishing the right amount of shocking in the tracer population. Too little, via low q_{ij}

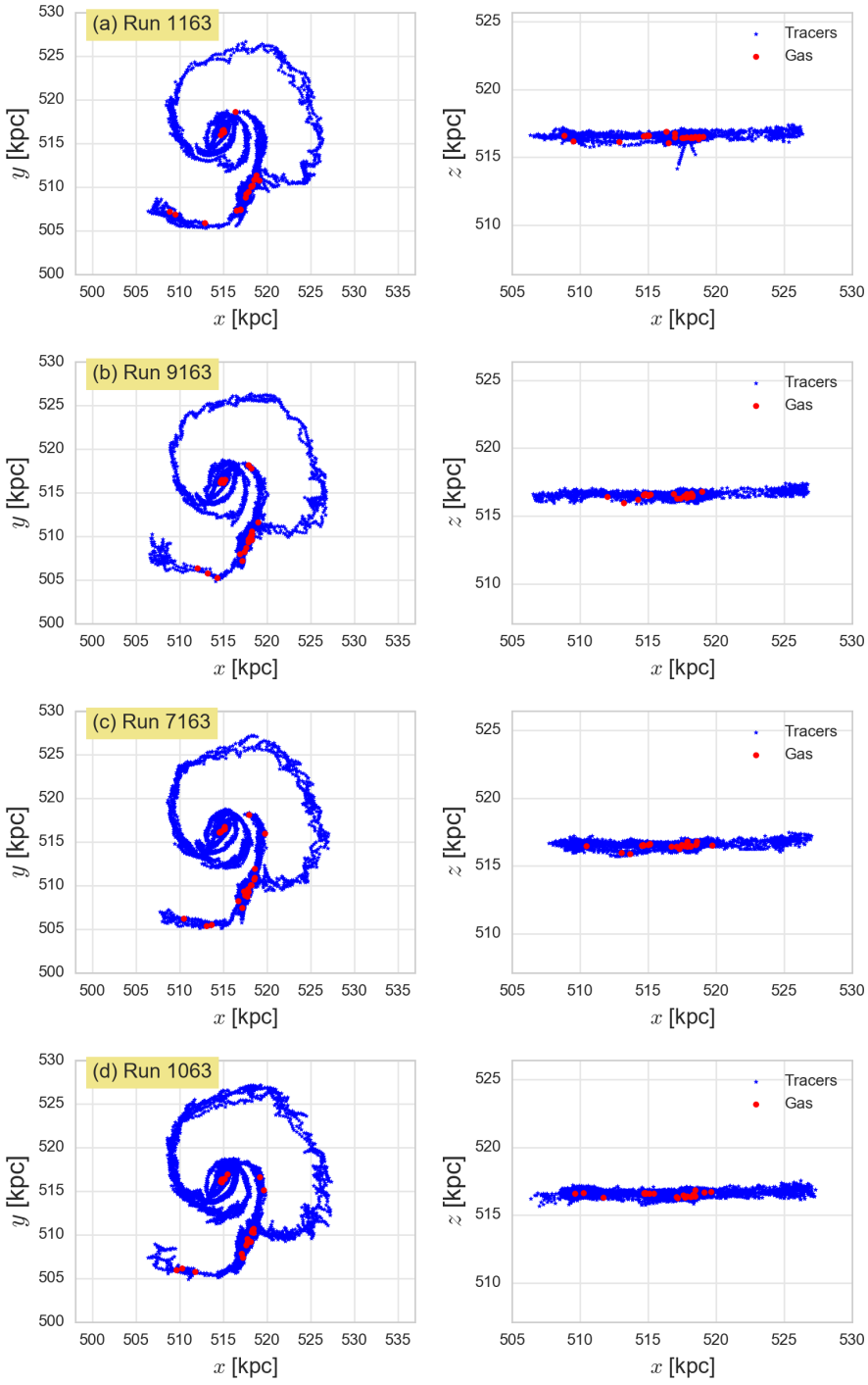

Figure 3.4: The contour at 270 Myr for panel (a): Run 1163, panel (b): Run 9163, panel (c): Run 7163, and panel (d): Run 1063 in the x-y plane (left) and the x-z plane (right). The blue stars are the tracer particles in the contour, and the red dots are the location of the gas particles in the contour.

value, leads to Type II errors even if particles are not ejected.

3.5 Effect of tracer mass

Normally, tracers are separate entities from the main simulation and merely interpolate fields to calculate evolution. However, SPH is a sampling method by nature and so can be easily adapted to mimic tracer behaviour subject to modifying the equation of motion to remove tracers from kernel summations among other considerations. In doing so, the fact that mass does appear in the equation of motion makes it sensible to give the tracers a small but non-zero mass. A test of appropriate tracer mass was conducted, and the four variations in tracer mass tested can be found in Table 3.3. We predicted that little difference would be seen between the given models due to the small dependence on the tracer mass in the calculations.

Table 3.3: The variations of Model 63 used to compare the effect of tracer mass on the overall evolution, where $m_{\rm gas}$ is the mass of a single gas particle. Note that Run 1163 had a tracer mass of $100 M_\odot$, which is equal to approximately $0.0005 \times m_{\rm gas}$. Also given in the remaining columns are the tracer q_{ij} (§3.4), timestep normalization ('dtnorm', §3.9), method of velocity and acceleration calculations (§3.6, 3.7), and minimum and maximum allowed temperatures for gas and tracer particles (§3.8).

Run ID	Tracer Mass	Tracer q_{ij}	`dtnorm`	Velocity Calculation	Acceleration Calculation	$e_{\rm min}$ [K]	$e_{\rm max}$ [K]
1163	$0.0005 m_{\rm gas}$	1.0	1.0	`velsmooth`	Pairwise	1000	None
1263	$0.001 m_{\rm gas}$	1.0	1.0	`velsmooth`	Pairwise	1000	None
1363	$0.01 m_{\rm gas}$	1.0	1.0	`velsmooth`	Pairwise	1000	None
1463	$0.1 m_{\rm gas}$	1.0	1.0	`velsmooth`	Pairwise	1000	None

Fig. 3.5 shows the differences in contour between the four simulations at 270 Myr in the

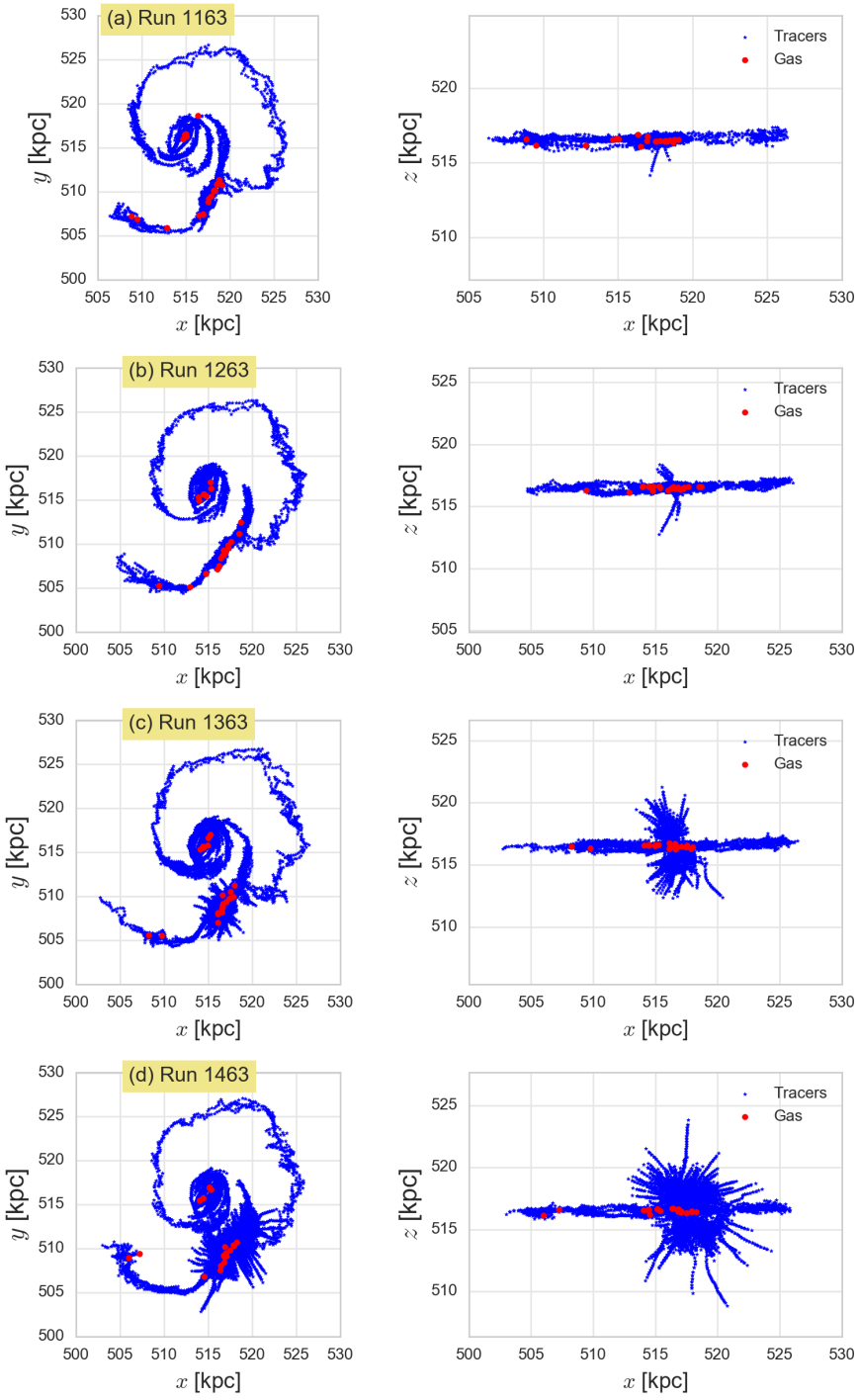

Figure 3.5: The contour at 270 Myr for panel (a): Run 1163, panel (b): Run 1263, panel (c): Run 1363, and panel (d): Run 1463 in the x-y plane (left) and the x-z plane (right). The blue stars are the tracer particles in the contour, and the red dots are the location of the gas particles in the contour.

x-y and x-z planes. Our initial assumption that the mass would have a negligible effect on the evolution of the contour was incorrect, as we clearly see the frequency of Type I errors increase as the mass of the tracers increases at the same evolutionary time. The goal was to use the smallest reasonable mass, so for all subsequent simulations, a tracer mass of 100 $M_\odot$, or about 0.05% of the mass of a single gas particle was used.

3.6 Using velocity smoothing versus pairwise velocity calculations for new tracer particles

Table 3.4: Runs used to compare the effects of using the velocity smoothing algorithm versus pairwise velocity calculations. Also given in the remaining columns are the tracer q_{ij} (§3.4), timestep normalization ('dtnorm', §3.9), method of acceleration calculation (§3.7), the mass of the tracers ($m_{\rm tracer}$ §3.5), and minimum and maximum allowed temperatures for gas and tracer particles (§3.8).

Run ID	Velocity Calculation	Tracer q_{ij}	`dtnorm`	Acceleration Calculation	$m_{\rm tracer}$ [$M_\odot$]	$e_{\rm min}$ [K]	$e_{\rm max}$ [K]
2063	`velsmooth`	0.5	0.25	Pairwise	100	1000	None
2163	Pairwise	0.5	0.25	Pairwise	100	1000	None

As introduced in Chapter 2, we used velocity smoothing `velsmooth` as a method to calculate the velocity of a tracer particle during the first iteration that it is inserted into the simulation. This method smooths the velocities of the neighboring particles in order to determine the appropriate velocity of the tracer particle being inserted at a given position, as eq. (2.4) illustrates. The pairwise velocity calculation, eq. (2.3), on the other hand, uses the information only of the two neighboring particles already in the contour, rather than

a full sphere of neighboring particles. As shown in Table 3.4, in Run 2063, the velocity smoothing algorithm `velsmooth` was used to calculate the initial velocity that the tracers would be given when inserted into the contour, and in Run 2163, the pairwise velocity was used during the single timestep where the tracer was added instead. Around a shock region, there can be sudden deceleration to keep the flow ordered, and when the contour is in this region, it is possible that the pairwise velocity might have a large difference relative to the mean flow in the area.

Fig. 3.6 shows the difference in velocity calculated between the `velsmooth` v_s and pairwise velocity v_p algorithms for all 2187 gas and tracer particles that were saved in the contour at iteration 9000 of Run 2063, a run with timestep normalization of 0.25 and q_{ij}=0.5 for tracers (Table 3.4). This late iteration was chosen so that there would be no bias in the direction that the particles of the contour were traveling at that time. If particles from an initial contour were chosen, there would have been a bias in the x-direction as the contour moved counterclockwise through the disk. In the figure, the absolute difference between the `velsmooth` velocity v_s and the pairwise velocity v_p divided by the maximum of the absolute value of v_s and v_p is shown for all particles, with Particle ID from 0 to 2186 shown on the x-axes. The figure shows that there is overall a surprisingly good agreement between v_p and v_s for a majority of the tracers in the three components of velocity, v_x, v_y, and v_z. In the figures, a value of 2.0 on the $|v_s - v_p|/\max(|v_p|,|v_s|)$ axis means that v_s and v_p are of equal magnitude but pointed in exactly opposite directions. In v_x, only 8.1% of all particles have $|v_s - v_p|/\max(|v_p|,|v_s|) \geq 0.5$, 6.6% in v_y, and 36.5% in v_z. However, neither v_s or v_p can be said to be the "correct" velocity, but rather this figure just helps to illustrate

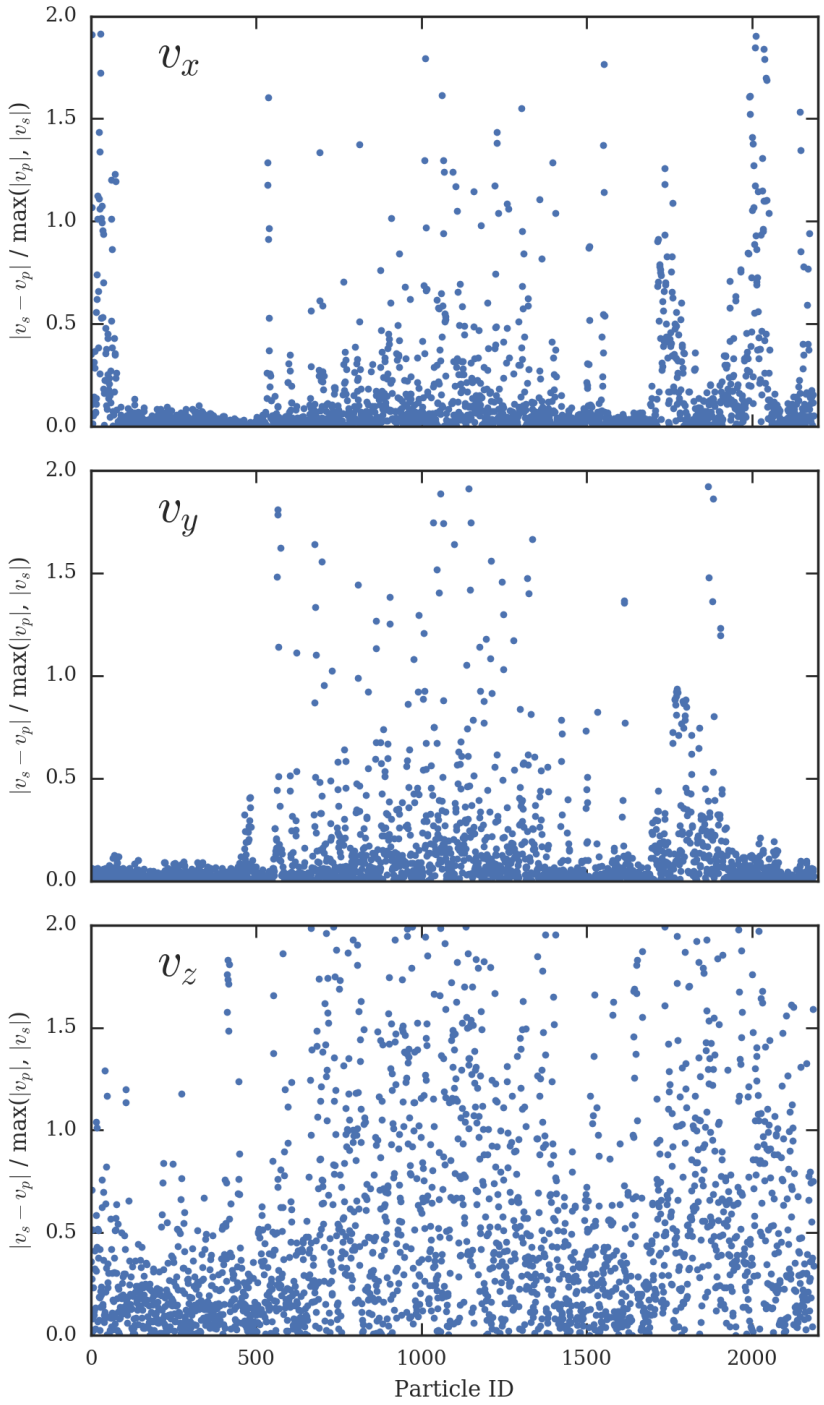

Figure 3.6: The velocities calculated from both 'velsmooth' (v_s) and pairwise velocity (v_p) methods from iteration 9000 of Run 2063. The top, middle, and bottom panels show the x, y, and z components of velocity respectively. For a majority of the particles, the difference between the two calculated velocities is small.

the differences due to the calculations.

In §2.3.1 discussing the spherical collapse model, it was determined that there was very little difference between the model that did include `velsmooth` and the one that did not, likely because the shock front was well-resolved in that simulation. Surprisingly, however, there was a considerable difference in the evolution of the contour of tracer particles between Run 2063 and Run 2163 when disk models were examined. The difference in the initial velocity of the tracers was the only difference between the two runs, but Run 2163 had *no* tracer particles get over-heated or ejected from the disk during its 285 Myr evolution. This surprising observation shows just how reliant on small changes tracer algorithms can be. It must be emphasized, however, that some of the other simulations in other sections (for example in §3.4) did not start showing signs of Type I errors until later in their evolution ($\geq$300 Myr). Despite not visually seeing Type I errors in the contour evolution of Run 2163, Type II errors were still experienced by the tracers in the contour.

Fig. 3.7 shows the very small difference that happened early on in the contour that led to the significant difference in evolution. Early in the simulation, there was a single tracer particle that was in a slightly different position in Run 2163 compared to Run 2063. That single tracer eventually led to the Type I errors in the tracer method in Run 2063, as it caused an extended arm to be developed which then moved through a very low-density region, causing the tracers to overheat around the shock fronts created. This extended arm can be considered a Type II error, as the tracer particles are incorrectly moving through a region of gas particles. In the figure, the top panels show the contour of particles in Run 2063 and the bottom panels show Run 2163 at 100 Myr (left) and 215 Myr (right). An orange

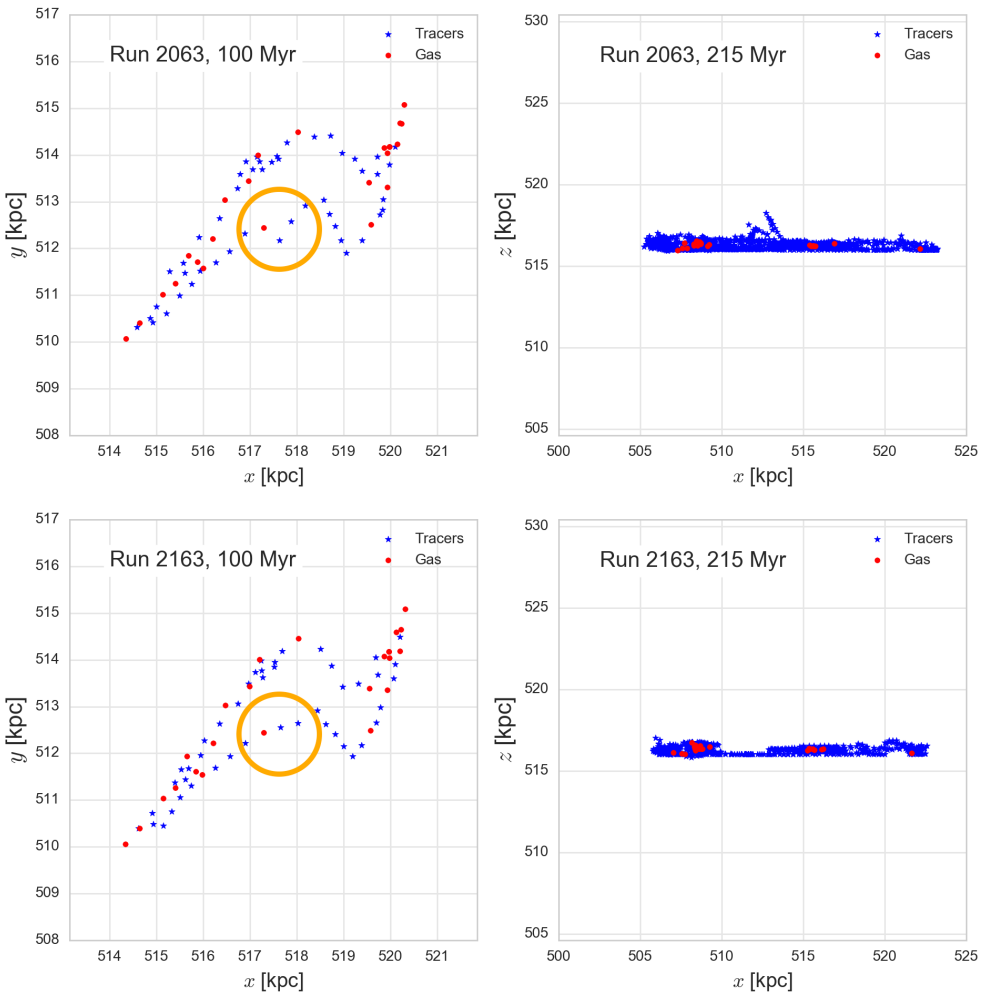

Figure 3.7: The tracked contour of Run 2063 (top) and Run 2163 (bottom) shown at 100 Myr (left) and 215 Myr (right). There is a single particle in the 100 Myr contour that is substantially different between the two runs, highlighted by the orange circle, leading to the eventual catastrophic failure of the tracer method in Run 2063, which can be seen as early as 215 Myr in the right panels. At 215 Myr, Run 2063 (top) already has tracers being ejected from the disk, whereas Run 2163 (bottom) does not have any ejected tracers.

circle highlights the main difference in the tracer particles at 100 Myr for both Run 2063 and Run 2163, and it is this small difference which leads to particles being ejected from the disk as early as 215 Myr into the evolution of Run 2063, as seen in the top right panel. At 215 Myr in Run 2163 (bottom right), there are no such tracers being ejected, and this run continued to 285 Myr without any particles being ejected from the disk. The delicate reliance on small and individual tracers shows how sensitive the tracer contour algorithm was.

3.7 Using acceleration smoothing and passive tracers instead of active tracers

For the spherical collapse model in §2.3.1, the effects of both velocity smoothing method `velsmooth` and acceleration smoothing method `asmooth` were tested and compared to a model without either of the smoothing methods. In the previous chapter, it was determined that the radial velocity profiles of the gas and tracer particles showed that smoothing the acceleration field produced inaccuracies by underestimating the change in the velocity field near shock fronts. Thus, since the `asmooth` method did not track velocity fields accurately in well-resolved, well-controlled isothermal spherical collapse model, we expected to see similar behaviour when testing the acceleration smoothing algorithm `asmooth` with our more complex disk model, the net behaviour being this method would not adequately slow particles near shocks and potentially produce Type II errors.

