Dissertation

submitted to the

Combined Faculty of Mathematics, Engineering and Natural Sciences

of Heidelberg University, Germany

for the degree of

Doctor of Natural Sciences

Put forward by

Johannes Meyer

born in: Cologne, Germany

Oral examination: 10th July 2024

Computationally Exploring Solar System Scale Dynamics

Implicit Hydrodynamics Solver Development, Simulation of Goldreich-Schubert-Fricke Instability, and Planetesimal Formation Analysis

Referees: Prof. Dr. Hubert Klahr

Prof. Dr. Matthias Bartelmann

Abstract

Numerical methods to solve the Navier-Stokes equations (NSE) are of fundamental importance in theoretical astronomy. In the theory of planet formation, they are used to tackle open problems such as the turbulence-generating mechanisms in protoplanetary disks or the formation of planetesimals from collapsing clouds of pebbles. In the context of stellar interiors, hydrodynamic instabilities described by the NSE are believed to be important in mediating angular momentum transport. This thesis introduces the novel, globally implicit hydrodynamics solver MATRICS. A combination of numerical methods, allowing for extensibility and efficient use in the low Mach number regime, is presented, and its functionality is proven. The MATRICS code is used to conduct the first-ever global simulations of the Goldreich-Schubert-Fricke instability (GSF) in a pressure-supported environment. The GSF is found to operate with analytically expected growth rates in the isothermal case. The predicted dependence on the Prandtl number in the diffusion process-controlled thermalization regime is confirmed. For the first time, hints to the concurrent presence of Convective Overstability (COS) within the same environment are found. GIZMO simulations of collapsing pebble clouds, modeling the Asteroid-and Kuiper-Belt, are also analyzed. The mass distribution of simulated planetesimals at 2.3 AU distance from a solar mass star matches the observed initial mass function (IMF) of asteroids. The size distribution of planetesimals at 25 AU shows an excess of small objects but follows a profile similar to the asteroid's IMF for larger objects. By incorporating pebble-to-gas coupling, it is confirmed that angular momentum conservation plays the single most important role in determining the planetesimal IMF. Through analysis of their formation history, it is found that many planetesimals are compounds of smaller objects with typical sizes of $\sim 10 - 30$ km. This is consistent with recent observations of objects in the Main-/ and Kuiper-Belt.

Zusammenfassung

Numerische Methoden zur Lösung der Navier-Stokes-Gleichungen (NSE) sind von fundamentaler Bedeutung in der theoretischen Astronomie. In der Theorie der Planetenentstehung werden sie eingesetzt, um offene Fragen wie die zu turbulenzerzeugenden Mechanismen in Protoplanetaren Scheiben oder die Bildung von Planetesimalen aus kollabierenden Staubwolken zu untersuchen. Es wird angenommen, das hydrodynamische Instabilitäten eine Rolle bei dem Transport von Drehimpuls im Inneren von Sternen spielen. Diese Arbeit führt den global impliziten Code zur Lösung der NSE - MATRICS - ein. Eine Kombination numerischer Methoden, die sowohl Erweiterbarkeit, also auch effiziente Nutzung im Bereich niedriger Machzahlen und diffusionsdominierter Prozesse ermöglicht, wird vorgestellt und ihre Funktionalität nachgewiesen. Mit dem MATRICS-Code wurden die ersten globalen Simulationen der Goldreich-Schubert-Fricke-Instabilität (GSF) in einer druckdominierten Umgebung durchgeführt. Es wurde festgestellt, dass die GSF im isothermen Regime mit analytisch erwarteten Raten wächst. Die vorhergesagte Abhängigkeit der GSF-Wachstumsrate von der Prandtl-Zahl wurde, im Regime der durch Diffusionsprozesse kontrollierten Kühlung, bestätigt. Erstmals wurden Hinweise auf das gleichzeitige Vorhandensein der konvektiven Überstabilität (COS) in dieser Umgebung gefunden. GIZMO-Simulationen von kollabierender Staubwolken, welche die Formation von Planetesimalen in Asteroiden- und Kuipergürtel modellieren, wurden ebenfalls analysiert. Es wurde festgestellt, dass die Massenverteilung simulierter Planetesimale in einer Entfernung von 2,3 AE zu einem sonnenähnlichen Stern mit der beobachteten ursprünglichen Massenverteilung (IMF) von Asteroiden übereinstimmt. Die Größenverteilung von Planetesimalen bei 25 AE zeigt zwar einen Überschuss an kleinen Objekten, folgt jedoch für größere Objekte einem Profil, welches der IMF von Asteroiden ähnelt. Durch Modellierung von Staub-zu-Gas-Kopplung wurde festgestellt, dass der Ursprüngliche Drehimpulsgehalt der kollabierenden Wolke entscheidend für die Form der IMF ist und nicht etwa dieser Effekt. Die Analyse ihres Formationsprozesses zeigt, dass viele Planetesimale aus kleineren Objekte mit typischen Größen von $\sim 10 - 30$ km bestehen. Dies steht im Einklang mit jüngsten Beobachtungen von Objekten im Asteroiden- und Kuipergürtel.