Recall, `asmooth` is a method that needed to be used for every calculation of tracer particle

acceleration. It was based on the SPH smoothing method but changed appropriately to calculate the smoothed acceleration field. We describe tracers in this method as passive rather than active since they do not participate in pairwise evaluations of force. Genel et al. (2013) described some problems of passive tracers in their work, such as inconsistent density fields between the tracer and gas particles, indicating poor following of the flow.

Table 3.5: Run IDs to determine the effects of acceleration smoothing. Also given in the remaining columns are the tracer q_{ij} (§3.4), timestep normalization ('dtnorm', §3.9), method of velocity calculation (§3.6), the mass of the tracers ($m_{\rm tracer}$ §3.5), and minimum and maximum allowed temperatures for gas and tracer particles (§3.8).

Run ID	Acceleration Calculation	Tracer q_{ij}	`dtnorm`	Velocity Calculation	$m_{\rm tracer}$ [$M_\odot$]	$e_{\rm min}$ [K]	$e_{\rm max}$ [K]
2063	Pairwise	0.5	0.25	`velsmooth`	100	1000	None
9363	`asmooth`	0.5	0.25	`velsmooth`	100	1000	None

Table 3.5 shows the two runs used to determine the effects of including `asmooth` in the particle calculations. Here, Run 2063 is used as the 'control' and Run 9363 is the same as 2063 except that it included acceleration smoothing instead of pairwise hydrodynamic forces for the tracers. For each run, the timestep normalization `dtnorm` is 0.25 and the artificial viscosity that the tracers feel is 0.5 times that which the gas particles feel, or q_{ij}=0.5. Both runs were started with the same initial conditions.

Figs. 3.8 and 3.9 show the contours of Runs 2063 and 9363 respectively at four evolutionary snapshots in both the x-y and x-z planes. When examining the evolution of the contour of Run 9363 visually compared to Run 2063, the errors with the passive tracer particles are evident. The tracer particles flow through shocks without properly decelerat-

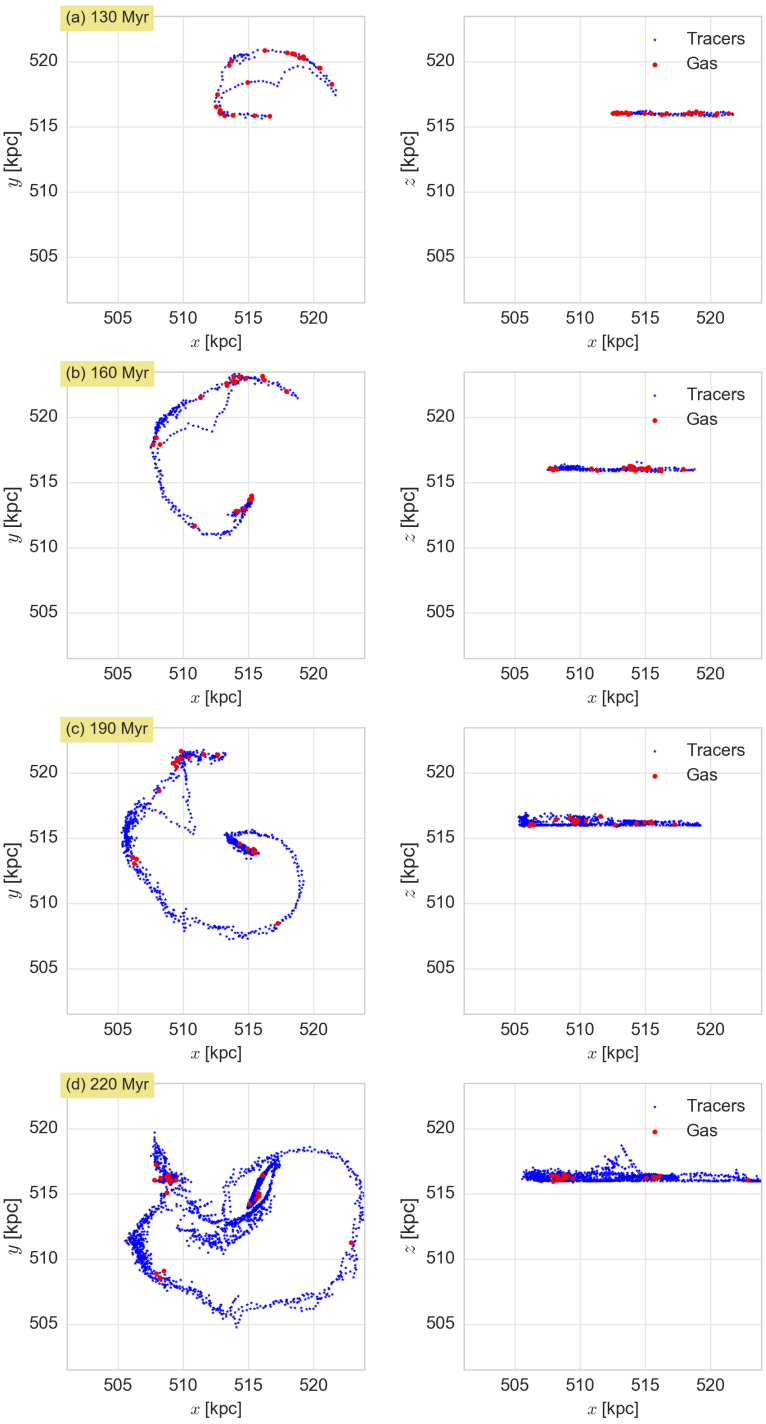

Figure 3.8: The evolution of the contour of **Run 2063** at four evolutionary snapshots in the x-y plane. The tracers shown here are active tracers, as their forces and accelerations are evaluated in the SPH and gravity calculations

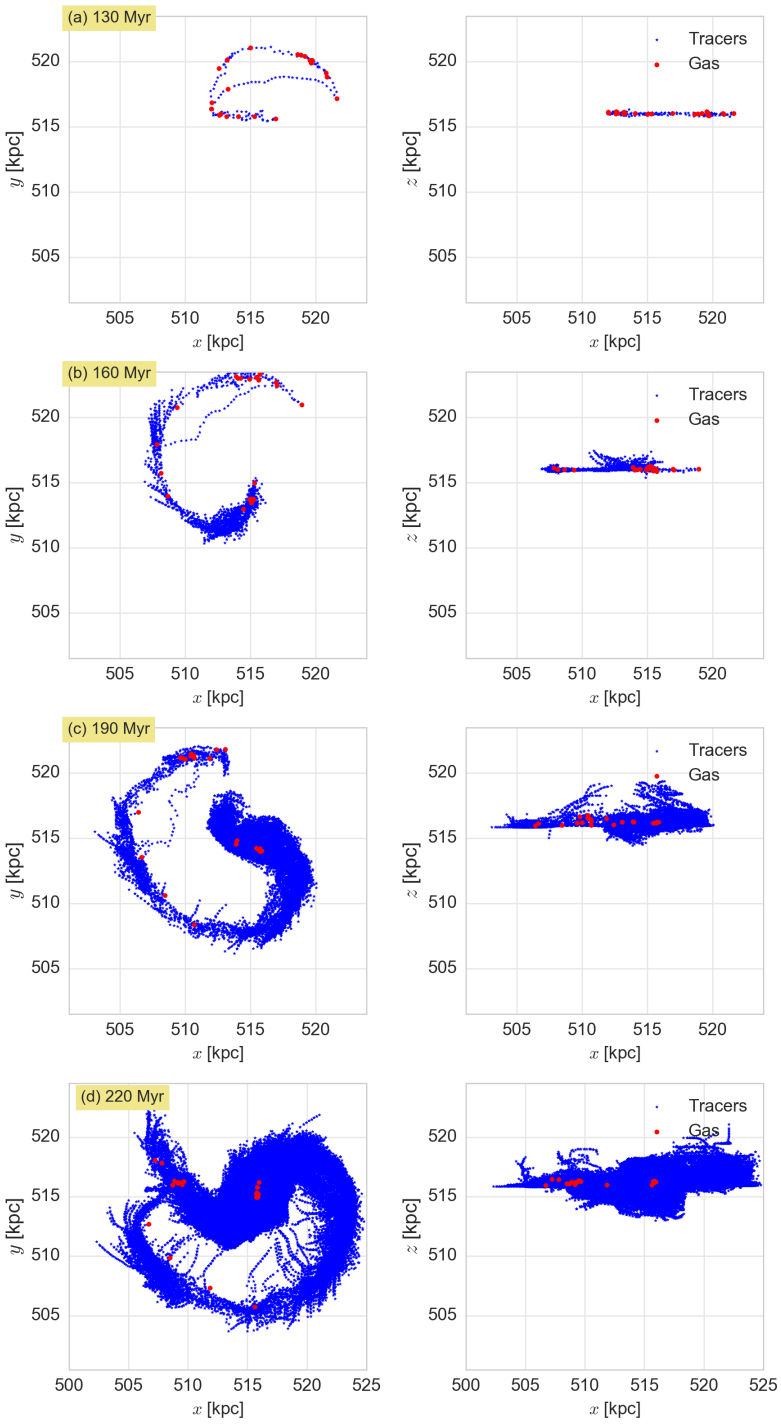

Figure 3.9: The evolution of the contour of **Run 9363** at four evolutionary snapshots in the x-y plane. The tracers shown here are passive tracers, as the calculation of acceleration is done by acceleration smoothing and bypasses the SPH calculation. These passive tracers show substantial Type II errors.

ing, causing the overall trajectories of the tracers to very poorly follow the flow of the gas particles in the disk. Additionally, there are many extra filamentary arms seen in the Run 9363 snapshots of Fig. 3.9. These extra additions in the contour add length that does not accurately reflect the amount of mixing seen in the gas. Overall, it is determined that the acceleration smoothing algorithm for the tracers causes much more significant Type I and Type II errors in the tracers.

3.8 Minimum and maximum temperatures for gas and tracer particles and limiting temperature increase based on density

A variety of effects related to heating are tested in the following subsections, such as the effect of changing the minimum or maximum allowed temperatures for gas and tracer particles or limiting the temperature growth based on a per cent temperature increase or density in the area of a particle during each timestep for just the tracers. Limiting cooling or heating has been used in simulations to improve stability (e.g. Katz et al. (1996)), and we are motived here by that previous work to use temperature limitations to control the relative strengths of shocks and shock heating. The challenge here is that the actual physics being implemented has more temperature and density dependence than we can implement in the simulation due to our lack of dynamical range.

Table 3.6: Run IDs along with their minimum $e_{\rm min}$ and maximum $e_{\rm max}$ allowed temperatures. Also given in the remaining columns are the tracer q_{ij} (§3.4), timestep normalization ('dtnorm' §3.9), the method of velocity and acceleration calculation (§3.6, 3.7), and the mass of the tracers ($m_{\rm tracer}$ §3.5).

Run ID	$e_{\rm min}$ [K]	$e_{\rm max}$ [K]	Tracer q_{ij}	`dtnorm`	Velocity Calculation	Acceleration Calculation	$m_{\rm tracer}$ [$M_\odot$]
2791	1000	None	1.0	1.0	Pairwise	Pairwise	100
2792	5000	None	1.0	1.0	Pairwise	Pairwise	100
4101	1000	None	0.5	0.25	`velsmooth`	Pairwise	100
4102	1000	15,000	0.5	0.25	`velsmooth`	Pairwise	100

3.8.1 Changing the minimum and maximum allowed temperatures

Both the effect of changing the temperature floor of the simulation $e_{\rm min}$ and the temperature ceiling $e_{\rm max}$ for gas and tracer particles was tested in this section. Table 3.6 shows a summary of the simulations considered here. Runs 4101 and 4102 are variations of Model 63 and Run 2063 (described in §3.7, 3.6, 3.8.2). Run 4101 was restarted from Run 2063 at 195 Myr, or 7530 iterations into the evolution, and the differences between Run 4101 and Run 2063 are that extra information about individual gas and tracer particles was saved to a file for extra analysis (see Chapter 4) and the refinement criterion was modified to help the simulation evolve more quickly on our machines. Run 4102, other than changing the $e_{\rm max}$, was a direct copy of Run 4101, but this run was started from the initial conditions of Model 63, not part-way through the simulation.

Runs 2791 and 2792 use a different set of initial conditions than found in Runs 4101, 4102, and the rest of this chapter, but they represent very early testing conducted to discover the effects of changing the minimum temperature allowed in a simulation. These runs used

a maximally unstable model that had a bulge but no stars in its disk. This model had 200,000 total particles, composed of 100,000 dark particles, 80,000 gas particles in a disk, and 20,000 star particles in a bulge around the galactic centre.

First, the differences between 2791 and 2792 are examined to determine if $e_{\min}$ affects the evolution of the tracers. Fig. 3.10 shows the difference in the evolution of the contour between Run 2791 (top) and Run 2792 (bottom) at the same evolutionary time. It is clear that Run 2792 with a higher $e_{\min}$ seems to have a greater pressure support in the interstellar medium, as the contour is pushed outwards from the centre relative to Run 2791 with a lower $e_{\min}$. Setting a higher minimum temperature forces energy to be maintained in the simulation which can be viewed as a net energy input into the interstellar medium. We were motivated to continue to use the lower temperature floor by considering the average temperature of the cold and warm phases of the ISM, so we set the somewhat high $e_{\min}$ of 10^3 K. The cold phase of the ISM is typically 10^2 K and the warm phase is 10^4 K, and having some minimum temperature between the two was computationally necessary.

Next, a maximum allowed temperature, $e_{\max}$, was implemented in the simulation to determine if preventing gas and tracer particles from becoming too heated would stabilize the evolution of the tracer particles. In Run 4102 in Table 3.6, we let $e_{\max} = 15,000$K and compared it to Run 4101 where there was no maximum temperature for the gas and tracer particles. The maximum temperature of 15,000 K was chosen as it is above 10^4 K that particles should have quickly cooled down to, as it is the temperature at which Hydrogen dominates cooling in cascading recombinations, so particles should not heat to much above this temperature.

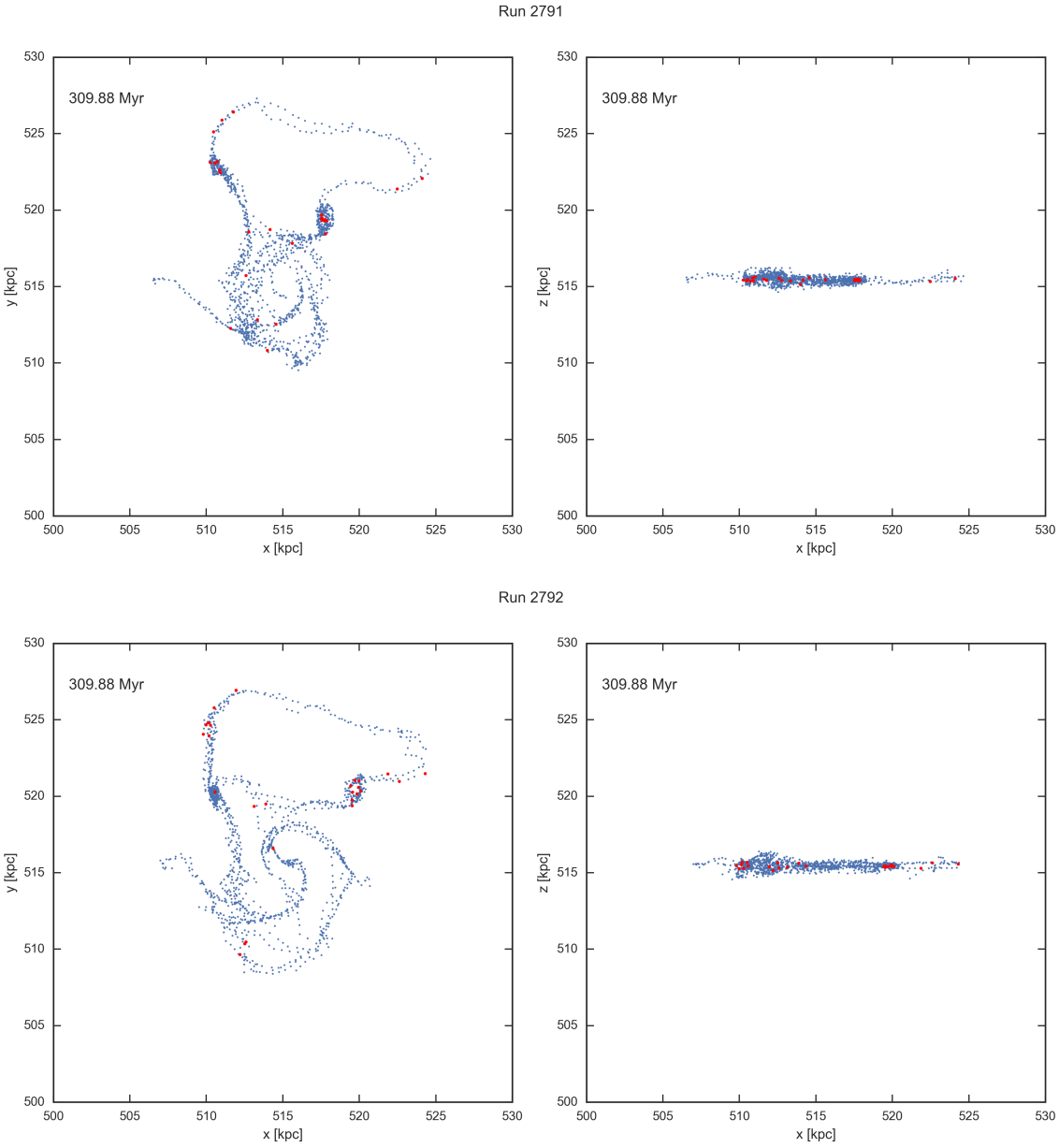

Figure 3.10: Contours from Run 2791 (top) and Run 2792 (bottom) are shown at 309.88 Myr into their evolution in the x-y plane (left) and the x-z plane (right). The red dots are the original gas particles and the blue dots are the tracers that have been added to the contour.

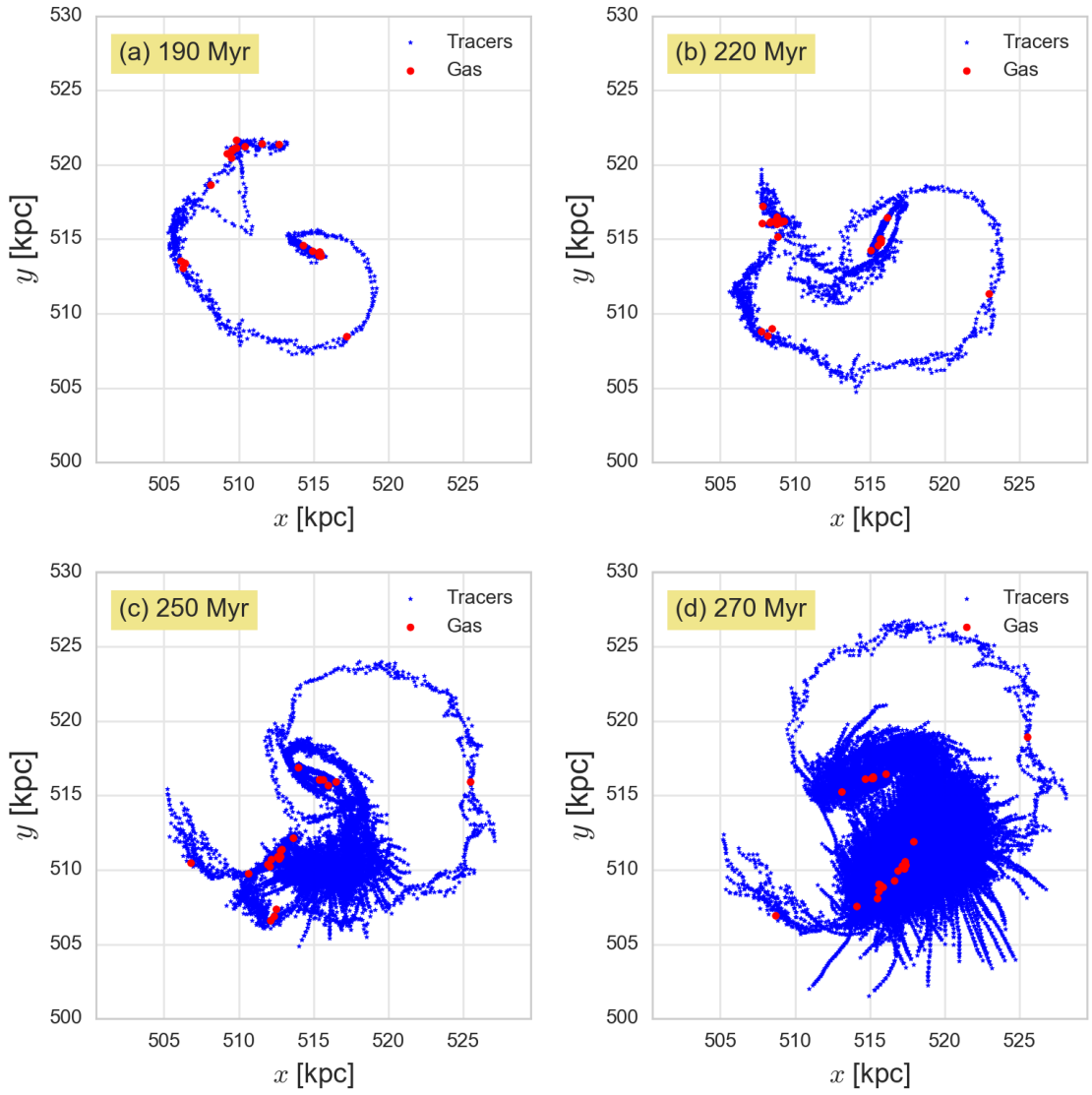

Figure 3.11: The contours of tracer (blue star) and gas (red circle) particles at (a) 190 Myr, (b) 220 Myr, (c) 250 Myr, and (d) 270 Myr for **Run 4101** in the x-y plane.

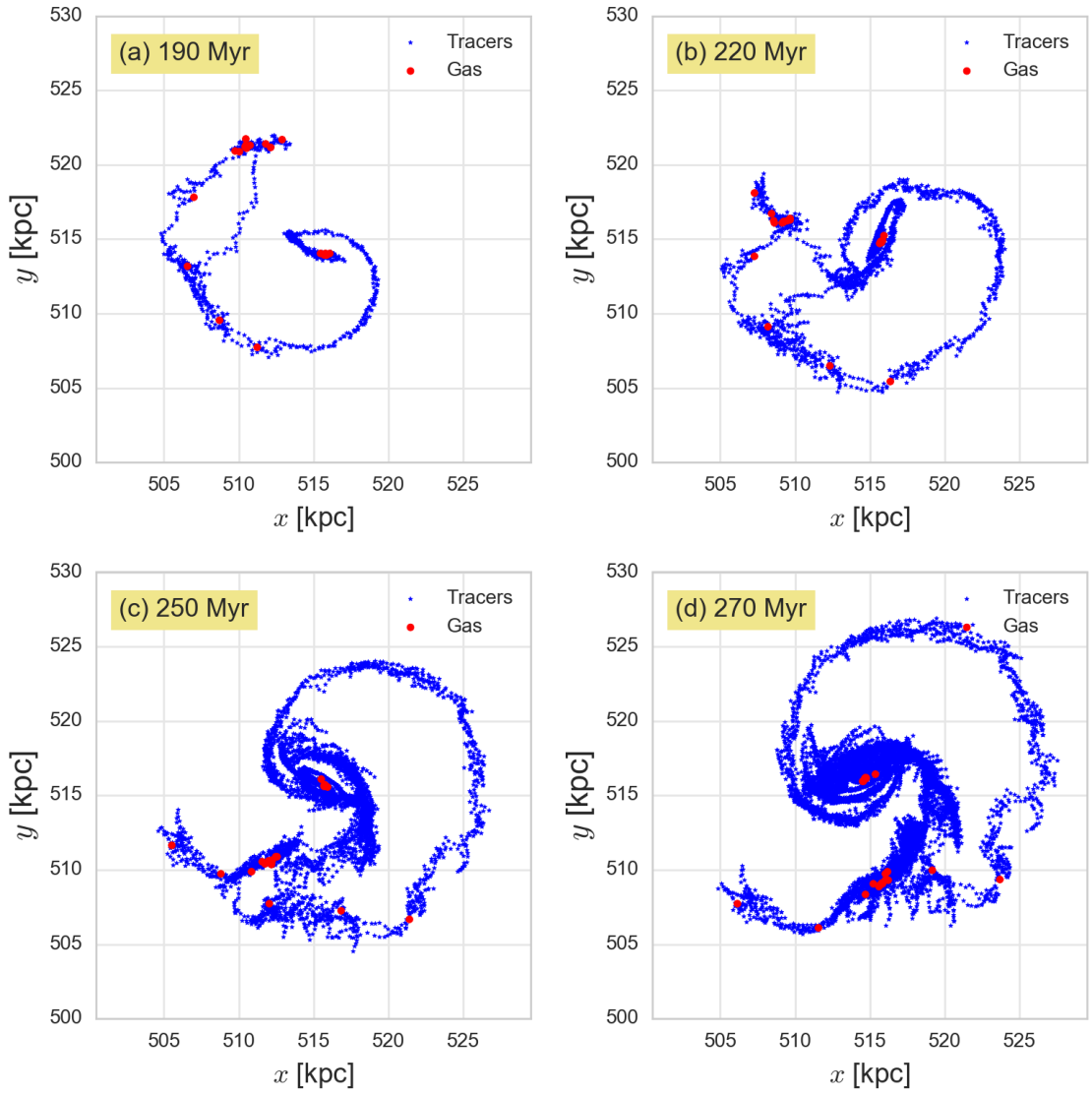

Figure 3.12: The contours of tracer (blue star) and gas (red circle) particles at (a) 190 Myr, (b) 220 Myr, (c) 250 Myr, and (d) 270 Myr for **Run 4102** in the x-y plane.

Figs. 3.11 and 3.12 show the contour evolution at four evolutionary snapshots in the x-y plane for Run 4101 and Run 4102 respectively. When we implemented a maximum temperature $e_{\max}$, the tracers were better behaved due to forcing particles to stay within a smaller temperature range. Forcing a maximum temperature has the impact that the particles may have difficulty decelerating around shocks, creating Type II errors, and later in the evolution (Run 4102, 270 Myr), we even still see that there are Type I errors occurring as tracers are being ejected from the contour. This set of simulations is akin to an isothermal simulation, as the temperature range for the gas and tracer particles was very limited.

3.8.2 Limiting the temperature growth based on fractional increase during each time step

In addition to testing whether the maximum or minimum allowed temperatures for gas and tracer particles affected the evolution of the contour, limiting the temperature growth based on a percentage increase of tracer temperatures during each timestep was also tested.

Table 3.7: Runs used to compare the evolution of the tracer particles when a limit in temperature increase was used. All other properties (tracer mass, timestep normalization, etc.) are the same for both Run 2063 and Run 2663, and can be found in previous tables such as Table 3.5.

Run ID	Notes
2063	control
2663	limit of 1% increase in temperature

In Table 3.7, the two runs being compared in this section are noted. Run 2063 here is used as the control run, as there are no limitations to how much temperature can increase

for any tracers between iterations. Run 2663 includes limitations in how much any tracer can heat up from one iteration to another. There is no limit on how much a tracer particle is allowed to cool, however. Each tracer particle is checked to be sure that the newly calculated temperature e is no greater than one per cent higher than its previous temperature. Thus, tracer particles can only increase in temperature by 1% during each iteration. Run 2663 was started from iteration 7350 of Run 2063, or at about 192 Myr into the evolution of Run 2063. At this evolutionary time, none of the gas or tracer particles in the disk of Run 2063 had over-heated, and thus it was reasonable to start the test from that point to save on simulation time and resources. The two runs were compared after 192 Myr by following individual tracer particles.

In Fig. 3.13, a single tracer that eventually overheated was followed over simulation time to see if limiting the temperature increase to 1 per cent at each iteration would be sufficient in preventing tracers from overheating. When including the 1% limit in temperature increase, it was clear that there was a difference in the continued evolution of the disk and that this limit prevented particles from becoming over-heated a small amount. The tracked tracer particle still become exceptionally heated in Run 2663, but it did not over-heat a second time as in Run 2063. However, it is unclear how much of a temperature limitation – if any – is physically reasonable and this heating limit could not be justified nor was physically-motivated.

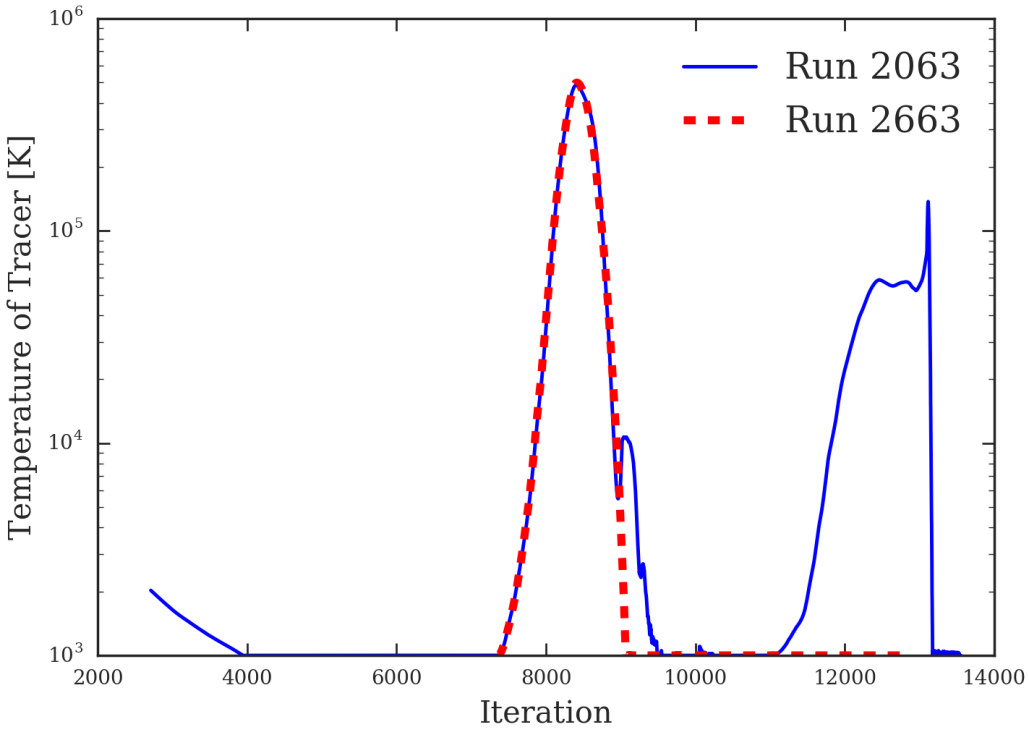

Figure 3.13: Temperature of a tracked tracer particle in Run 2063 (blue line) and Run 2663 (red dashed line) over iteration number.

3.8.3 Limiting the temperature growth based on density at particle position

While examining Run 2063, it was clear that particles first started to get over-heated while passing through a low-density region. If tracer particles experienced a shock front in a low-density region of the model, their temperatures often drastically increased, causing parts of the contour to be ejected. To prevent this from happening in these low-density environments, the temperature growth of the tracer particles were limited based on the surrounding density. If the density was lower than some target density $\rho_{\rm limit}$, then the temperature of a tracer in the region was not allowed to increase.

To choose the values of $\rho_{\rm low}$ and $\rho_{\rm high}$, the approximate minimum and maximum density regions in the disk of Run 2063, the visualization program Tipsy (N-Body Shop 2011) was used, seen in Fig. 3.14. The low-density $\rho_{\rm low}$ and high-density $\rho_{\rm high}$ regions are shown in the lime green and white rectangles respectively. In the figure, it is clear that the tracer particles encounter areas of extremely low density and relatively high density, so following how the density affects the temperature increase was important.

We used the density limits chosen in Fig. 3.14 to restart Run 2063 at 192 Myr, before any tracers were overheated and ejected from the disk. The restarted runs are shown in Table 3.8, where a density threshold $\rho_{\rm limit}$ was added to the tracer algorithm: if the tracers were in a region that was less dense than the given $\rho_{\rm limit}$, then the tracer particles were not allowed to increase their temperature during that given iteration.

In these runs, $\rho_{\rm high} = 10^{1.5}$ cm^{-3} and $\rho_{\rm low} = 10^{-2.5}$ cm^{-3}, as given by the high and low

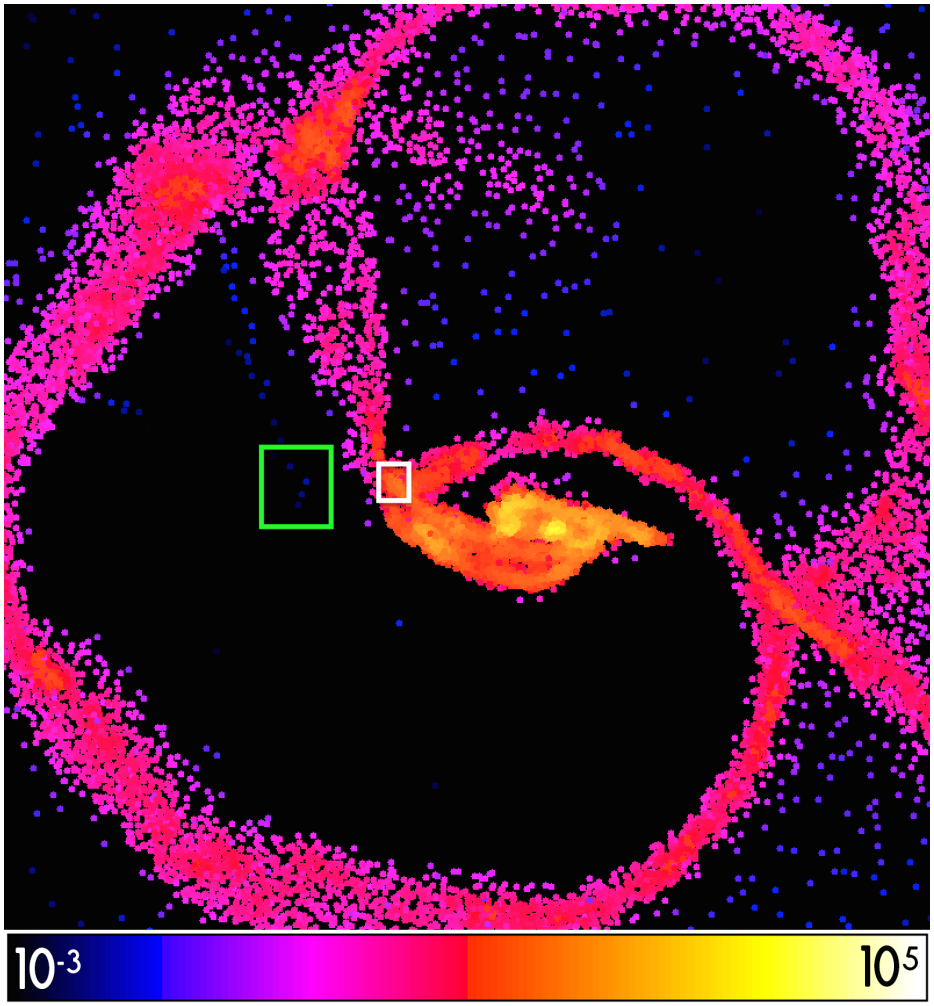

Figure 3.14: Tipsy (N-Body Shop 2011) visualization showing the disk of Run 2063 at 194 Myr into its evolution, with the low-density and high-density regions labeled by the lime green and white boxes respectively. The colorbar at the bottom shows the density of the medium in cm^{-3}. The length of a side of this box is 18 kpc.

Table 3.8: Run IDs with varying density thresholds for preventing tracer particles from heating. The Run IDs along with the threshold density $\rho_{\rm limit}$ are given. All other properties (tracer mass, timestep normalization, etc.) are the same for both Run 2063 and Run 2663, and can be found in previous tables such as Table 3.5.

Run ID	$\rho_{\rm limit}$
2063	None
2763	$\rho_{\rm low}$
2863	$\rho_{\rm low} \times 5$
2963	$\rho_{\rm low} \times 10$
2873	$\rho_{\rm low} \times 100$
2973	$\rho_{\rm low} \times 1000$
2173	$\rho_{\rm low} \times 2500$
2273	$\rho_{\rm low} \times 5000$
2373	$\rho_{\rm low} \times 7500$
2773	$\rho_{\rm high}$

density regions in iteration 7400 of simulation Run 2063, at 194 Myr before the shock front pushes tracers particles above the disk in Fig. 3.14. Note that $\rho_{\rm high} = \rho_{\rm low} \times 10^4$. While the time chosen is admittedly somewhat arbitrary, by this time the disk has evolved far enough to produce density increases and decreases significantly beyond the initial conditions.

Fig. 3.15 shows the temperatures versus evolutionary time for all runs from Table 3.8 in panels (a) through (j). The blue color indicates tracer particles, and the red color is for gas particles. The shaded region in each plot shows the range between the maximum and minimum temperature for all particles of a given particle type, and the solid line gives the mean at each evolutionary time. We found that the tracers were still heating up in all simulations except Run 2773, which effectively locks the temperature, but the temperatures of those over-heated tracers were not quite as extreme as without the heating limits based on temperature, a small net improvement. Except for the control, Run 2063, and the

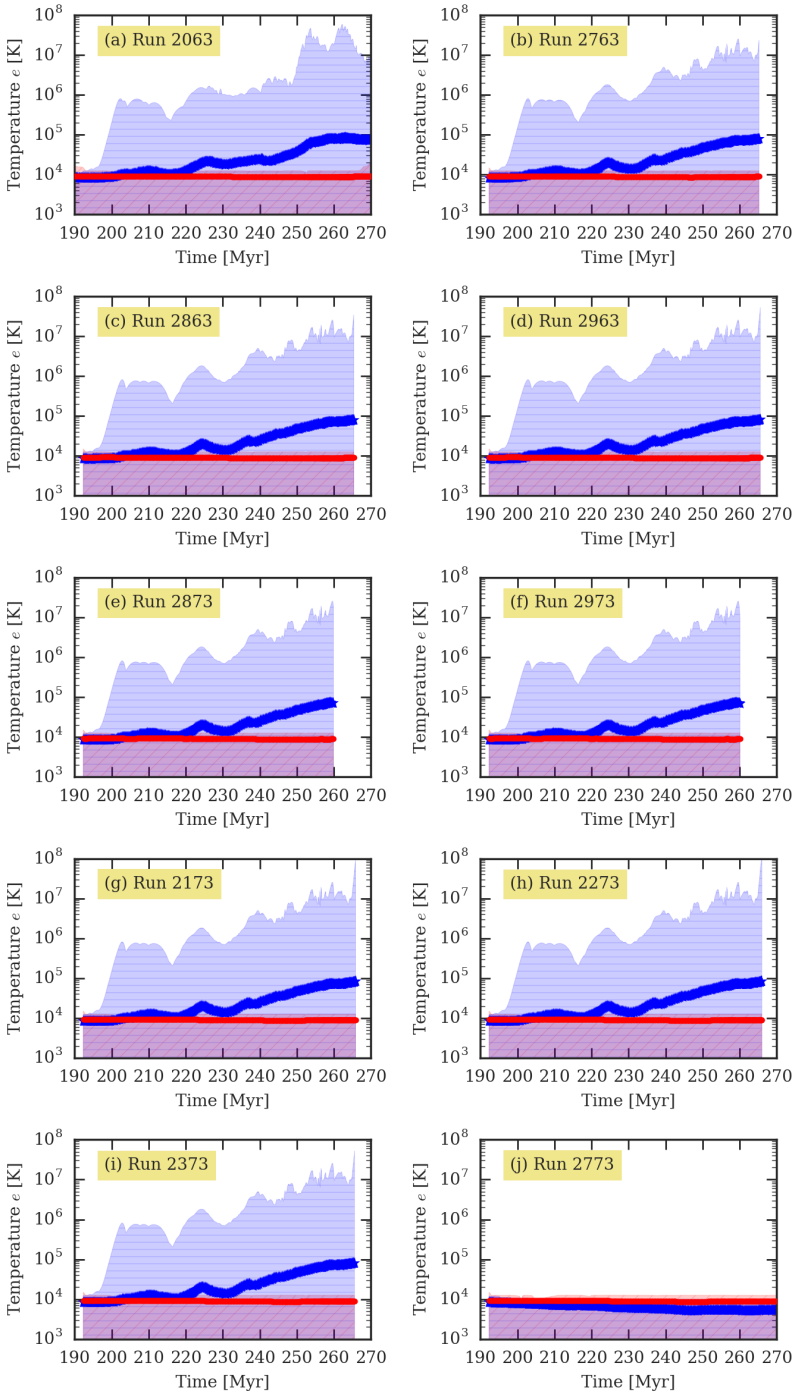

Figure 3.15: In panels (a) through (j), the temperature versus evolutionary time is shown for all tracer (blue) and gas (red) particles for the runs in Table 3.8. The shaded region shows the range between the minimum and maximum temperatures for any particle of that type, and the dark line shows the mean temperature at the given time.

simulation where $\rho_{\text{limit}} = \rho_{\text{high}}$, Run 2773, all of these temperature profiles are virtually indistinguishable from one another. Runs 2763, 2863, and 2963 in particular are nearly identical, and Runs 2873, 2973, 2173, 2273, and 2373 are nearly identical as well, though some peaks and valleys in the maximum tracer temperatures in these plots are *slightly* lower in temperature than Run 2963, but the difference is very small.

The density threshold method is also problematic in that it is tuned and relies heavily on this particular set of initial conditions. These density limits are specific to *this* model at the evolutionary time that the high and low density limits were chosen, and thus is not justifiable as a general solution if moving to other models which was our eventual goal. The density limits were indeed found to not be meaningful in other sets of initial conditions, such as in the isothermal collapse of a spherical shell found in Ch. 2.

However, all of the simulations with some ρ_{limit} do show maximum tracer temperatures that are less at nearly every evolutionary time than that of Run 2063, which does show that implementing some density-based temperature increase limitation could be worthwhile, but the specifics of what density is appropriate is uncertain. This observation suggests that the problem with the shock heating of tracer particles may have been a density gradient issue, as not enough particles may have been available around the shock region for proper calculations.

3.9 Effect of changing the timestep normalization of the simulation

We test whether changing the timestep normalization `dtnorm` has any effect on the results of the disk simulations. The timestep normalization is a parameter that determines how long the timestep is based on the normal timestep criterion used. Previous experience of working with the HYDRA code suggest that setting `dtnorm`=0.5 is necessary in simulations with cooling. It is worth emphasizing that in simulations with cooling, timestepping is non-trivial because cooling curves versus temperature can be very steep. Consequently, the cooling in HYDRA is implemented in a semi-implicit form that does not limit the timestep resolution (Couchman et al. 1995, 1996).

In Fields (2017), the effect of changing `dtnorm` was not considered and we found little dependence on the timestep normalization in the spherical collapse analyzed in Chapter 2. Some of the disk simulations reported in Fields (2017), however, were restarted midway through their evolution with a smaller `dtnorm` to attempt to prevent particles from being ejected from the disk.

Table 3.9 shows three runs used to determine the effect of changing the timestep normalization. All runs listed here use the same initial conditions and the same run parameters other than `dtnorm`.

Fig. 3.16 shows the differences in λ between the three runs due to `dtnorm` alone. A fit is shown with the data in each set at 200 Myr and above. Additionally, the component of contour length increase due to shearing has been removed from each of the data sets,

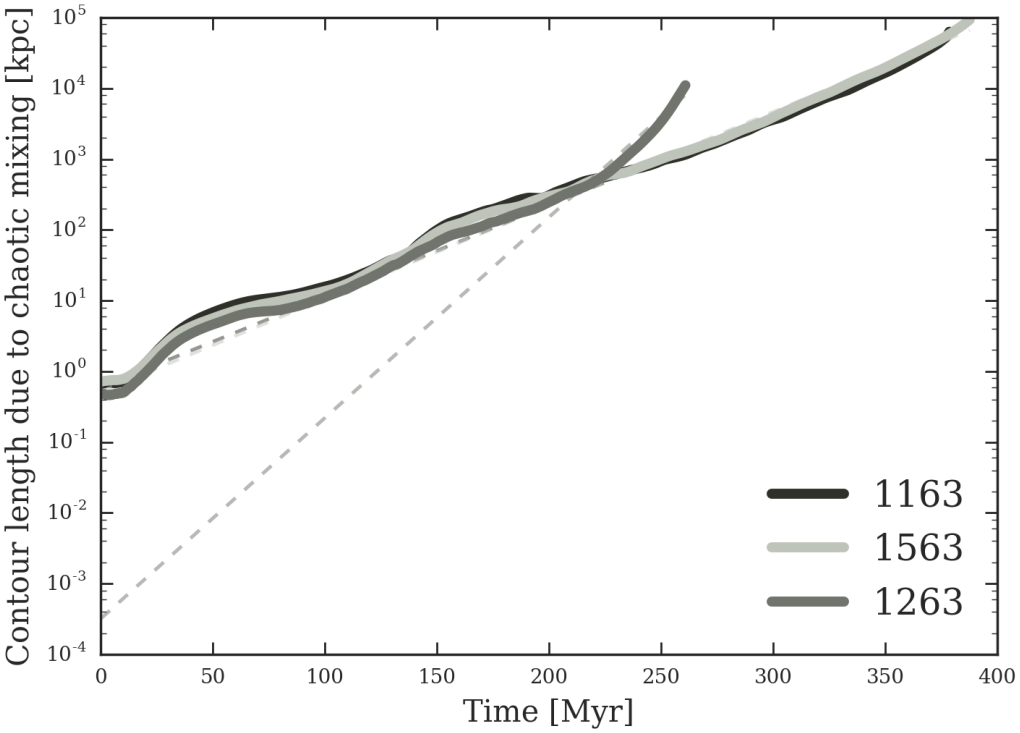

Figure 3.16: The change in length of the contours of Run 1163, Run 1563, and Run 1263 due to the chaotic mixing in the disk alone are shown over evolutionary time. The fits to the data (dashed lines) are seen at 200 Myr and above, and the Lyapunov exponents λ along with the doubling time t_d are given in Table 3.10 with their associated 2σ uncertainties.

Table 3.9: Run IDs to determine the effects of changing the timestep normalization. Also given in the remaining columns are the tracer q_{ij} (§3.4), method of velocity and acceleration calculation (§3.6, 3.7), the mass of the tracers ($m_{\rm tracer}$ §3.5), and minimum and maximum allowed temperatures for gas and tracer particles (§3.8).

Run ID	`dtnorm`	Tracer q_{ij}	Velocity Calculation	Acceleration Calculation	$m_{\rm tracer}$ [$M_\odot$]	$e_{\rm min}$ [K]	$e_{\rm max}$ [K]
1163	1.0	1.0	`velsmooth`	pairwise	100	1000	None
1563	0.5	1.0	`velsmooth`	pairwise	100	1000	None
1263	0.25	1.0	`velsmooth`	pairwise	100	1000	None

therefore the contour length change is due only to the amount of chaotic mixing in the disk. The amount of shearing is indeed part of the evolution of the disk, but in order to systematically account for the evolution that would happen regardless of the local mixing, we removed it from our datasets. Here, there is only a small difference between Run 1163 and Run 1563, and they are not within the estimated uncertainties of each other. However, the main difference is seen between Run 1263 and either other run. Table 3.10 shows the λ and doubling time t_d for the data presented in Fig. 3.16. Using λ, we can calculate the doubling time t_d for each of the models, where

$$t_d = \ln 2/\lambda \tag{3.3}$$

and the uncertainty in t_d is found using Gaussian uncertainty propagation,

$$\sigma_{t_d} = (\ln 2/\lambda^2)\sigma_\lambda \tag{3.4}$$

where σ_λ is the uncertainty in λ. This figure shows that the timestep normalization can

indeed significantly affect the overall amount of chaotic mixing in the disk, and the results shown in Fields (2017) about the relationship between Q and λ were likely affected, as seen in the table.