Contents

1 Introduction

Throughout the history of humankind, the interest in the day and night skies has been a constant across many cultures. With the first evidence of human interest in astronomy dating back at least 16,000 years and proof of the first application of mathematics in the context of Astronomy at least 3,500 years ago by the ancient Babylonians, Astronomy is the oldest of the natural sciences. The discovery of Mercury, Venus, Mars, Jupyter, and Saturn also dates back to the ancient Babylonian era. The term 'Astronomy' itself translates from the ancient Greek language to 'law of the stars,' and it was already noted by Anaxagoras almost 2,500 years ago that the sun may be one of these stars, revealing an astonishing level of understanding of the skies. Aristarchus of Samos, another Greek savant, was the first to propose a heliocentric system 200 years later. It took almost 2,000 years for this idea to finally be accepted as a fact in the Renaissance.
In the course of its long history, Astronomy has seen many scientific revolutions and today, Astronomy is a matured science aiming to make theoretical predictions that are testable by observations. One of these revolutions was the invention of the telescope in the early 17th century. It enabled Galileo to discover the 'Galilean' moons around Jupyter, the first discovered astronomical objects orbiting around an object other than the earth or sun. By the middle and end of the 18th century, respectively, Kant [1755] and Laplace [1796] hypothesized the birthplace of planets and their satellites to be an atmosphere-like structure around the sun. This model, known as the Nebular Hypothesis, allowed for an explanation of planet formation for the first time without direct divine intervention. The invention of the Charge Coupled Device (CCD), alongside countless other technical progressions since then, has enabled astronomers to directly observe these 'nebulae' outside of our own solar system. Today, they are termed 'protoplanetary disks' (PPDs), expressing our knowledge that planets indeed form inside these disks around young stars. Observations show not only that PPDs exist, but also that they evolve in time and exhibit substructure (Miotello et al. [2023], Benisty et al. [2023] and references therein).
It is the task of theorists to explain the evolution process of PPDs, link it to their observed substructure, and describe the formation process of planets and their satellites as consistently as possible. Likewise, as with observations, the tools used by theorists to do their work have gone a long way. From the methods used to describe celestial mechanics to the description of the theory of waves, mathematics has shaped the way theoretical astronomy is done and vice versa. With the invention of computers and subsequently dedicated numerical techniques, another class of powerful methods has found its way into the theorists' toolbox. Today, the combined application of analytical and numerical methods has led to a dramatic improvement

in our understanding of the formation process of planets. Nonetheless, neither is planet formation a 'solved problem,' nor are all available numerical methods used in astrophysics to their full extent.
This thesis is divided into three parts. The first contributes to astrophysicists' numerical toolbox by developing the globally implicit MATRICS code. The second contributes to the understanding of the Goldreich-Schubert-Fricke instability and the Convective Overstability, by modeling them in the context of stellar interiors, probing theoretical considerations in that environment. The third part contributes to the formation theory of planets by analyzing pebble cloud-collapse simulations that aim to reproduce the initial planetesimal mass function observed in the Asteroid- and the Kuiper belt. The thesis is structured as follows

- **Chapter 1**: The rest of this chapter gives, in the first part, an introduction to the current theory of planet formation and, with it, the physical context of this work. In the second part, it gives an overview of implicit numerical methods and a brief comparison to explicit methods.