Table 3.10: The Run IDs and their λ found from the fit in Fig. 3.16, along with the calculated doubling time t_d for each and their associated 2σ uncertainties.

Run ID	λ [$\times 10^{-4}$]	t_d [Myr]
1163	128.67 $\pm$ 0.46	53.87 $\pm$ 0.19
1563	131.84 $\pm$ 0.28	52.57 $\pm$ 0.11
1263	283.78 $\pm$ 1.6	24.42 $\pm$ 0.14

The table clearly shows a large difference between Run 1263 where `dtnorm`=0.25 and the other two runs. Thus, shortening each timestep does not necessarily mean more well-behaved results, as the shorter timestep caused additional chaotic mixing to be present, due to the tracers in the contour experiencing Type I errors. This is not as counterintuitive as it may sound. For example, if particles entering into unstable regions is the cause of the errors, then a shorter timestep makes that more likely to occur.

3.10 Difficulty in identifying when particles are ejected from the disk

The major problem with the tracer particles was that as a strong density contrast was encountered, such as in regions of a shock, that the tracers would be extremely over-heated and thus be ejected from the disk of the galaxy, in Type I errors. These errors made the tracers not useful for the type of investigations we wanted to do, as this movement above

the disk did not accurately represent the type of mixing that was happening with the gas particles. The only way to determine if this was happening was to visually inspect the particles of the disk, such as in snapshots of the whole disk or of the contour itself as it evolved.

Using temperature tracking alone was insufficient as numerous particles heat up but cool down rapidly before any ejection events. Further, it was not appropriate to track velocities to see if the tracer velocity was larger than the escape velocity $v_{\rm esc}$ because the difference between the circular velocity and the escape velocity is small – $v_{\rm esc} = \sqrt{2}v_{\rm circ}$, and the circular velocity was already fairly high at 180 km/s. We considered a global method that tried to determine if the tracers had stopped following the flow by looking at the ratio of the change in the simulation time between two iterations, $\mathrm{d}t_x/\mathrm{d}t_{x-1}$ where $\mathrm{d}t$ is the change in evolutionary time from one timestep to another, and the x and $x-1$ denote the current and previous timestep. The C-F-L condition triggers a shorter timestep when velocity is large, thus the goal here would be that any sudden change in $\mathrm{d}t_x/\mathrm{d}t_{x-1}$ would be clearly visible in a graph as a sharp spike, when the timestep had to drastically decrease due to particles being overheated and thus ejected.

This concept can be seen in Fig. 3.17, where $\mathrm{d}t_x/\mathrm{d}t_{x-1}$ is plotted against simulation time for Runs 2063 and 9363, as described in §3.7. In this figure, however, it is not possible to discern at what time either simulation failed to properly continue tracing the flow. There are many small but sharp spikes throughout the simulation time, and it is not clear which (if any) are indicative of tracers encountering errors. In the blue line for Run 2063, it is quite clear that there is a very drastic peak at the end of the simulation, around 270 Myr,

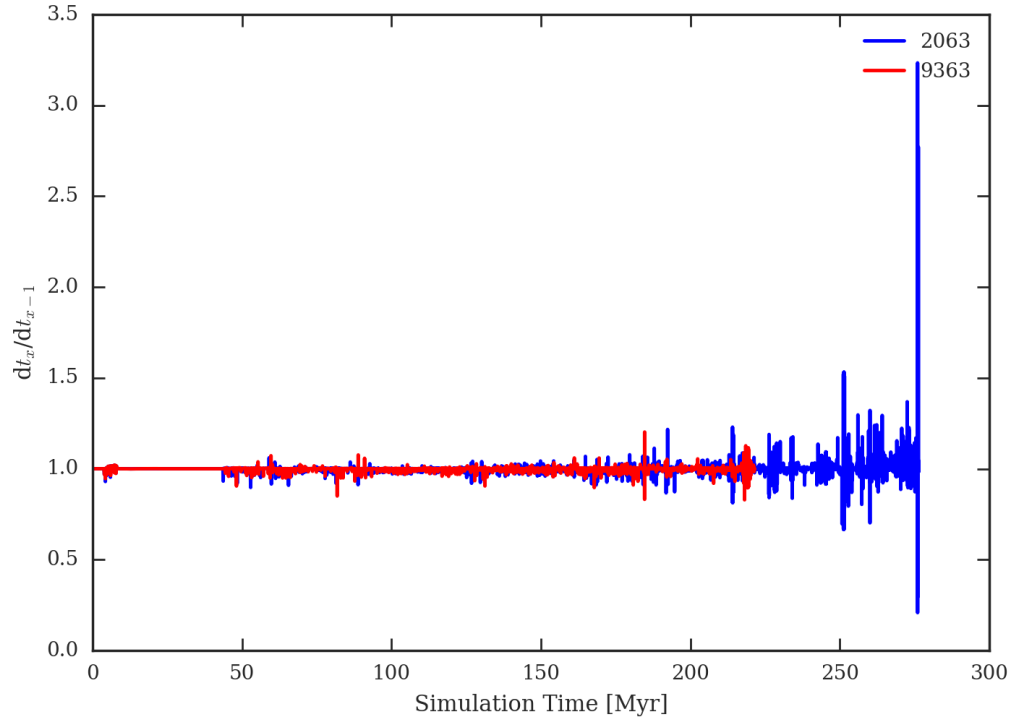

Figure 3.17: Change in the timestep difference per iteration versus simulation time in Myr for Run 2063 (blue line) and Run 9363 (red line). The evolutionary time at which tracers in the contour no longer follow the flow is not clear from this figure.

but these data do not correlate precisely with respect to when the contour begins to fail.

When visually inspecting a movie of the contour over time for Run 9363, whose snapshots are seen in Fig. 3.18, it becomes clear that the contour tracing method begins to fail starting at approximately 150 Myr, something that is not seen in Fig. 3.17. By 160 Myr, particles are catastrophically being ejected as seen in the x-z plane of Run 9363, a feature that is *not* seen in Fig. 3.17.

The same analysis can be done for Run 2063, as seen in Fig. 3.7. By 215 Myr of Run 2063, it is clear in the x-z plane that some tracers are starting to travel above the disk, as

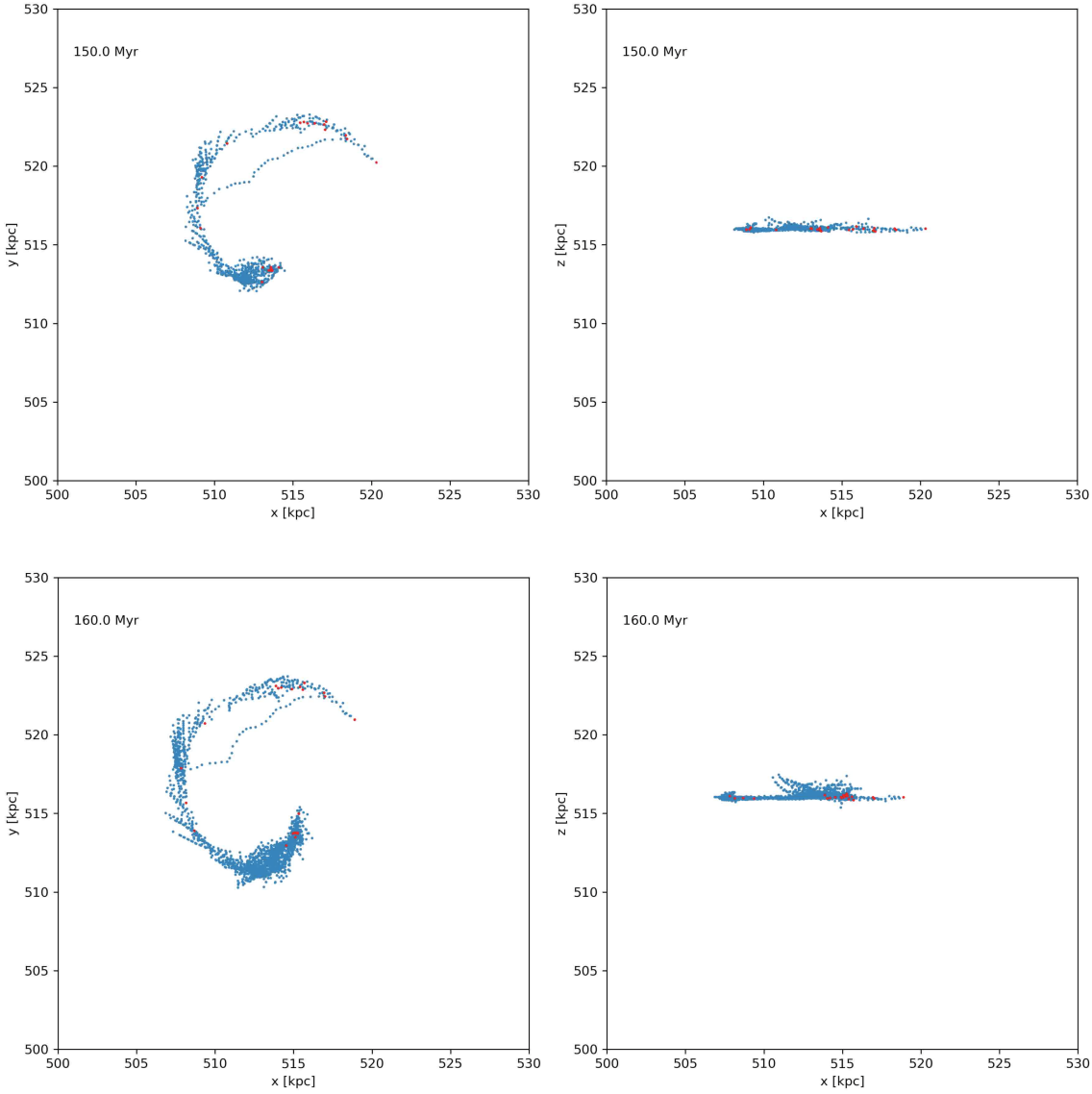

Figure 3.18: Evolutionary snapshots of the tracked contour in Run 9363 at 150 Myr (top row) and 160 Myr (bottom row) with the x-y plane shown on the left panels and x-z planes shown on the right. The original gas particles in the contour are shown in red dots, while the blue dots show the tracer particle positions. Approximately 150 Myr is when the tracers first start to be ejected from the disk (visible in the x-z plane) and by 160 Myr, it is much more evident that the tracers are being ejected and not accurately following the flow of the gas particles.

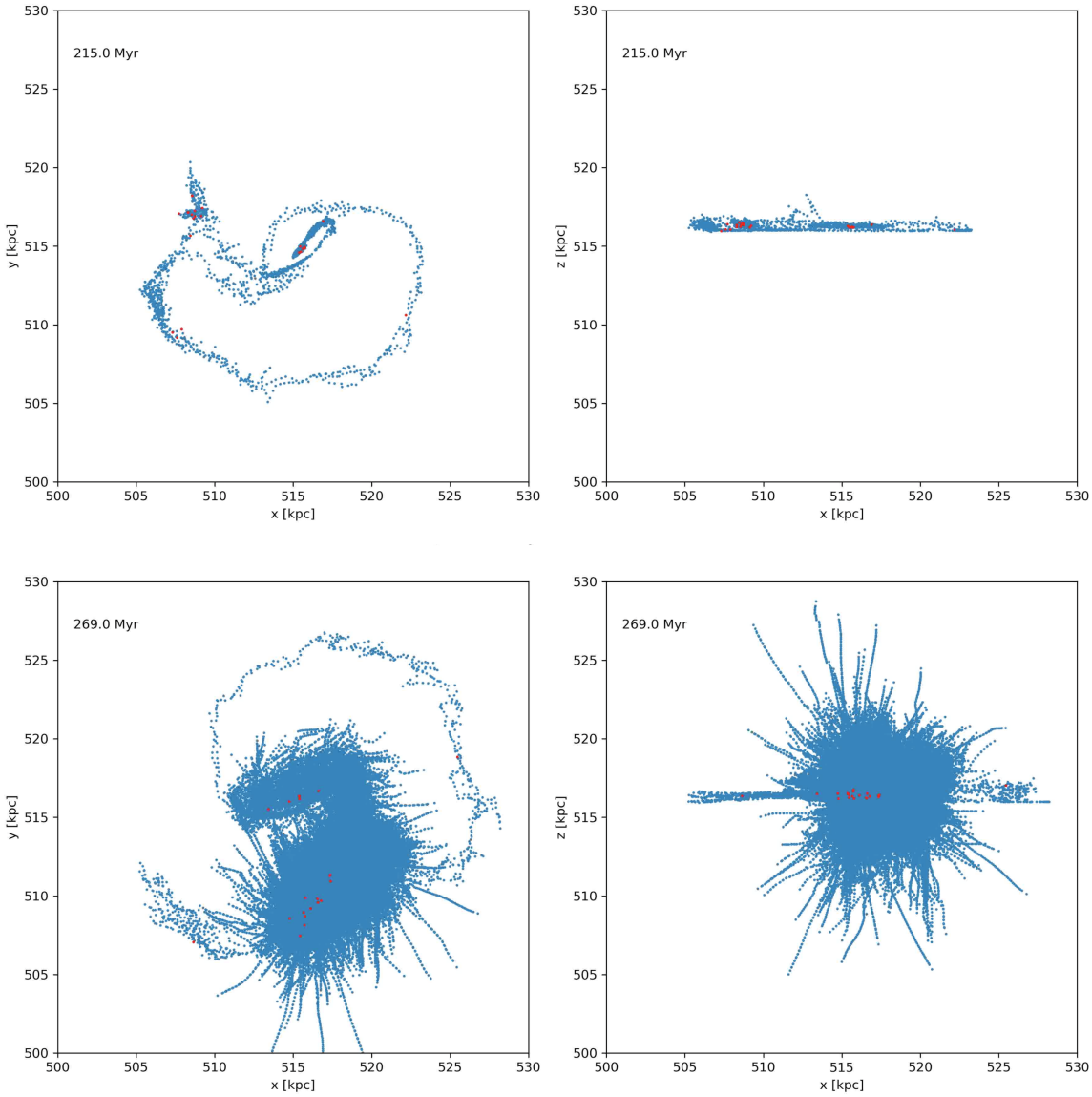

Figure 3.19: Evolutionary snapshots of the tracked contour in Run 2063 at 215 Myr (top row) and 269 Myr (bottom row) in the x-y (left) and x-z (right) planes. The original gas particles in the contour are shown ih red dots, while the blue dots show the tracer particle positions. Approximately 215 Myr is when the tracers first start to be ejected from the disk (visible in the x-z plane) and by 269 Myr, the tracers are being catastrophically ejected in all directions.

seen between x=510 and x=515 kpc. There is an associated spike in Fig. 3.17 around 215 Myr in 2063, but just by visual inspection, it does not seem much more significant than the sharp spike that occurs just before 200 Myr. By the contours of Run 2063 in Fig. 3.19 at 269 Myr, it is clear that the evolution of the tracers in the contour has catastrophically failed, something not seen if the contour is not visually inspected.

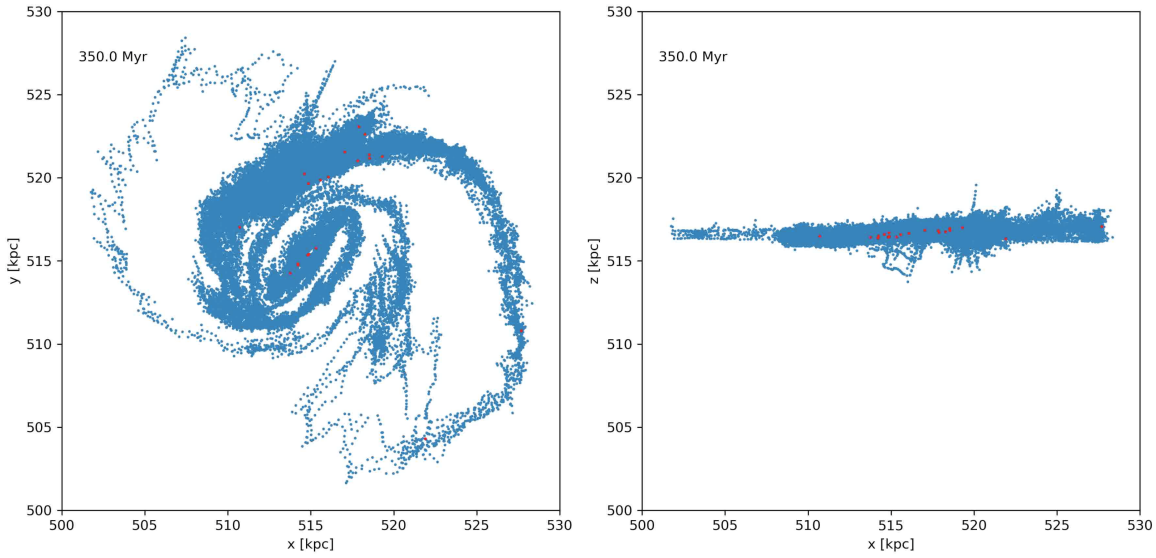

Figure 3.20: A snapshot of Run 1563 (§3.9) is shown at 350 Myr into its evolution in the x-y plane (left) and x-z plane (right). The tracers are blue points, and the gas particles are red points.

Additionally, the change in length of the contour over time is not a reliable indicator of tracer flow either. In §3.9, Fig. 3.16, the change in length of the contour due to chaotic mixing in simulations with varying `dtnorm` was examined. If the data in that figure for Run 1563 is inspected, there is no obvious point at which we could determine for certain that the tracers start to experience either Type I or Type II errors. However, when visually inspecting the contour shown in Fig. 3.20, it is obvious that at least by 350 Myr, there are

significant failures in the tracer method. Thus, even the contour length change over time does not provide enough information to determine if tracer errors are occuring.

3.11 Summary

This chapter described many of the tests that were done with our tracer method and how the changes in our simulation, such as the smoothing algorithms for velocities and accelerations and limiting the temperature growth based on percentage increase or density, affected the evolution of the tracer particles.

While we had one simulation that ran through to 285 Myr without overheated tracers (Run 2163), this result was unusual and not typical of the behaviours observed. Even though no tracers being ejected from the disk, Run 2163 still encountered Type II errors where the tracers flowed through the shock without interacting with it.

Despite attempts to fix the issue, the two fold problem remains: if tracers shock heat too much they get ejected, if they are not shocked enough then flow interpenetration occurs. Both of these results are problematic although the second allows integration to proceed.

Given the problems noted, in the next chapter we further examine tracer behaviour on a largely individual basis, and place this in context of known shortcomings of SPH.

Chapter 4

Analysis of tracer behavior and identifying unavoidable limitations

In the previous chapter, we reviewed a number of approaches to improve the isothermal approach taken in Fields (2017) to keep tracer evolution stable at all times. In that work, there was no consideration of the effects of changing the timestep normalization during the evolution of a disk, nor was the artificial viscosity or the use of a velocity smoothing algorithm considered for systematic testing. However, as seen in the previous chapter, all of these algorithmic changes to the simulations produced varying results of success for the evolution of a disk with tracers. In this chapter, we delve more deeply into the tracer evolution problems by examining individual tracer behaviour and discussing some fundamental limitations that the tracer approach has in SPH. The exploration conducted in this chapter is motivated by additional aspects of SPH which we briefly outline.

In SPH, there is a positional instability between hot and cold gas phases with large density contrasts. This is manifest most clearly above and below the galactic disk of particles, as Agertz et al. (2007) have shown. Particles slightly above the dense disk have an asymmetric number of neighboring particles around them with more particles below, and experience a force that pushes the particles further outward. Consequently, a gap is opened between a dense disk and particles existing above the disk, as can be seen in Fig. 4.1. This may contribute to our tracer particles being ejected above and below the disk.

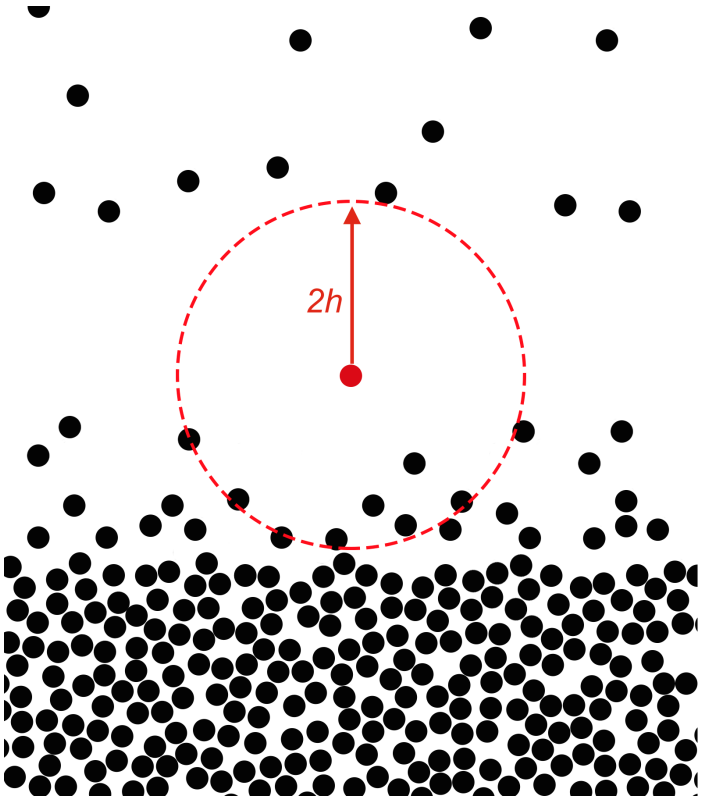

Figure 4.1: Illustration showing the "gap" between high- and low-density regions described in Agertz et al. (2007). The high density region of a disk can be seen at the bottom of the figure, and the low-density region above the disk is seen at the top of the figure. In the middle, there is a particle that is above the high-density region but not quite in the low-density region, and thus it is in the "gap."

It is also very difficult to implement an accurate "shock tracer" in SPH (Morris & Monaghan 1997; Wadsley et al. 2017). The key concern with tracer evolution is that the particle distribution cannot adjust to their presence the same way regular particles will. Hence, tracer particles may find themselves in regions of instability without any adjustments to the regular particle distribution resulting. The reason for this is that SPH particle distribution adjusts on a pairwise basis to each other position. These SPH particles will generally "even-out" in position, but since the tracers do not impact the mass distribution, they cannot do the same and their forces could be more noisy. Taken together, these two aspects of SPH evolution have the potential to make tracers behave in more complicated ways than the actual simulation particles.

In this chapter, we first verify that only the tracer particles are being over-heated. Then, we look at the individual behaviours of over-heated tracer particles and compare those properties to the gas particles in the same simulations.

4.1 Confirming tracer over-heating

Since particle ejection is one of the known ways in which SPH simulations can go wrong, we first confirmed particle ejection was limited solely to the tracers. To verify that tracers were being ejected, I used the Tipsy visualization program (N-Body Shop 2011) to mark the tracers by mass to ensure that there were no gas particles being over-heated or ejected from the disk of the galaxy. The models shown in Table 4.1 were checked, and the table includes the relevant sections which the particular simulation was analyzed in this work. We

could have also used a positional search on the particles to check whether they were above or below a certain height.

Table 4.1: Run IDs and their associated iteration number and evolutionary time checked to ensure that only tracers and not regular gas particles were being overheated. The section in which these runs appear in the text is given in the final column, if applicable.

Run ID	Iteration No.	Time [Myr]	Relevant Sections
2063	9990	235.25	§3.6, 3.7, 3.8.2
2973	8500	210.81	§3.8.3
2873	8460	210.26	§3.8.3
7163	5410	365.29	
3063	9630	349.77	
1163	5570	378.58	§3.5, 3.9
9563	9570	379.49	
4101	14390	277.45	§4.2.1

Fig. 4.2 shows the gas (red) and tracer (blue) particle temperatures over time for Run 2063, one of the simulations given in Table 4.1. In the figure, the minimum to maximum temperature range for all gas particles at each iteration are shown in the red shaded region with the diagonal hatches. The line of red circles gives the average gas particle temperature at each iteration. In the blue shaded region with the horizontal hatches, the minimum to maximum temperature range for all tracer particles is shown, and the line of blue stars gives the average tracer particle temperature at each iteration. Here, it is evident that the gas particles (in red) never exceed 10^5 K, whereas the average and maximum temperatures of the tracers exceed this substantially. The maximum gas particle temperature increases to 10^5 K near the end of the available data, but that is likely due to gas particles becoming bunched around shock fronts. This figure shows that for Run 2063, it is *only* the tracer particles that

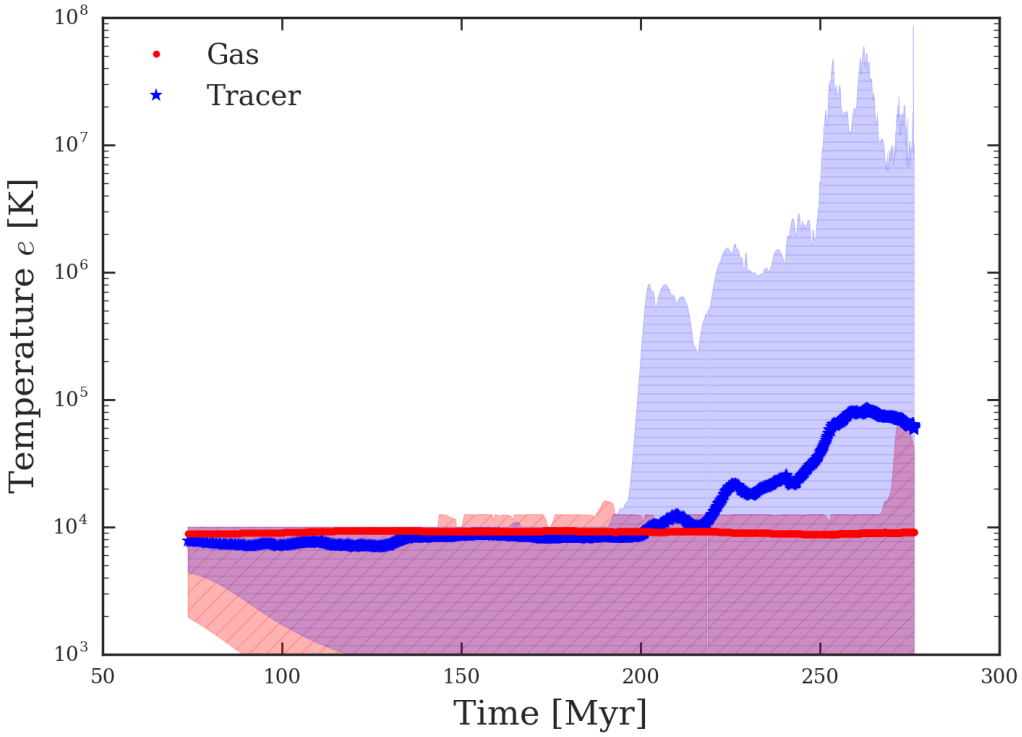

Figure 4.2: The minimum, maximum, and average temperatures for the gas (red) and tracer (blue) particles in Run 2063. The points indicate the average temperatures for all particles of the given type at each iteration, and the shaded regions show the maximum and minimum temperature ranges for all particles of a given type at each iteration.

overheat, subsequently get ejected from the disk, and cause our measurements of the total length change of the contour to be unusable.

Checking that only the tracers were affected by overheating was important to better understand the way in which our contour method was inaccurate. Therefore, determining that only the tracer particles overheated shows that it was the tracer particles that were the source of the problems in our method.

4.2 Following individual tracers throughout the disk evolution

Tracers were tracked throughout the simulation code to determine which hydrodynamic properties were causing them to be ejected. Initially, we did not know if the temperature, density, velocity, artificial viscosity, smoothing length, acceleration, or any other specific property was the cause, so a number of properties were tracked and printed during every timestep for individual tracer and gas particles.

4.2.1 Behaviour of overheated tracers

A majority of the tracers that were heated were able to cool rapidly. However, for a small number of tracers, it was the excessive heating that caused the tracers to pick up a high velocity that pushed them out of the disk. Once these tracers had a velocity pushing them in the z-direction, any cooling was too late to bring the particles back down into the disk. As the tracers rose out of the galactic disk, their density fell rapidly, but the cooling time

grew longer as the density lowered. The adiabatic cooling that results for an expanding system is not sufficient for the particles to fall back quickly either. These particles cooled adiabatically as they left the disk region. It must be emphasized, surprisingly, that out of the thousands of tracer particles that were added to the contours in the simulations, it was typically only a few dozen tracers that became substantially over-heated. Having so few tracers overheat meant that a majority of the tracers followed the flow of the gas particles well, as long as they were in areas of the disk with moderate density gradients.

To understand how the tracer population evolved differently than the gas population, we next focused on comparing the two populations. However, comparing local gas to tracer properties is not always a well-defined procedure since the most nearby gas particle to a particular tracer can change. Before doing that, we first analyze the comparison between a single tracer and the average population of gas particles in the contour, so we followed a well-behaved tracer to investigate the variations in behaviour. The tracer particle chosen in this section is from **Run 4101**, previously mentioned in §3.8.2. The simulation has a temperature floor of $e_{\rm min} = 1000$ K, and was restarted from Run 2063 (described in §3.6, 3.7, 3.8.2) at 195 Myr. Initially, this tracer was tracked in the acceleration calculations to attempt to better understand how the tracers as a whole may have been given an uncharacteristically high velocity, but as mentioned, this particular tracer did *not* overheat or get ejected from the disk. The properties of this tracer from 195 Myr onward are given in Fig. 4.3, marked by the line of blue stars. Also included in the figure are the properties of all 23 gas particles in the contour, where the average of this small population of gas particles is given by the line of the red circles, and the red shaded region represents the range of minimum to maximum

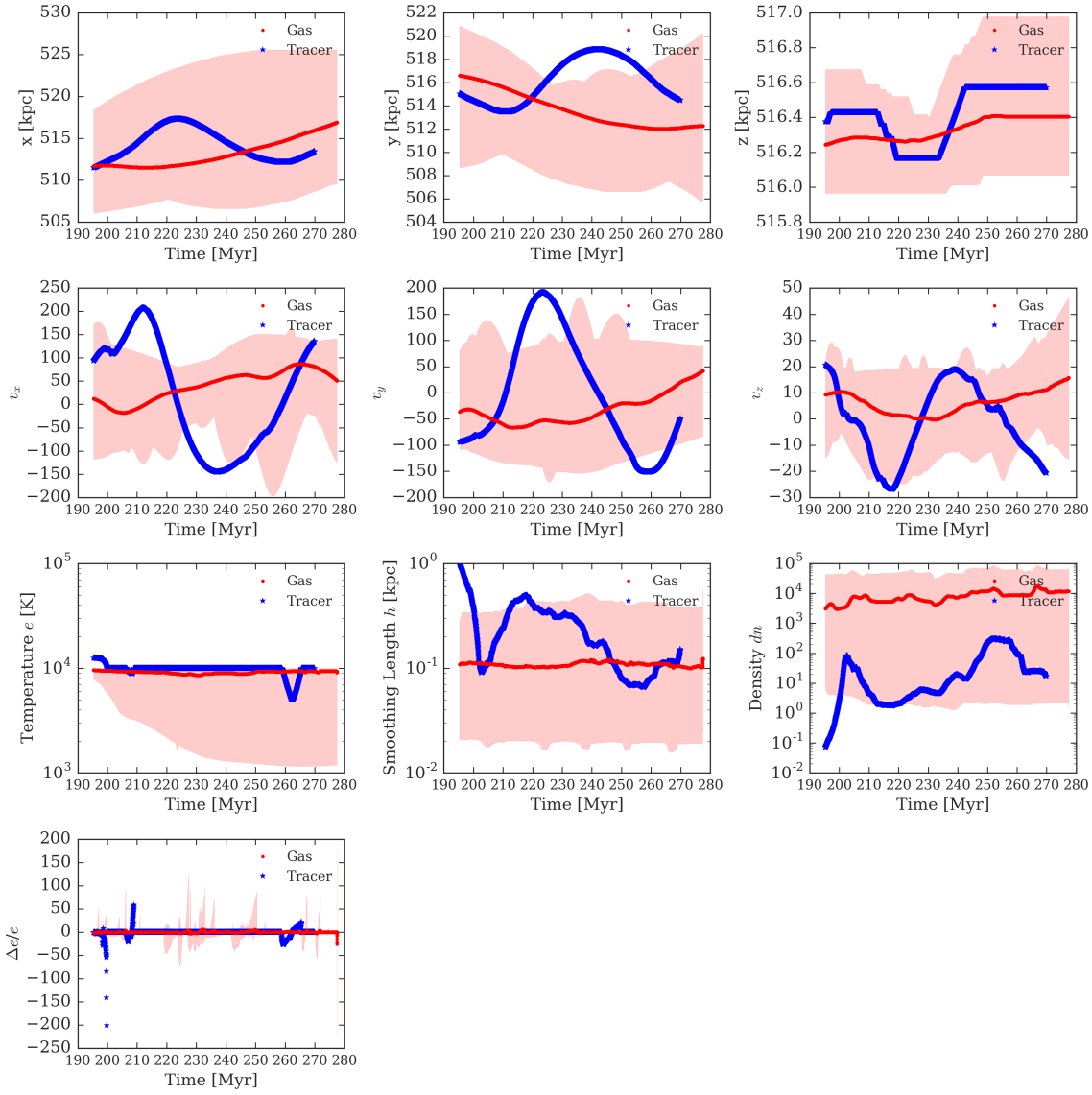

Figure 4.3: Properties of tracer number 148,565 from Run 4101 over simulation time and the associated properties of the 23 gas particles in the contour. The line of blue stars shows the properties of the tracer particle and the line of red circles shows the average value of the gas particles in the contour at each simulation time. The shaded red regions give the minimum to maximum range in property values from all of the gas particles at that given time. By examining z-position and temperature e, it is clear that this particular tracer was *not* one that was ejected from the disk.

values for a given property for all of the gas particles in the contour at that given simulation time. It is important to emphasize that the red line does not give the properties of any single gas particle, but rather the average of the 23 particles in the contour, and further that this sample of gas particles is extremely small compared to the entirety of the disk, as there are 40,000 total gas particles included in the simulation.

In the figure, the properties given over simulation time are positions, velocities, temperature e in K, smoothing length h in kpc, and density dn. In the bottom row of Fig. 4.3, the change in temperature at each timestep divided by the current temperature of the particle, $\Delta e/e$ is given over evolutionary time. Here, when examining the temperature e of the tracer over time, the tracer never heated to much more than 1.2×10^4 K, therefore it stayed within a reasonable temperature range from 195 Myr to approximately 213 Myr, which is the evolutionary period when other tracers overheated and were ejected. When also looking at the z-position over time of the tracer, there is very little change, showing that this particular tracer particle was *not* one that was expelled from the disk. We can see that the tracer never exceeded the maximum z-position of any gas particle in the contour at any iteration. Overall, these results show the expected level of variance between well-behaved tracers and gas particles. The only notable issue might be the lower density of the tracer – but this is more than possible given the large density contrasts in the disk and the small sample of gas particles (23 examined of the 40,000 total in the disk) relative to how many tracers there are (from about 1000 tracers at 195 Myr to more than 380,000 total tracers by nearly 280 Myr).

To find a tracer that *did* overheat, data from the same Run 4101 were used to find all

tracers that exceeded 15,000 K anytime before 260 Myr. In Run 4101, there were only thirty of these such tracers. A value of 15,000 K was a reasonable temperature to consider because radiative cooling typically takes over very quickly and creates a barrier at 10^4 K which particles are quickly cooled to. If a particle is reaching 1.5×10^4 K, then it has been significantly overheated and has much less of chance of cooling down again before its velocity feels a kick out of the disk. Note that out of thousands of tracers that were added to the contour, having only *thirty* overheat meant that in most cases, the tracers followed the flow of the gas particles well, as long as they were not added above the disk to account for the stretch due to other tracers being ejected. However, since the contour that we wanted to track was sensitive to any minor extensions in length, even if there was only one tracer that was ejected, it would cause the entire length to be inaccurate.

Fig. 4.4 focuses on a single particle that was substantially over-heated and then ejected from the disk of the galaxy in Run 4101. Similar to Fig. 4.3, the properties of the tracer particle are given over time. Also shown in the figure are the properties of the 23 gas particles in the contour in this run – the shaded region gives the range between the minimum and maximum values for a given property, and the red line gives the average value of the property over time for all of the gas particles – just as in the previous figure, this line does not correspond to any individual gas particle. In Fig. 4.4, the first thing to notice is the extreme temperature increase for the tracer between 200 and 210 Myr. This significant temperature increase leads to lagged but rapid changes in v_x, v_y, and v_z, and causes the tracer particle to start increasing in the z-direction more rapidly, thus showing the particle being ejected from the disk as it moves higher than any of the other gas particles in the

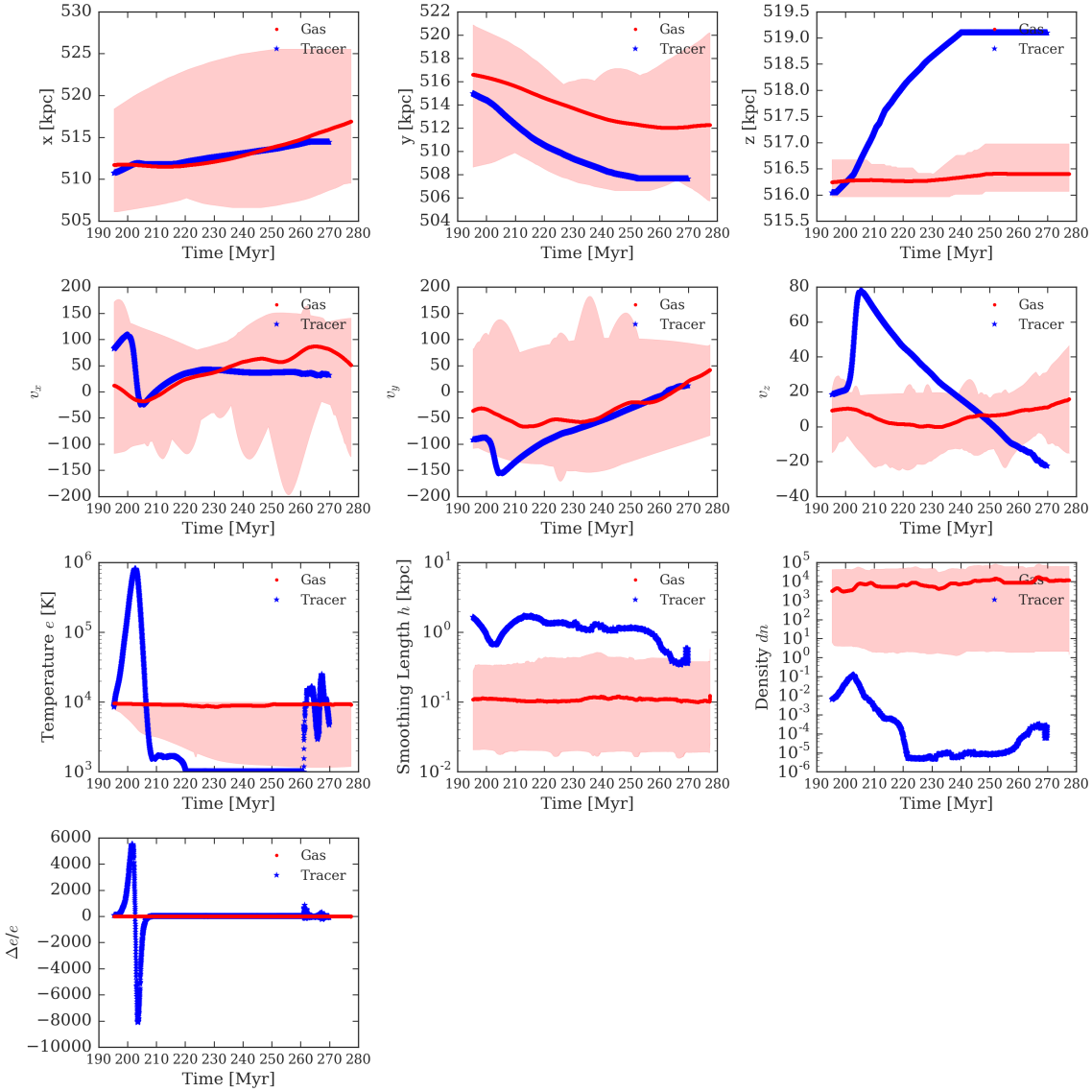

Figure 4.4: Properties of tracer number 148,863 and the 23 gas particles in the contour from Run 4101 over simulation time. The line of blue stars shows the properties of the tracer particle and the line of red circles shows the average value of the gas particles in the contour at each simulation time. The shaded red regions give the minimum to maximum range in values from the gas particles at that given time. Here, when examining z-position and temperature e, it is clear that this tracer overheats and is ejected above the disk of the galaxy.

contour. During the rapid temperature increase, the smoothing length h also significantly and rapidly decreases. Here, it may be that the particle is reaching an area of higher density than it was previously experiencing, causing the smoothing algorithm to require a shorter smoothing length to keep a consistent number of neighbors. The density plot in the bottom right of the figure confirms this hypothesis, as the density dn around the tracer particle spikes at nearly the same evolutionary time as the temperature e spikes, and the tracer is experiencing a much lower density than the gas particles in the contour. Noticed in the temperature e plot in Fig. 4.4, however, is that the temperature rapidly declines after about 10 Myr and the overheated tracer particle returns to a temperature similar to the disk particles. This cooling comes too late, though, as the boosted v_z cannot prevent the tracer from being expelled from the disk, seen in the z-position over time.

To understand on a particle by particle comparison, we looked at the force calculation properties of the hot tracer from Fig. 4.4 compared to a single gas particle in the disk. These two particles at approximately 190 Myr are very close to each other, but they separate over time. This comparison can be seen in Fig. 4.5. Here, the positions, scaled refinement velocities, and the scaled refinement accelerations of the two particles are shown over simulation time, in addition to the number of neighbors, the scaled pairwise force, and the scaled artificial viscosity felt by the two particles. There are extreme changes in the scaled refinement acceleration of the tracer between 200–210 Myr, which correlates to the extreme change in temperature during that time for this tracer particle. That same timeframe corresponds to a huge dip in the scaled artificial viscosity of the tracer as well. Even the velocities of the tracer in this figure show a significant change between 200 and 210 Myr, further showing

that the properties of the tracer are being affected before it is ejected from the disk.

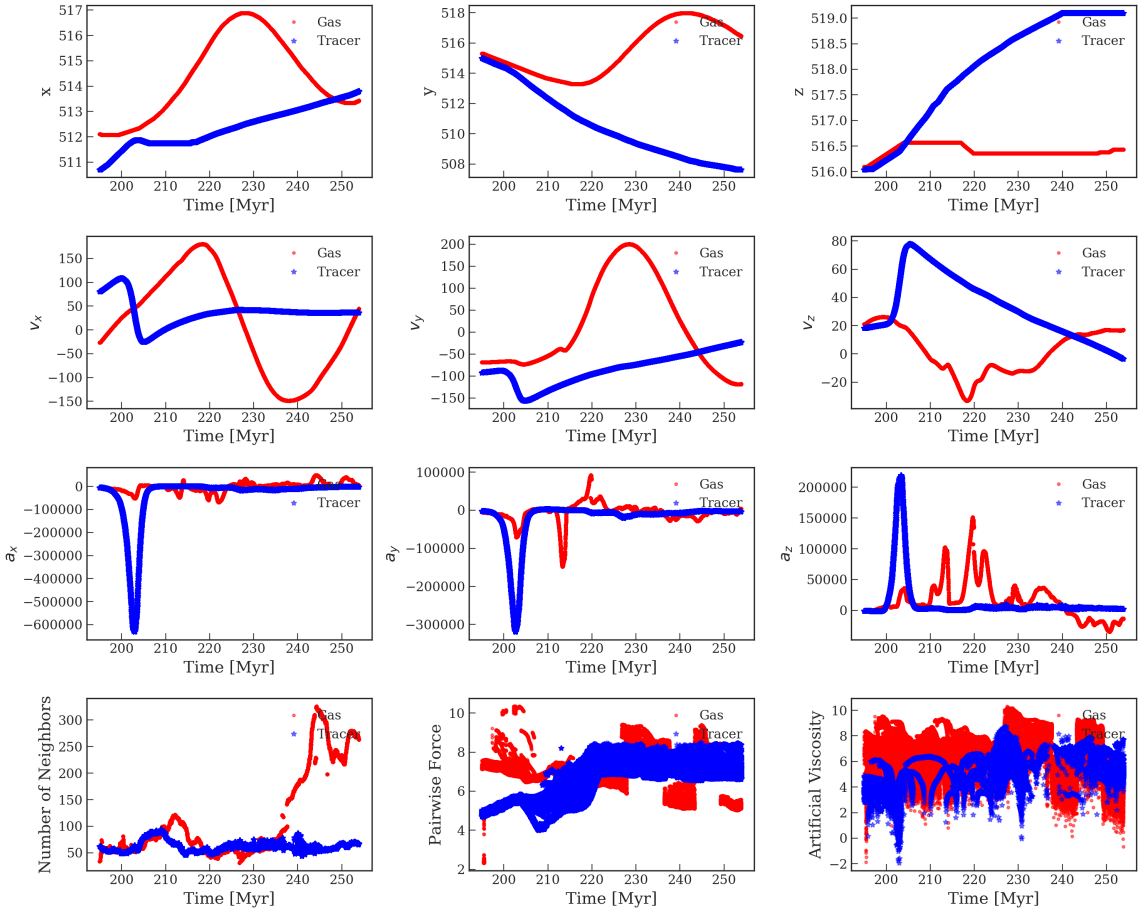

Figure 4.5: Properties of tracer number 148,863 (blue stars) and gas particle 18,983 (red circles) from Run 4103 over simulation time. All velocities and accelerations are scaled for the given refinement, in addition to the pairwise forces and artificial viscosities. Therefore, we can appropriately compare both the tracer and gas particle even if they are in different refinements. All units, except for Number of Neighbors, are internal code units.

4.2.2 Looking at the nearest gas particle to the hot tracer

In addition to considering the properties of an overheated tracer and a single gas particle, we also consider the properties of the *nearest* gas particle to the overheated tracer at each timestep. **Run 4103** was a restart of Run 4101 where the only difference was the tracer that was being tracked in the force calculations. Previously, we looked at a tracer that did not overheat or be ejected from the disk (as seen in Fig. 4.3), so Run 4103 included tracer the tracer shown in Fig. 4.4 that was overheated to beyond 8×10^5 K. Here, we compared the properties of this hot tracer to the properties of the *nearest* gas particle at each data output. Fig. 4.6 shows the properties of the tracer (blue stars) and of the nearest gas particle (red circles), where the nearest gas particle was redetermined for each iteration. The top left panel shows the distance of the nearest gas particle from the tracer over simulation time, and the bottom figure shows the particle ID of the nearest gas particle to see how the nearest particle changes over time. The tracer initially tracks forwards to a high-density region and is then ejected, passing by other gas particles as it leaves the disk. The properties between these two particles are consistent, but because the nearest gas particle changes as the tracer is ejected, there were no obvious systematic conclusions to be drawn.