- **Chapter 2**: This chapter introduces the MATRICS code. It outlines the utilized spatial and temporal discretization schemes as well as the solution procedure to the resulting system of equations. The chapter concludes with various test cases, proving the operability of the code.

- **Chapter 3**: In this chapter, the MATRICS code is used to show the first global simulations of the Goldreich-Schubert-Fricke instability and the Convective Overstability in stellar interiors. The chapter shows simulations using conditions inspired by those in the sun's radiative zone.

- **Chapter 4**: This chapter analyzes pebble cloud collapse simulations to compare the initial mass function of planetesimals in the asteroid vs. the Kuiper belt. It also introduces a simple merger tree algorithm that is used to follow the formation history of the planetesimals that contribute to the initial mass function.

- **Chapter 5**: This chapter summarizes the most important findings from the previous chapters and gives an outlook on future work.

1.1 Protoplanetary Disks

This section briefly overviews the physics of protoplanetary disks (PPD). It starts with a general introduction in Section 1.1.1. With considerations of the equilibrium state of gas in a PPD in Section 1.1.2, the geometry of such disks is introduced. Section 1.1.3 extends the considerations to the presence of dust in PPDs. Section 1.1.4 gives an overview of instabilities that may be responsible for the observed substructure in PPDs. The transition to the formation of planetesimals is made in Section 1.1.5.

1.1.1 The General Picture

Protoplanetary disks, in combination with the still-forming star they encompass also referred to as Young Stellar Objects (YSOs), are a byproduct of star formation in the cold and dense cores of molecular clouds. They can be classified into four categories, depending on the slope of their near- to mid-infrared spectral energy distribution (SED) F_λ (Lada and Wilking [1984], see e.g Williams and Cieza [2011] and Armitage [2020] for further reading). This slope is defined as

$$\alpha_{IR} \equiv \frac{d \log \lambda F_\lambda}{d \log \lambda}. \tag{1.1}$$

Class 0 disks exhibit very little or no emission in the infrared. This class represents the main accretion phase of the disk. The infrared emission is $\alpha_{IR} < 0.3$ for class I disks, with no clear separation from class 0 disks. Class I disks are still embedded in an envelope. This envelope is not present for class II disks, where $-1.6 < \alpha_{IR} < -0.3$. Sources with $-0.3 < \alpha_{IR} < 0.3$ are referred to as flat spectrum sources (Greene et al. [1994]). The gas disk has dissipated in class III disks and $\alpha_{IR} < -1.6$. The disk categories correspond to time evolution (Adams et al. [1987]). Planet formation is believed to occur mainly in class II disks but may also start at earlier phases. For this reason, class II disks are of special interest in the context of this thesis and are referred to as just disks in the following.
Although it is difficult to determine the lifetime of disks, it is clear that their lifetime is finite. It is inferred to be in the order of 10^6 years (Kimura et al. [2016]). From the observed gas-accretion rates of the central stellar objects (Hartmann et al. [1998], Ingleby et al. [2014], Manara et al. [2016]), it is known, that gas has to be transported from outer disk regions to inner ones and finally onto the central star. The transport of angular momentum is essential to explain the observed mass accretion rates onto young stars from their PPDs. As first proposed by von Weizsäcker [1946], vortices in PPDs cause turbulent motion which in turn results in turbulent viscosity through Reynolds stresses (Shakura and Sunyaev [1973]). This turbulent viscosity can be expressed with the disk height H, the speed of sound c_s and the dimensionless α-parameter as

$$\nu_{turbulent} = \alpha c_s H. \tag{1.2}$$

In the absence of considerable molecular viscosity[1], this turbulent viscosity moderates angular momentum transport from inner disc regions to outer ones (Lynden-Bell and Pringle [1974]). Additionally, vortices in PPDs are anti-cyclonic (Adams and Watkins [1995]) and thus contain pressure maxima at their respective centers. These pressure maxima act as dust-traps (Whipple [1972]), where the local dust density is further augmented by streaming instability (Youdin and Goodman [2005], Johansen et al. [2006a]), may exceed a critical value and subsequently cause gravitational collapse to take over (Toomre [1964], Safronov [1972], Goldreich and Ward [1973], Johansen and Youdin [2007]). However, if this process pipeline is the (only) route to the formation of planets, is not yet settled.