4.3 Summary

By following the individual tracer particles over time and comparing them to the gas particles, we could see how the smoothing lengths and densities changed when certain tracer particles quickly increased in temperature. We saw that these quick and drastic temperature

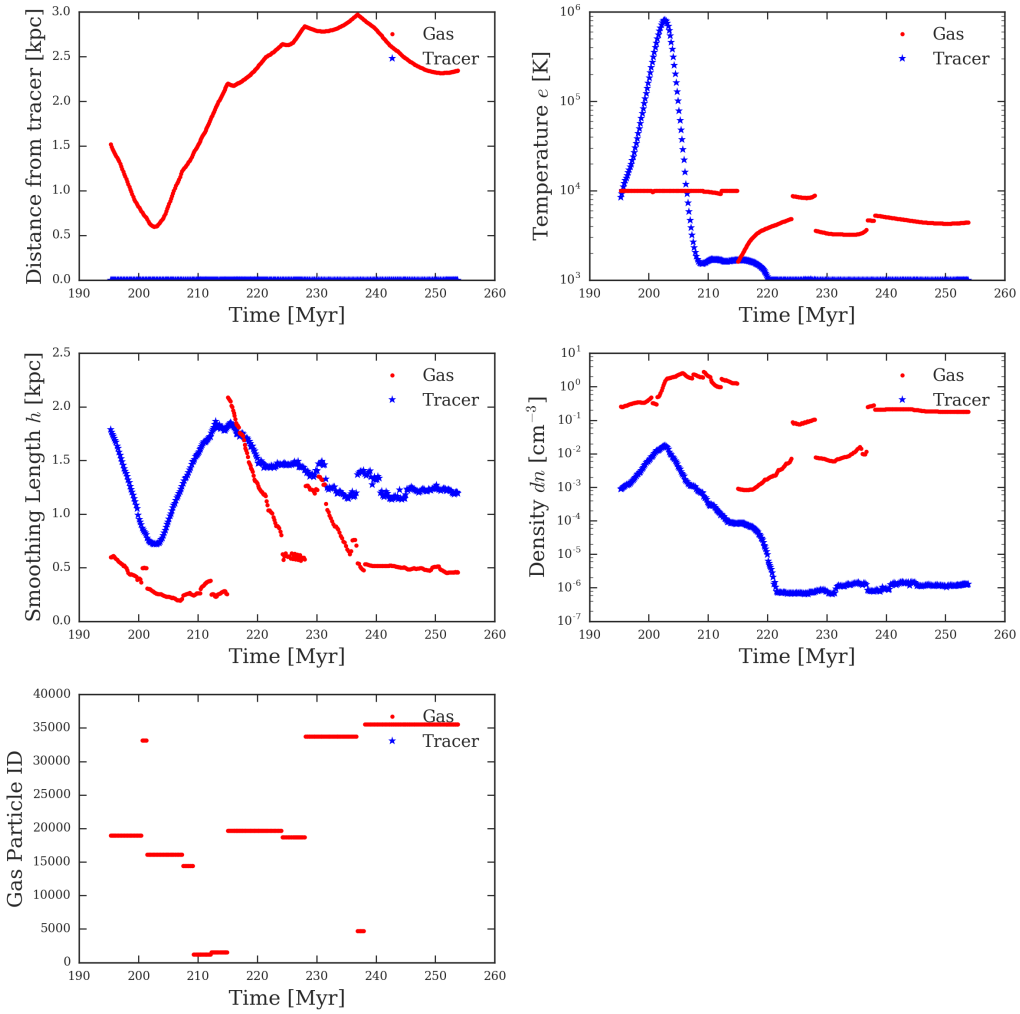

Figure 4.6: Properties of tracer ID 148,863 (blue stars) and of the nearest gas particle (red circles) over simulation time in Run 4103.

increases lead to sudden changes in velocity and thus changes in position, cause the tracers that do become over-heated to be propelled out of the disk, dragging the rest of the contour along with them.

Chapter 5

Runs with temperature limits: Lyapunov results

While we were unable to fully solve the tracer evolution problem, it was possible to set limits on temperature that allowed some thermal evolution beyond the isothermal model used in Fields (2017). Here we summarize those results borrowing methodologies from Fields in the analysis.

5.1 Comparing different initial conditions using the same integration parameters

We ran Models 63 (Run 8063), 118 (Run 8118), and 165 (Run 8165) using the same physical simulation parameters, just with the differing initial particle positions, and these can be seen in Table 5.1. The initial conditions of Model 63 have been described previously in this thesis

Table 5.1: The simulations used to compare the effects of using differing initial conditions with the same physics. The Run ID, iteration, and time that the three runs are compared at are given. The circular velocity at the virial radius V_{200} is the same for all models, and the Q values are the same as seen in Table 3.1.

Run ID	Iteration No.	Time [Myr]	V_{200} [km/s]	Q
8063	15000	316.2	180	0.751
8118	6451	316.2	180	0.893
8165	5963	316.2	180	2.91

and this model is the set of initial conditions used in the majority of the discussion in this work. Models 118 and 165 are described more thoroughly in Foyle (2007), Fields (2017), and Table 3.1, but the main difference between the three models is their stability parameter Q. All three models have the same circular velocity at their virial radii, V_{200}. The runs are compared at different iterations because they evolve differently depending on their Q – if a model is more stable, it requires fewer iterations to get to the same evolutionary time, as seen in the Table. Model 63 is the most unstable, Model 118 is borderline stable with $Q \approx 1$, and Model 165 is the most stable. Together, these models span the range for highly stable, to borderline stable, to unstable. The periodic nature of the system complicates analysis, but the contour approach allows us to account for phase-wrapping, which can happen when the particles have done a full rotation around the centre of the galaxy – two particles may look nearby in position space, but one may have completed a full revolution more than another.

In the simulation, the minimum temperature $e_{\min} = 1000$ K, maximum temperature $e_{\max} = 15,000$ K, `dtnorm` $= 0.25$, and $q_{ij} = 0.5$ for tracers, with the tracer mass equal to 100 $M_\odot$or approximately 0.0005 $\times m_{\text{gas}}$. Note that $e_{\max}$ is set for these three simulations,

so no tracer particles were able to overheat substantially.

Fig. 5.1 shows the contours of the given runs each at 316.2 Myr, as given in Table 5.1. The top panel shows Run 8063, the middle panel shows Run 8118, and the bottom panel shows Run 8165. In all three panels, the x-y plane is given on the left and the x-z plane is given on the right. The three runs are at different stages of their evolution in the figure, where the most unstable model, Run 8063, shows that the contour has rotated about the center at least twice, whereas in the most stable model, Run 8165, the contour has yet to rotate once fully around the center of the disk.

Fig. 5.2 shows the change in length of the contour over time for Run 8063, Run 8118, and Run 8165. As in Fields (2017), the shearing component of the length of the contour for each model has been subtracted from the overall length. Here, a fit is applied to the data at $\geq$200 Myr, where the change in length over time best fits a log-linear slope. The slope of the line in these log-linear plots gives us the Lyapunov exponent, λ. The doubling time and its uncertainty were found using eq. (3.3) and eq. (3.4) from §3.9. Table 5.2 gives the doubling time found in Fields (2017) along with the new doubling time from this fit, and can be compared to Table 3.4 in Fields (2017). It is important to note the physical differences between the simulation in this work and the simulations in Fields (2017), mainly that these test in this section included a temperature ceiling for all of the gas particles, therefore there were no overheated or ejected gas particles, as was achieved in Fields (2017) but by the application of isothermal assumption. In this new work, the pressure in the disk can change by more than an order of magnitude when compared to the isothermal equation of state used in Fields (2017).

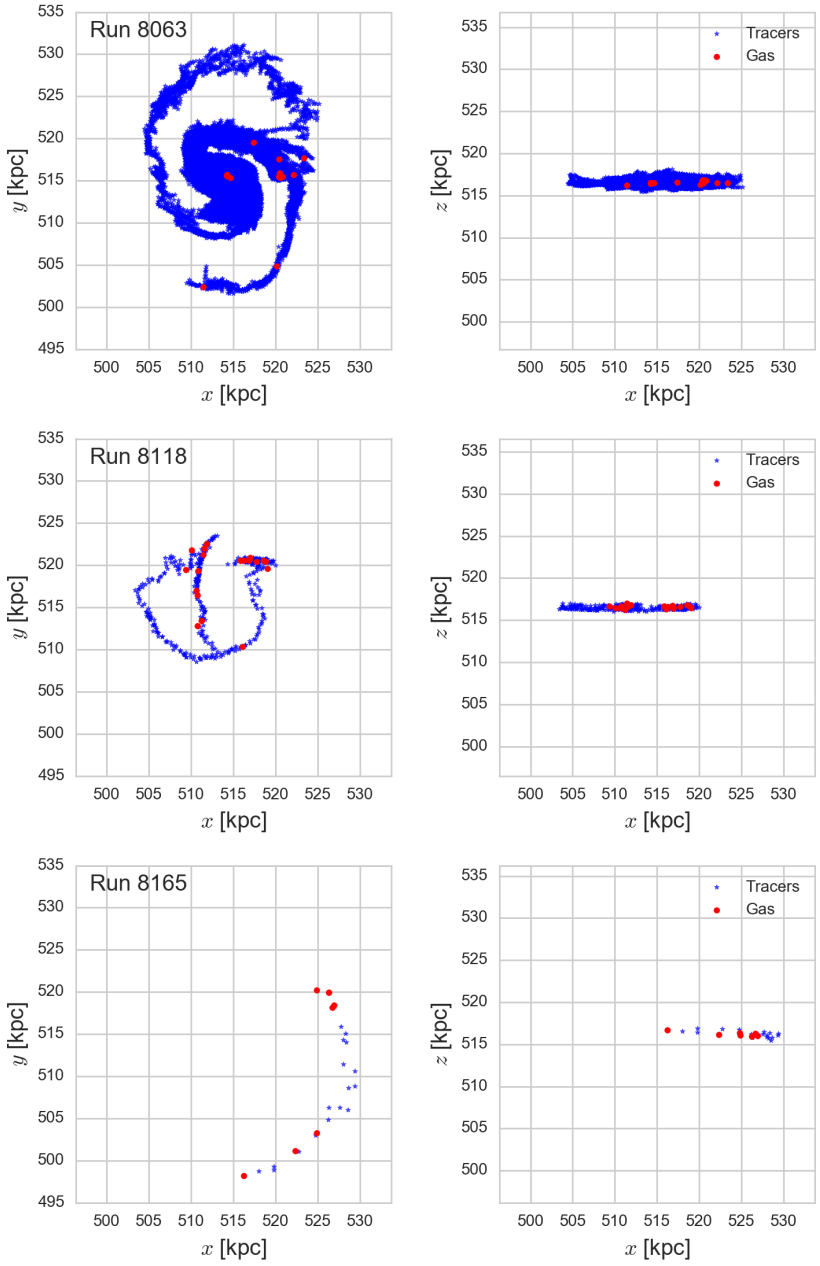

Figure 5.1: Contour showing the position of the gas (red circles) and tracer particles (blue stars) for Run 8063 (top), Run 8118 (middle) and Run 8165 (bottom) at 316.2 Myr, given in Table 5.1. The x-y plane is given in the left column and the x-z plane is shown in the right column.

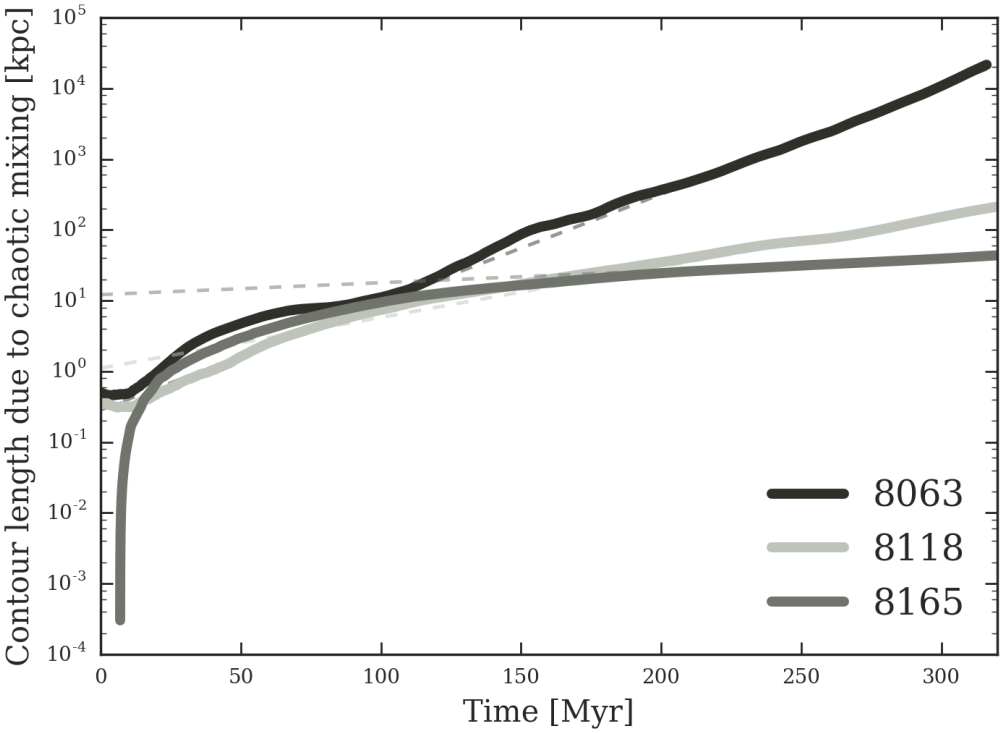

Figure 5.2: Comparison of the change in length of the contours of Run 8063 (top), Run 8118 (middle), and Run 8165 (bottom) until approximately 320 Myr. The dashed lines are fit at 200 Myr and above to calculate λ and t_d, seen in Table 5.2.

Table 5.2: Run IDs used to compare the effects of using differing initial conditions with the same physics. Given here is λ and t_d found in this work, along with t_d from Fields (2017).

Run ID	λ [$\times 10^{-5}$]	t_d this work [Myr]	t_d Fields (2017) [Myr]
8063	1523.9 ± 1.7	45.484 ± 0.051	84.07 ± 0.34
8118	$716.98 \pm .46$	96.675 ± 0.062	122.7 ± 1.4
8165	$169.19 \pm .56$	409.7 ± 1.4	457 ± 10

For all three runs in Table 5.2, t_d is smaller, thus, using our current simulation code, the models experience more mixing than in the isothermal version run in Fields. The higher temperatures that the tracers can experience here also allows for faster sound velocities c_s so we could reasonably expect to get more mixing with higher velocities, which is what we report here. This is because dynamics of regions at different pressures tend to equalize on scales proportional to the sound crossing time which is proportional to $1/c_s$, indicating that higher c_s produces more rapid mixing (e.g. Stasińska, G. et al. (2007)). We can naively estimate what an upper limit on the reduction of mixing time would be by considering the largest possible increase in c_s. Since

$$c_s = \sqrt{\gamma(\gamma - 1)\rho\epsilon} \tag{5.1}$$

where γ is the ratio of specific heats, ρ is the density, and ϵ is the internal energy, we can expect a maximum increase in c_s of approximately $\sqrt{10} \sim 3$ – suggesting the maximum possible reduction in the mixing time would be a factor of 3. In practice, we would likely expect somewhat less than this as increasing the average temperature will likely be lower

than the maximum possible difference from the isothermal to adiabatic cases. Thus, our decrease in t_d by about a factor of two between this work and Fields (2017) is reasonable.

5.2 Calculating full Lyapunov Exponent without contour tracing method

In principle, we can also calculate the Lyapunov exponent by considering two separate simulations with initial conditions that are separated slightly in phase space. However, such an approach has challenges because not all of the system undergoes the same level of mixing. Hence, just considering a system with a displacement of say one particle may not fully capture the evolution. Technically what should be considered is a Lyapunov spectrum. Nonetheless, in this section, we consider variations across simulations with small differences made by changing the position of individual particles. In essence, the difference between the two methods is about trying to follow the change of a contour versus the changes in the boundary of the contour.

Here, seven versions of the same initial conditions were used to estimate a full Lyapunov Exponent of a single model. We used an unmodified version of Hydra that did not include any contour tracing algorithms, therefore we knew the algorithm was accurate and the only difference between the runs was the slight change in the initial position of a single particle. The model chosen was Model 63 from Foyle (2007), and the parameters for this model can be found in Table 3.1. The gas particle ID that was moved was number 3128, and it was moved by three-quarters of the average inter-particle distance at that radius in the model,

or 0.064 kpc. The six modified runs plus the one control run are described in Table 5.3 and all had `dtnorm`=0.25. We note that to *fully* follow the phase space volume, we would have needed a huge number of simulations for each degree of freedom change. Since we did not have the resources or the time for that difficult task, we instead sample a small region of the phase space, and the six modified simulations described here are enough to estimate λ for this model.

Table 5.3: Run identification numbers and descriptions for the variations of Model 63 used to estimate a full Lyapunov exponent. The single gas particle was moved by three-quarters of the interparticle distance at its radius, or about 0.064 kpc from its original position in the designated direction.

Run ID	Description
0630	No modifications
0631	Particle moved in **positive** direction on **x**-axis
0632	Particle moved in **negative** direction on **x**-axis
0633	Particle moved in **positive** direction on **y**-axis
0634	Particle moved in **negative** direction on **y**-axis
0635	Particle moved in **positive** direction on **z**-axis
0636	Particle moved in **negative** direction on **z**-axis

Fig. 5.3 shows the displacement distribution of every gas particle in each of the six simulations of the modified initial conditions shown in Table 5.3 compared to the unmodified Run 0630. Here, we can see that particles are displaced as many as 18.3 kpc from the position they are found in Model 0630 at exactly the same simulation time. However, the majority of gas particles in the simulations (of which there are 40,000 gas particles found in each set of initial conditions) are within 1 kpc of their original location.

Fig. 5.4 shows the difference in phase space between each of the models and the control model Run 0630. Here, the phase space difference is calculated by finding the difference in

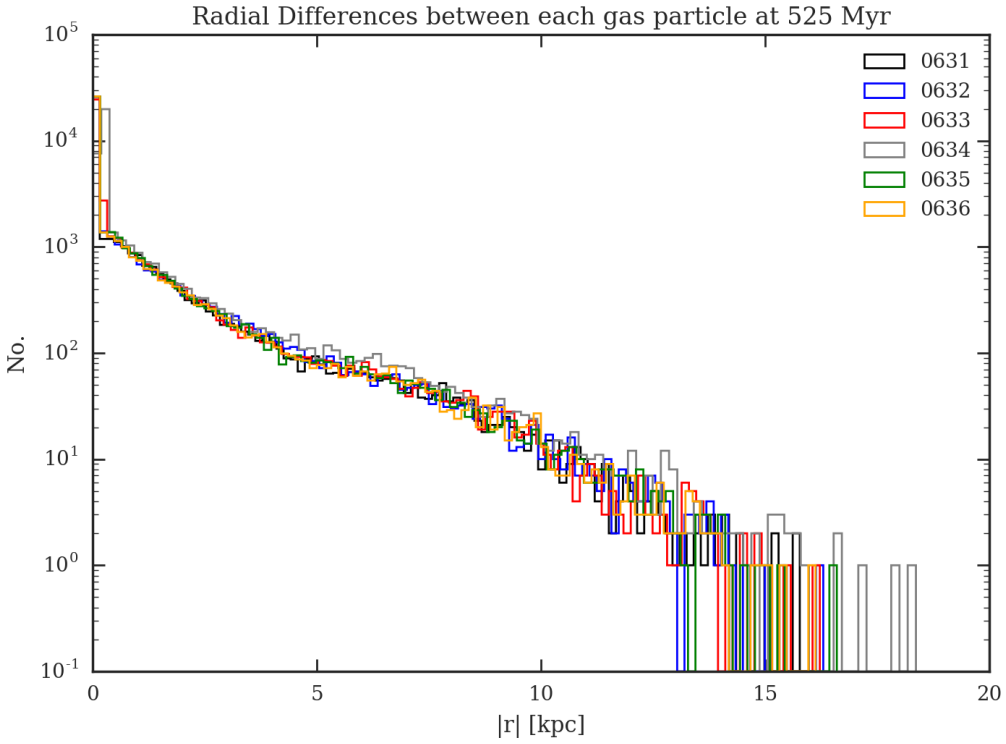

Figure 5.3: The displacement in position of all gas particles at 525 Myr between the the unmodified initial conditions (Run 0630) and each modified version seen in Table 5.3.

position and velocity between every gas particle in the given model and between the control model, where

$$\text{distance} = \frac{\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}}{r_{200}} + \frac{\sqrt{\Delta v_x^2 + \Delta v_y^2 + \Delta v_z^2}}{V_{200}}. \tag{5.2}$$

In the above equation, r_{200} is the virial radius, the radius at which the density is 200 times that of the characteristic overdensity of the disk ρ_{crit} (Foyle 2007) and contains the virial mass M_{200}, such that

$$M_{200} = 200\rho_{\text{crit}}\frac{4\pi}{3}r_{200}^3. \tag{5.3}$$

Further, V_{200} is the circular velocity at the given radius, given by

$$V_{200}^2 = \frac{GM_{200}}{r_{200}}, \tag{5.4}$$

and the above equations are explained further in Foyle (2007). In this and the following figures, $V_{200} = 180$ km/s and $r_{200} = 300$ kpc since it is dependent on background density in cosmology.

Fig. 5.5 shows the difference in phase space over time for all of the gas particles, and the difference in phase space is given by the following:

$$d = \frac{\sum \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}}{r_{200}} + \frac{\sum \sqrt{\Delta v_x^2 + \Delta v_y^2 + \Delta v_z^2}}{V_{200}} \tag{5.5}$$

where the change in values are the difference between the model given and the control model,

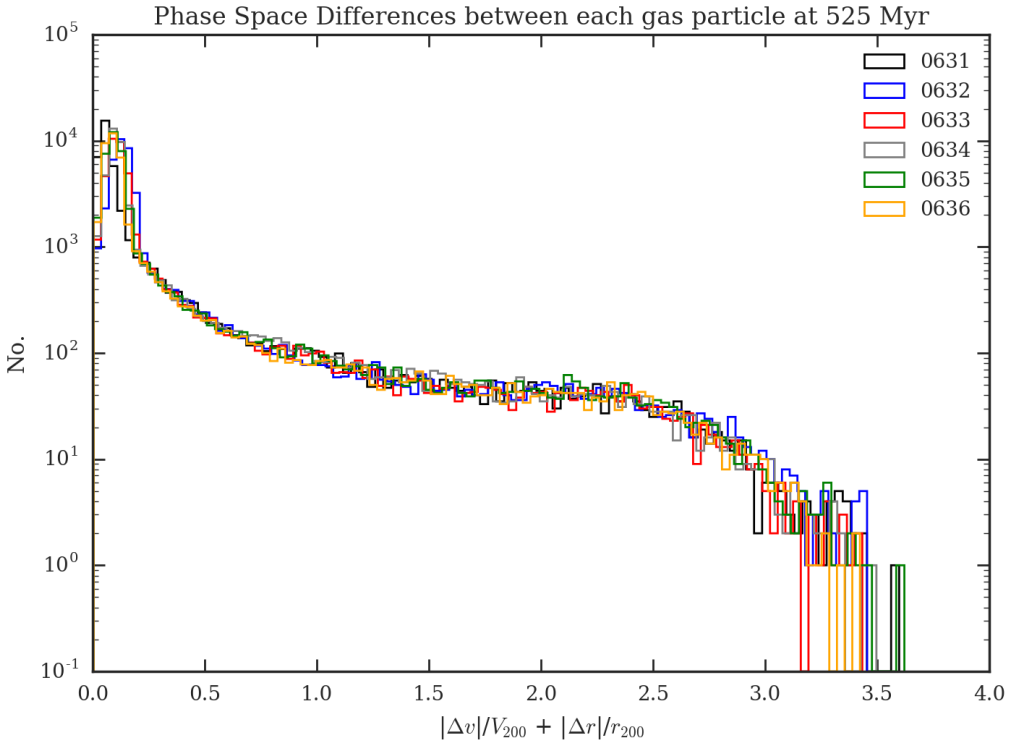

Figure 5.4: The displacement in phase space of all gas particles between the the unmodified initial conditions (Run 0630) and each modified version seen in Table 5.3.

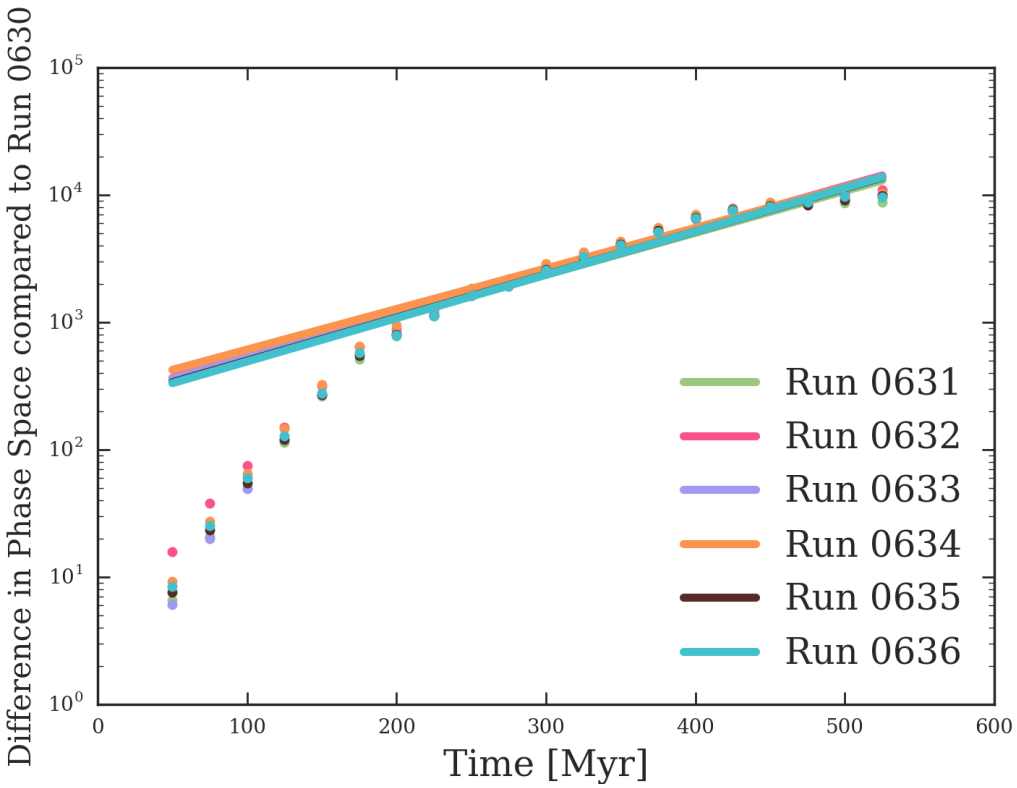

Figure 5.5: The phase space difference over time for all of the gas particles in each run compared to the phase space positions in the control, Run 0630 (circular points). Additionally, the line of best fit to determine the Lyapunov exponent is given for each model, and those results are given in Table 5.4.

Run 0630. This figure shows the difference in phase space for every particle of the given type during each of the outputs over time. Included is also the line of best fit that is fit to all data points at $\geq$ 200 Myr to determine the Lyapunov exponent of each run, and both λ and t_d for the fits in this figure along with their associated 2-σ uncertainties are given in Table 5.4. The main question here is if the subtle change in position of a single gas particle over time was enough to change the amount of chaotic mixing happening in the disk, as given by the Lyapunov exponent. As seen in the table, all λ and t_d for the six comparison runs are within uncertainties of each other. For the curves in Fig. 5.5, the data start to plateau at 525 Myr, which was the end of our simulation runtime.

Table 5.4: Run IDs used to compare the effects of moving just a single particle a small amount with the Lyapunov exponent λ and associated doubling time t_d. The data here are from Fig. 5.5.

Run ID	λ [$\times 10^{-4}$]	t_d [Myr]
0631	32.9 ± 4.9	211 ± 32
0632	33.4 ± 3.7	207 ± 23
0633	33.4 ± 4.3	207 ± 27
0634	31.8 ± 4.3	217 ± 29
0635	33.8 ± 4.6	205 ± 28
0636	34.0 ± 4.5	203 ± 27

The difference in position *only* between each of the six models and the control Run 0630 are presented in Fig. 5.6. This figure is very similar to Fig. 5.5 but instead of finding the difference in phase space using both velocity and position as done there, this figure shows the difference in positions of all gas particles in position only. The data are fit $\geq$ 200 Myr and the slope of the fit λ is given in Table 5.5 along with the doubling times t_d and associated 2σ uncertainties. Here, it can be seen that all λ and t_d agree within their

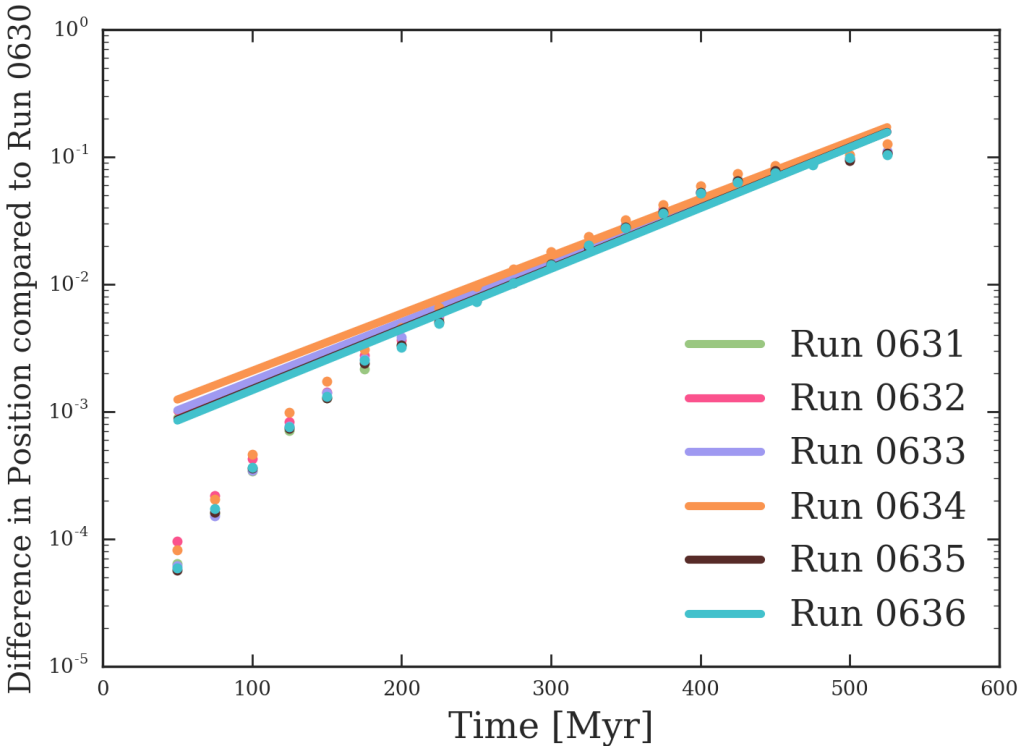

Figure 5.6: The difference in positions only over time for all of the gas particles in each run compared to the positions in the control, Run 0630 (circular points). Additionally, the line of best fit to determine the Lyapunov exponent is given for each model, and those results are given in Table 5.5.

Table 5.5: Run IDs used to compare the effects of moving just a single particle a small amount with the Lyapunov exponent λ and associated doubling time t_d. The data here are from Fig. 5.6.

Run ID	λ [$\times 10^{-4}$]	t_d [Myr]
0631	47.2 ± 4.8	146 ± 15
0632	46.1 ± 4.6	150 ± 15
0633	46.1 ± 4.4	150 ± 14
0634	45.0 ± 4.2	153 ± 14
0635	47.5 ± 4.9	145 ± 15
0636	47.6 ± 4.9	145 ± 15

estimated uncertainties, so even though individual particles may be in different positions throughout the disk in different simulations (seen in more detail in Figs. 5.7 through 5.12), the overall difference in position of the gas particles in each simulation is the same within uncertainties for the simulations presented in this section.

Fig. 5.7 shows the difference in the physical appearance of the gas disk at 525 Myr between Run 0630 (left, the control) and Run 0631 (right). There are some remarkable differences shown here, considering the very small change in the initial conditions between these two models. One thing to note is the position of the gas particle that was moved by three-quarters of the local interparticle separation in the positive direction on the x-axis, this particle is marked in lime green in the figure. In the control model, Run 0630, the gas particle is near the centre of the distribution. However, in Run 0631, the marked particle is near the outer edge of the disk. There are other differences that can be visually seen. Similar figures are given for Run 0632, Run 0633, Run 0634, Run 0635, and Run 0636 in Figs. 5.8 to 5.12. In each of the figures, the marked gas particle that was moved by a very small amount in the initial conditions is in a substantially different position by 525 Myr into

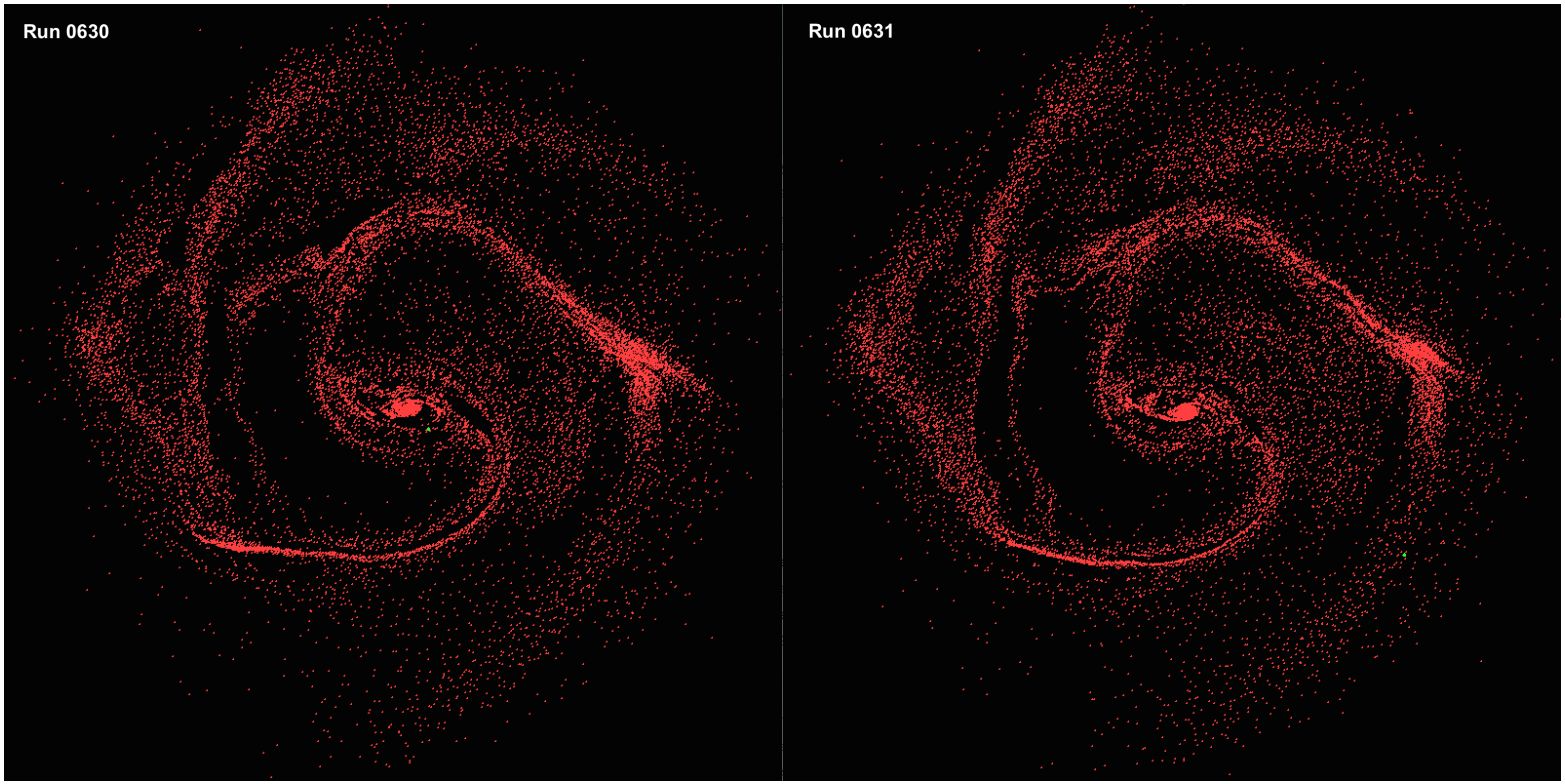

Figure 5.7: The difference in the disk of gas particles at 525 Myr between Run 0630 (left) and Run 0631 (right). The marked green particle is the particle whose initial position was changed by three-quarters of the local interparticle distance in a positive direction along the x-axis. The length of one side in the images is approximately 31 kpc.

the evolution of each disk.

Perhaps unsurprisingly, this analysis has shown that the changes in position space appear to be more significant to the evaluation of the relative amount of mixing than velocity space. However, it is worth emphasizing that both of these variables are bounded in similar ways. Despite a single particle being able to move a surprisingly large distance, the large number of particles in the simulation means that most of the signal is still dominated by the rest of the simulation. A better approach might have been to consider simulations where particles are offset across multiple regions in the disk as opposed to one single particle. Nonetheless, the results do give a useful upper bound on the mixing times, albeit considerably larger than that found via contour approaches.

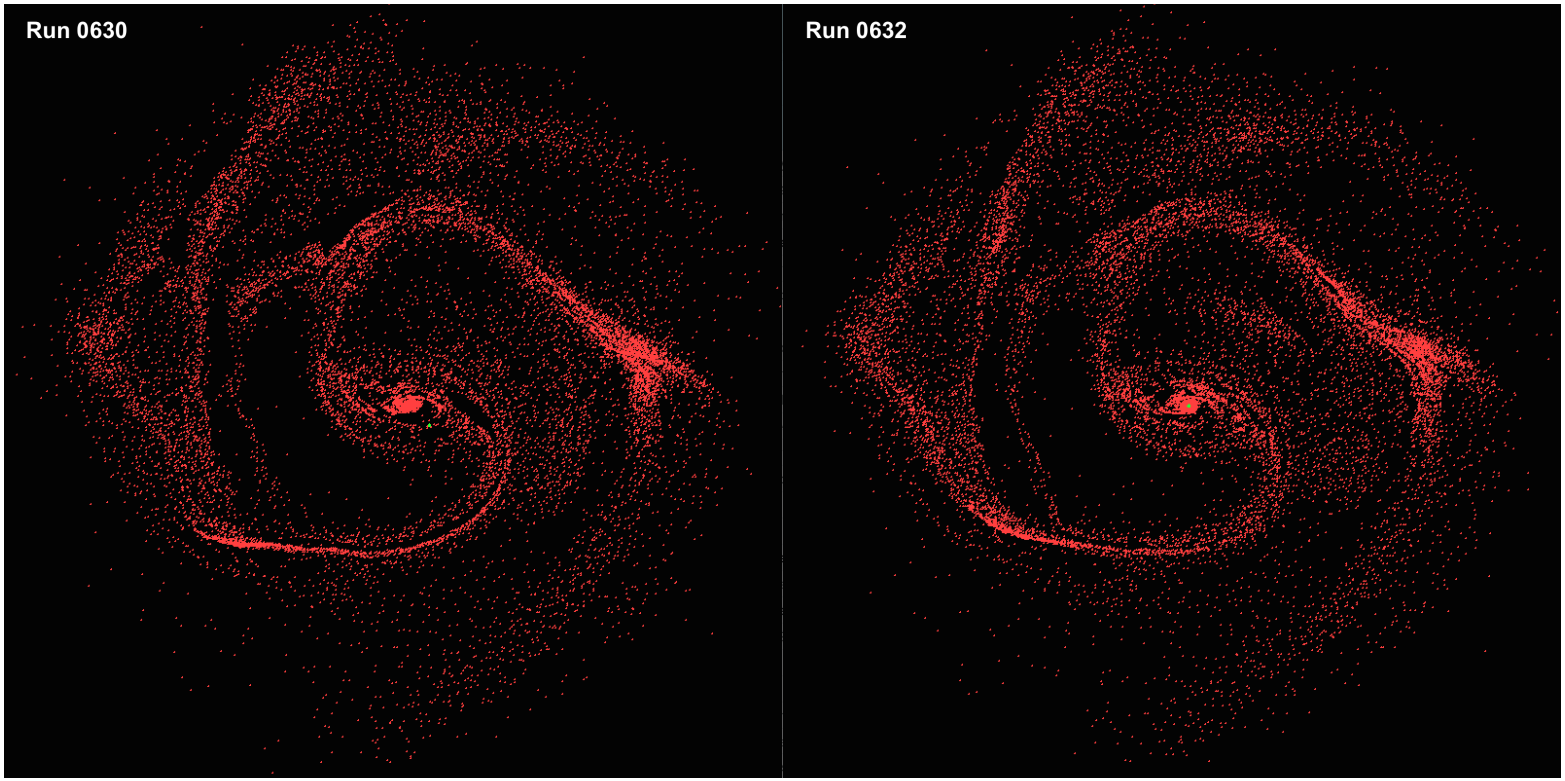

Figure 5.8: The difference in the disk of gas particles at 525 Myr between Run 0630 (left) and Run 0632 (right). The marked green particle is the particle whose initial position was changed by three-quarters of the local interparticle distance in a positive direction along the x-axis. The length of one side in the images is approximately 31 kpc.

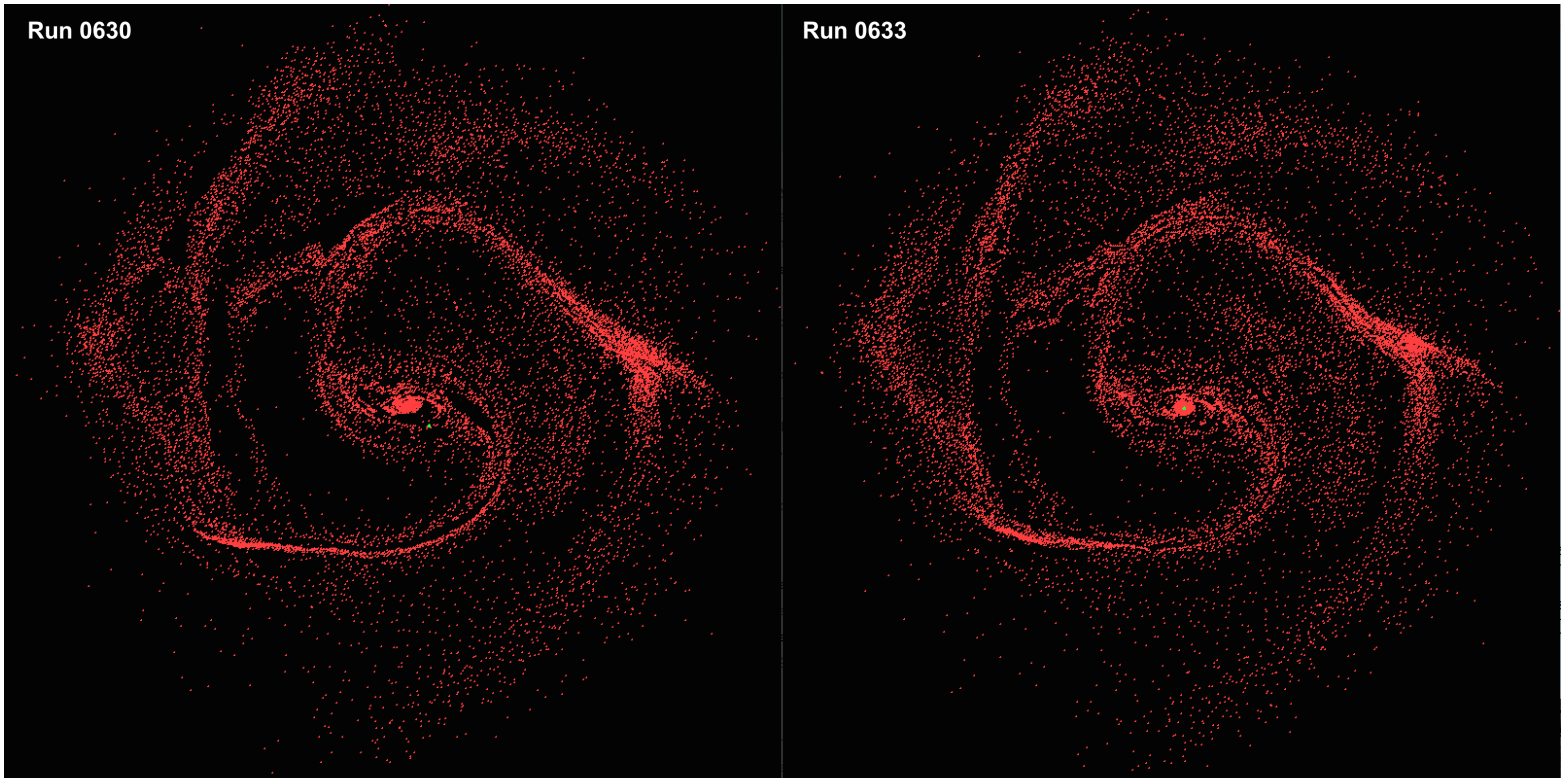

Figure 5.9: The difference in the disk of gas particles at 525 Myr between Run 0630 (left) and Run 0633 (right). The marked green particle is the particle whose initial position was changed by three-quarters of the local interparticle distance in a positive direction along the x-axis. The length of one side in the images is approximately 31 kpc.

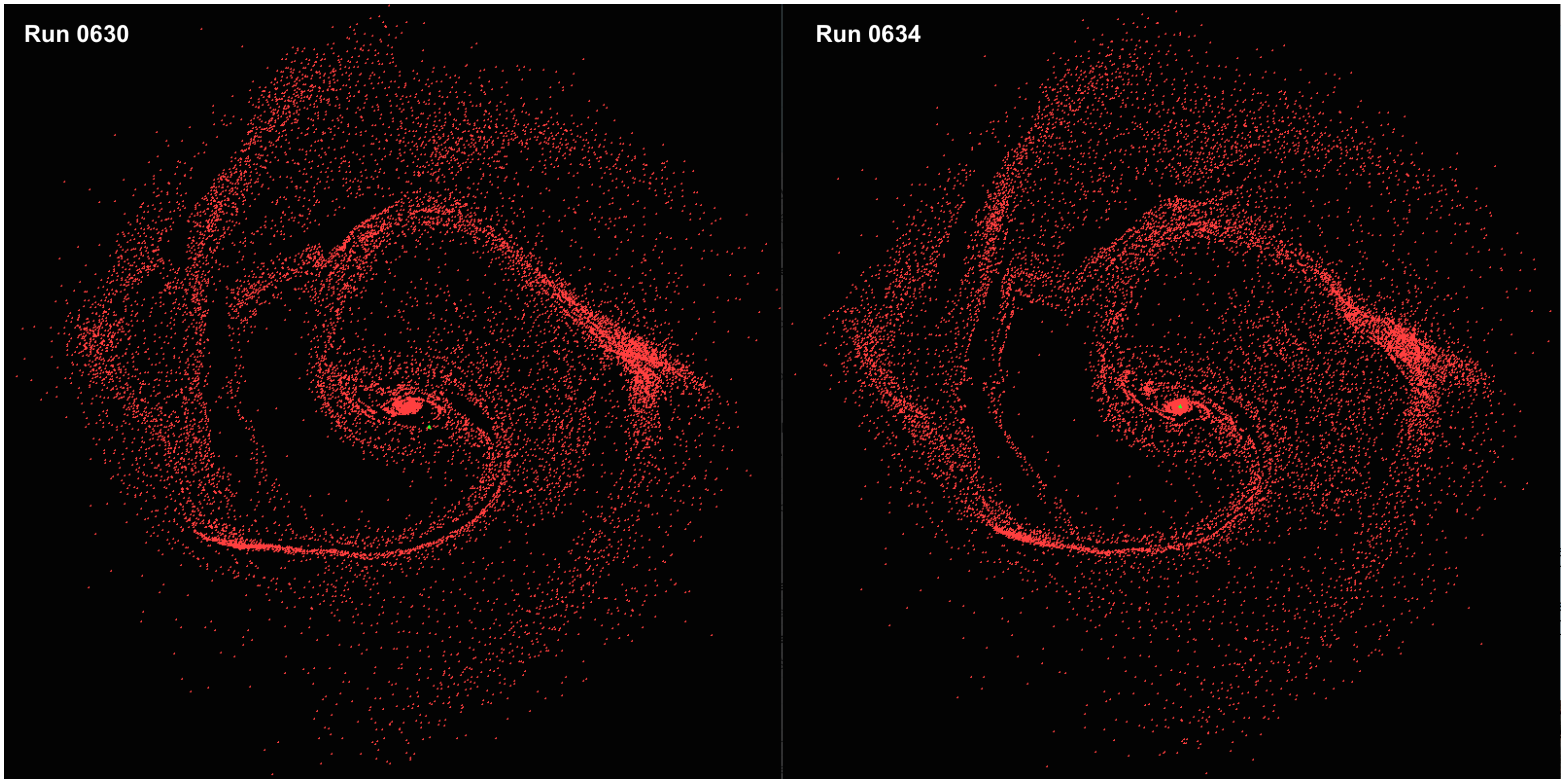

Figure 5.10: The difference in the disk of gas particles at 525 Myr between Run 0630 (left) and Run 0634 (right). The marked green particle is the particle whose initial position was changed by three-quarters of the local interparticle distance in a positive direction along the x-axis. The length of one side in the images is approximately 31 kpc.

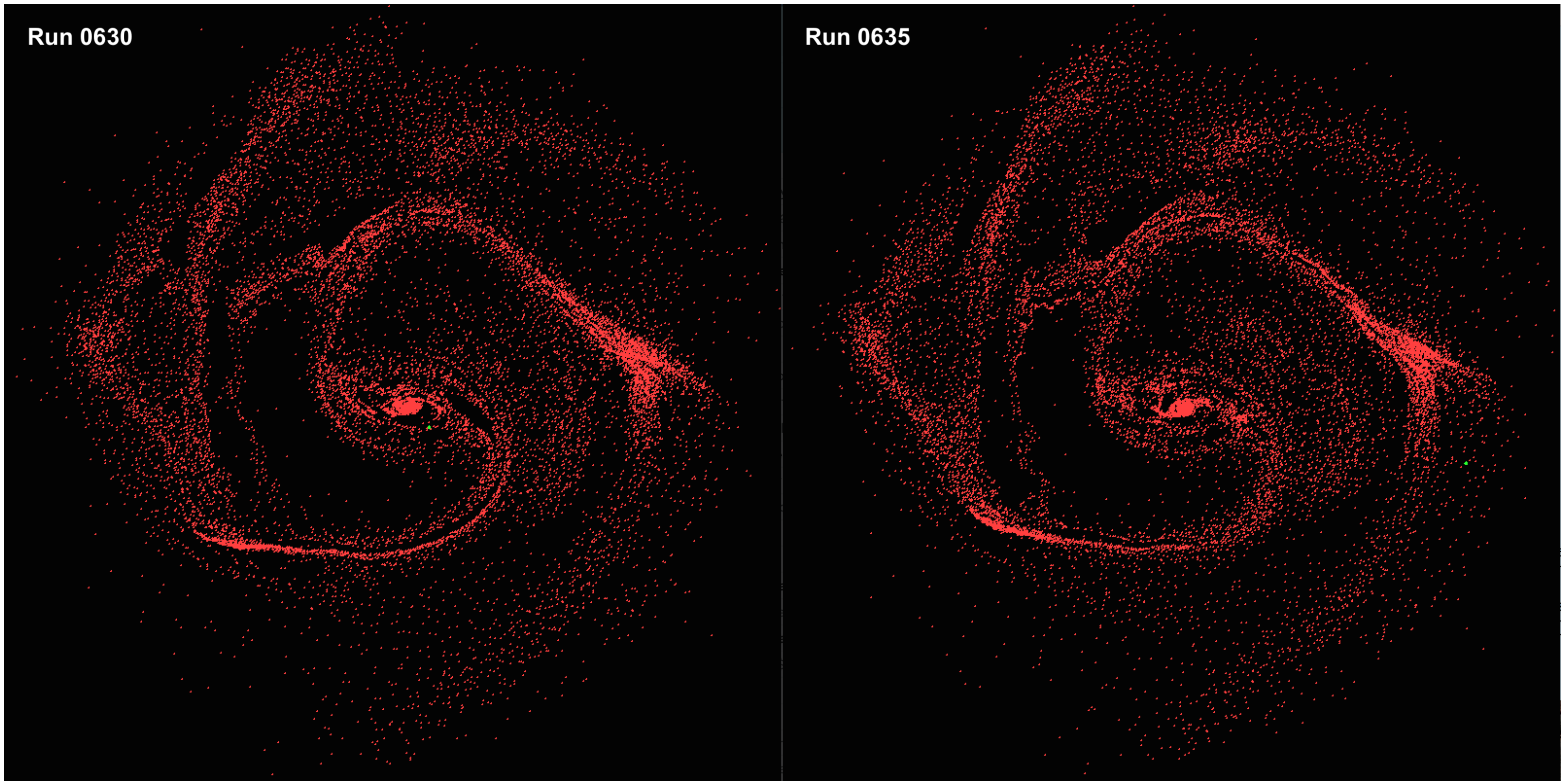

Figure 5.11: The difference in the disk of gas particles at 525 Myr between Run 0630 (left) and Run 0635 (right). The marked green particle is the particle whose initial position was changed by three-quarters of the local interparticle distance in a positive direction along the x-axis. The length of one side in the images is approximately 31 kpc.

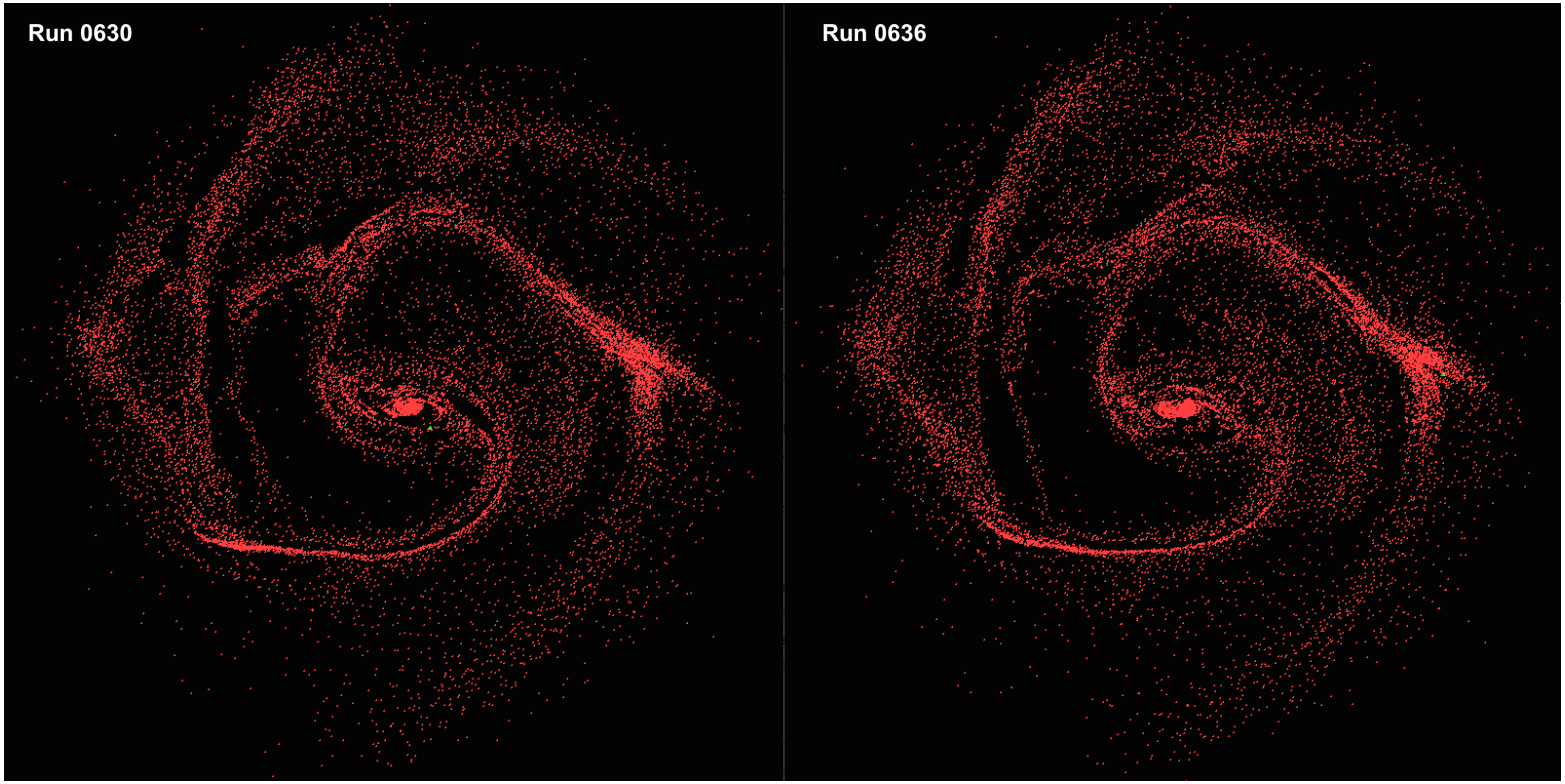

Figure 5.12: The difference in the disk of gas particles at 525 Myr between Run 0630 (left) and Run 0636 (right). The marked green particle is the particle whose initial position was changed by three-quarters of the local interparticle distance in a positive direction along the x-axis. The length of one side in the images is approximately 31 kpc.

Chapter 6

Discussion & Conclusion

This work was an extensive exploration of the investigations initiated in Fields (2017) and aimed to improve the contour advection method with tracers in order to be generally applicable. In this thesis, we examined a number of simulations to determine what properties affected the way that the tracer particles in our contour-tracing method followed the flow of the gas particles. We did not attempt to find a relationship between the stability of a galaxy Q and the amount of chaotic mixing with λ as was done in Fields (2017), but we rather focused on examining and understanding the problems with the tracer method we had developed.

6.1 Discussion

To highlight the importance of the investigation in this thesis, recent work by Genel et al. (2019) and Keller et al. (2019) show that galaxy formation and evolution simulations depend

heavily on the initial conditions, simulation code, and feedback models; and large-scale properties can be affected by perturbations as small as floating point round-off errors. Keller et al. (2019) argue that any small variation between properties of two simulations must have statistical evidence that these differences are not due to stochasticity alone. Our work has shown that there can be large differences in properties of a disk galaxy, in our case the amount of chaotic mixing, by changing small parameters within the integration as well. The unavoidable conclusion of this is that the variance in simulated ensembles contains stochastic elements that we do not yet understand. Thus comparison of simulations to observational results is far more complex than we first anticipated.

From a methodology perspective, Agertz et al. (2007) outlined an instability in simulations around high-density contrasts that is likely contributing to the issues we are seeing. In our simulations, there were a number of instances of high-density regions and where tracer particles were ejected from the disk galaxy. Agertz et al. report that there is an "interference gap" between high- and low-density regions, and particles that find themselves in this gap are pushed by the particles in the high-density region to the area of lower density, an example of which is given in Fig. 4.1. The figure shows a disk-like high-density region in the bottom third of the image, along with a gradual decline in density followed by a gap moving upward from the bottom of the image. A low-density region exists above the gap in the figure, and here a tracer particle is illustrated above the gap. Since more particles from the high-density disk contribute to the smoothed properties of the tracer than do the particles in the low-density region, the tracer gets pushed upwards and outwards of the disk, such that it will eventually be positioned above the interference gap. This type of gap and

outward movement of particles is similar to what we found in this work.

This issue has motivated the formation of new gridless models, and is something that could be considered further when looking for improvements in our contour tracking method with tracers. Other approaches that attempt to avoid this problem include so-called multi-phase SPH techniques, e.g. Ritchie & Thomas (2001).

6.2 Conclusion

We demonstrated in Chapter 2 that in a moderately well-resolved adiabatic spherical collapse model, tracer particles followed the flow of the gas well. For simulations of this kind, essentially those without large density contrasts, many algorithms appeared to produce reasonable results. However, as demonstrated in Chapters 3 and 4, the tracers were not as accurately following the flow of the gas particles in disk galaxies, particularly in the unstable disk galaxy Model 63 from Foyle (2007). We attribute the errors in the tracer flow to their shock heating. In some simulations, the tracers were over-shocked, causing Type I errors where the tracers would overheat and be ejected from the disk. In the remaining simulations, the tracer flow felt too little shock, causing Type II errors where the tracers would not be affected by the shock and would instead just flow right through it. With Type II errors, even though the simulation could continue integrating and evolving, the contour would be too long to account just for the chaotic mixing, leading to an over-estimate of the Lyapunov exponent.

It is worth noting that we did improve upon the contour method presented in Fields

(2017). We showed that implementing maximum temperatures produced contours that better follow the tracer flow, and we were able to compare doubling times found (in §5.1) to those presented in Fields (2017). Further, our tracer method was found to perform more reliably on very stable models, such as Model 165 (§5.1).

Our contour-tracing method was based on the work of Waugh & Plumb (1994), but it is important to note that their CAS (contour advection surgery) method included both inserting tracers when particles were too far spread apart (as we have done), *and* a surgery to remove the tracers when particles were nearby. Their method was specific for atmospheric physics, so they even included instances where the contour would be separated into multiple contours in their surgery, but that would not have been appropriate for the goals of our work. If we had implemented the removal of tracers in the surgery method, however, there may have been a difference in the accuracy of the tracer particles, as there would be fewer tracer particles bunched up around the location of a shock front. This could have improved the accuracy of our tracer flow.

Our method is ultimately limited in its current implementation because there is no bound on the number of tracers that can be added, and even in a well-behaved system, the number of tracers could become overwhelmingly large. Exploration of the CAS method and removal of tracers could be appropriate for future work.

We report smaller doubling times t_d than found in Fields (2017) when allowing more hydrodynamical evolution, indicating more mixing. Allowing the tracers to evolve in temperature rather than be isothermal as in Fields contributed heavily to this results. In §5.1, we estimated that the maximum amount that the doubling time could be reduced was a

factor of three, and in our results (Table 5.2), we found that the doubling times were reduced by about a factor of two at the most.

In well-resolved density gradients, the tracer method that was developed in Fields and further studied in this work can be used to measure the amount of mixing present. However, this work shows that shock capturing in unstable disks is where this contour tracing method struggles. If the shock capturing is too high, then tracer particles are ejected from the disk in Type I errors. If the shock capturing is too low, however, then the tracer particles just move through the shocks without interacting with it, a Type II error which does not accurately capture the flow of the gas particles through the simulation. When there are instabilities and steep density gradients, this tracer method fails. In Model 63 and its variations (shown extensively in Chapters 3, 4, and 5), the high density contrast gave us the perfect worse-case scenario to test this contour tracing method and proved to be beyond the capability of any model we devised.

Additionally, our tracer method may have been more successful in a newer Lagrangian code such as GIZMO (Hopkins 2015) that uses a pointwise sampling method and mesh grids and a smoothed volume or a code that addresses the gap problem (e.g. Wadsley et al. (2017)). The difficulty in using a newer code, however, is that we need the contour to produce exact lengths, and Monte Carlo-based tracers that rely on statistical properties would not be appropriate for this method. Tracers that go through cell faces get split, and it is also likely that any bends in the contour as it enters low-density regions would be smoothed out by the Monte Carlo tracers. Using this, the length of the contour may be less than what would be appropriate for any given simulation that encounters low-density

regions, something that we found was common in the models that were investigated in this thesis.

The exact length of the contour change over time is the measurement we needed in order to measure the chaotic mixing on small scales, and this proved very difficult even with standard SPH methods. In terms of its application, this method is further hindered by the fact that chaotic mixing is not measurable directly in observations, although it can potentially be inferred. We do however believe that future examinations of tracer methods in these codes would be a worthwhile avenue of investigation, as we have shown that our tracer method can be improved upon in some ways.

Bibliography

Agertz, O., Moore, B., Stadel, J., et al. 2007, MNRAS , 380, 963

Arnold, V. 1963, Russian Mathematical Surveys, 18, 9

Baugh, C. M. 2006, Reports on Progress in Physics, 69, 3101

Baugh, C. M. 2008, Philosophical Transactions of the Royal Society of London Series A, 366, 4381

Binney, J. & Tremaine, S. 1994, Galactic Dynamics, 1st edn., Princeton Series in Astrophysics (Princeton University Press)

Book, L. G., Brooks, A., Peter, A. H. G., Benson, A. J., & Governato, F. 2011, Monthly Notices of the Royal Astronomical Society, 411, 1963

Colless, M., Dalton, G., Maddox, S., et al. 2001, MNRAS , 328, 1039

Couchman, H. M. P. 1991, ApJ , 368, L23

Couchman, H. M. P., Pearce, F. R., & Thomas, P. A. 1996, arXiv e-prints, astro

Couchman, H. M. P., Thomas, P. A., & Pearce, F. R. 1995, ApJ , 452, 797, with modifications as detailed in Thacker *et al.* MNRAS, 319, 619, 2000.

Courant, R., Friedrichs, K., & Lewy, H. 1967, IBM Journal of Research and Development, 11, 215

Dubey, A., Daley, C., ZuHone, J., et al. 2012, ApJS , 201, 27

Eckmann, J. P. & Ruelle, D. 1985, Rev. Mod. Phys., 57, 617

Evrard, A. E. 1988, MNRAS , 235, 911

Fardal, M., Katz, N., Roberts, S., et al. 2019, Monthly Notices of the Royal Astronomical Society, 484, 2021

Federrath, C., Glover, S. C. O., Klessen, R. S., & Schmidt, W. 2008, Physica Scripta, 2008, 014025

Fields, T. M. 2017, B.sc. honours thesis, Saint Mary's University

Foyle, K. 2007, Master's thesis, Queen's University

Frenk, C. S., White, S. D. M., Bode, P., et al. 1999, The Astrophysical Journal, 525, 554

Genel, S., Bryan, G. L., Springel, V., et al. 2019, ApJ , 871, 21

Genel, S., Vogelsberger, M., Nelson, D., et al. 2013, MNRAS , 435, 1426

Gingold, R. A. & Monaghan, J. J. 1977, MNRAS , 181, 375

Goldhirsch, I., Sulem, P. L., & Orszag, S. A. 1987, Physica, 27D, 311

Goldstein, H., Poole, C. P., & Safko, J. L. 2001, Classical Mechanics (Addison-Wesley)

Henon, M. & Heiles, C. 1964, AJ , 69, 73

Hopkins, P. F. 2015, MNRAS , 450, 53

Ivezić, v., Tyson, J. A., Acosta, E., et al. 2008

Katz, N., Weinberg, D. H., & Hernquist, L. 1996, ApJS , 105, 19

Keller, B. W., Wadsley, J. W., Wang, L., & Kruijssen, J. M. D. 2019, MNRAS , 482, 2244

Kolmogorov, A. 1954in , 2–3

Lacey, C. & Cole, S. 1993, MNRAS , 262, 627

Lorenz, E. N. 1963, Journal of the Atmospheric Sciences, 20, 130

Lucy, L. B. 1977, AJ , 82, 1013

Mitchell, N. L., McCarthy, I. G., Bower, R. G., Theuns, T., & Crain, R. A. 2009, MNRAS , 395, 180

Monaghan, J. J. 1992, ARA&A , 30, 543

Morris, J. P. & Monaghan, J. J. 1997, Journal of Computational Physics, 136, 41

Moser, J. 1962, On invariant curves of area-preserving mappings of an annulus. Message from the Academy of Sciences, Gottingen II, Math. Phys.

N-Body Shop. 2011, TIPSY: Code for Display and Analysis of N-body Simulations, Astrophysics Source Code Library

Naab, T. & Ostriker, J. P. 2017, Annual Review of Astronomy and Astrophysics, 55, 59

Nelson, D., Vogelsberger, M., Genel, S., et al. 2013, MNRAS , 429, 3353

Peebles, P. J. E. 1969, ApJ , 155, 393

Press, W. H. & Schechter, P. 1974, ApJ , 187, 425

Price, D. J. & Federrath, C. 2010, in Astronomical Society of the Pacific Conference Series, Vol. 429, Numerical Modeling of Space Plasma Flows, Astronum-2009, ed. N. V. Pogorelov, E. Audit, & G. P. Zank, 274

Rafikov, R. R. 2001, MNRAS , 323, 445

Ritchie, B. W. & Thomas, P. A. 2001, MNRAS , 323, 743

Robertson, B. E., Kravtsov, A. V., Gnedin, N. Y., Abel, T., & Rudd, D. H. 2010, Monthly Notices of the Royal Astronomical Society, 401, 2463

Schoeberl, M. R. & Bacmeister, J. T. 1993, in The Role of the Stratosphere in Global Change, ed. M.-L. Chanin (Berlin, Heidelberg: Springer Berlin Heidelberg), 135–152

Somerville, R. S. & Davé, R. 2015, ARA&A , 53, 51

Springel, V., Frenk, C. S., & White, S. D. M. 2006, Nature , 440, 1137

Springel, V. & White, S. D. M. 1999, Monthly Notices of the Royal Astronomical Society, 307, 162

Stasińska, G., Tenorio-Tagle, G., Rodríguez, M., & Henney, W. J. 2007, A&A, 471, 193

Thacker, R. J. & Couchman, H. M. P. 2006, Computer Physics Communications, 174, 540

Thacker, R. J., Tittley, E. R., Pearce, F. R., Couchman, H. M. P., & Thomas, P. A. 2000, MNRAS , 319, 619

Thomas, P. A. & Couchman, H. M. P. 1992, Monthly Notices of the Royal Astronomical Society, 257, 11

Toomre, A. 1964, ApJ , 139, 1217

Trac, H. & Pen, U.-L. 2003, Publications of the Astronomical Society of the Pacific, 115, 303

Vitvitska, M., Klypin, A. A., Kravtsov, A. V., et al. 2002, The Astrophysical Journal, 581, 799

Wadsley, J. W., Keller, B. W., & Quinn, T. R. 2017, MNRAS , 471, 2357

Wang, B. & Silk, J. 1994, ApJ , 427, 759

Waugh, D. W. & Plumb, R. A. 1994, Journal of the Atmospheric Sciences, 51, 530

White, S. D. M. & Rees, M. J. 1978, MNRAS , 183, 341

Wolf, A., Swift, J., Swinney, H., & Vastano, J. 1985, Physica D: Nonlinear Phenomena, 16, 285

York, D. G., Adelman, J., Anderson, Jr., J. E., et al. 2000, AJ , 120, 1579