1.1.2 Gas Equilibrium

Broadly speaking, PPDs are circular structures around young stars, flattened by angular momentum conservation. They contain gas and dust, both in concentrations that allow for modeling through the equations of hydrodynamics[2]. An approximation of the mass contained in a PPD can be obtained by the minimum mass solar nebula model (MMSN, Weidenschilling [1977]), where the mass of the solar system objects is distributed in the radial direction to obtain a surface density for the dust material. Using solar metallicity, a radial gas surface density profile can be inferred. The obtained surface density value is $\Sigma(R) = 1700 \left(\frac{R}{AU}\right)^{-3/2} \frac{\mathrm{g}}{\mathrm{cm}^2}$ (Hayashi [1981]). The dynamics of the gas are governed by the conservation laws of mass, momentum, and energy. These manifest in the Navier-Stokes- or, in the non-viscous case, the Euler-Equations

$$\partial_t \rho + \nabla \cdot (\rho \vec{v}) = 0 \tag{1.3}$$

$$\partial_t (\rho \vec{v}) + \nabla \cdot (\rho \vec{v} \otimes \vec{v}) = -\nabla P - \rho \nabla \Phi_{grav} + \nabla \cdot \hat{\vec{\tau}} + \mathcal{S} \tag{1.4}$$

$$\partial_t E + \nabla \cdot [(E + P)\vec{v}] = -\rho \vec{v} \cdot \nabla \Phi_{grav} + \nabla \cdot (\hat{\vec{\tau}} \cdot \vec{v}) + \mathcal{S}^*. \tag{1.5}$$

Here, ρ is the (gas) density, $\vec{v}$ the velocity vector, P the pressure, E the total specific energy $\hat{\vec{\tau}}$ the viscous stress tensor, Φ_{grav} the gravitational potential of the central stellar object only, and $\mathcal{S}$ and $\mathcal{S}^*$ are further source terms due to e.g radiative

[1] The molecular viscosity can be obtained from the mean-free path (l_{fp}) of molecules and the speed of sound as $\nu_{molecular} \sim l_{fp} c_s$. For the expected densities and temperatures in PPDs, this yields viscous time-scales $\tau_\nu = L^2/\nu$ of several Hubble-times (using L in the order of Astronomical Units). The lifetime of PPDs is with a value the order of several million years much shorter.

[2] For gas parcels the expected mean-free path is in the order of 1 m, compared to the disc height, given by the pressure scale-height in the order of 10^{11} cm (values taken from Manger [2019]).

heating/ cooling or thermal conduction. Self-gravitating effects are neglected, as justified by the MMSN model, which yields a disk-to-star mass ratio of $10^{-2} - 10^{-1}$. In a stable solution to Equations (1.3) - (1.5), usually the source terms $\mathcal{S}$ and $\mathcal{S}^*$ as well as viscosity are neglected. Time derivatives vanish, and the transport of mass, momentum, and energy is set to be divergence-free. In this case, the momentum Equation (1.4) reduces to equality of pressure- and gravitational potential gradients as well as geometric terms giving the centrifugal- and Coriolis forces in spherical coordinates or only the centrifugal force in cylindrical coordinates.
Due to the thin geometry of disks, the latter is usually chosen. It brings the problem to

$$R\Omega^2 - \frac{GM_\star R}{(R^2+z^2)^{3/2}} - \frac{1}{\rho}\frac{\partial P}{\partial R} = 0 \tag{1.6}$$

$$-\frac{GM_\star z}{(R^2+z^2)^{3/2}} - \frac{1}{\rho}\frac{\partial P}{\partial z} = 0. \tag{1.7}$$

This can only be solved when the system is closed with an equation of state. Often used are the ideal gas law $P = (\gamma - 1)\rho e$ with the adiabatic index γ and specific internal energy e as well as the isothermal equation of state $P = \rho c_s^2$ with isothermal sound speed $c_s^2 = \mathcal{R}T/\mu$. Here, $\mathcal{R}$ is the gas constant, and μ is the mean molecular weight. Assuming an isothermal equation of state as well as $c_s \neq c_s(z)$ and solving in the vertical, one obtains the density profile from Equation (1.7) as

$$\rho(R,z) = \rho(R)e^{-\frac{z^2}{2h^2}}, \tag{1.8}$$

where $h = \frac{c_s}{\Omega_K}$ is the vertical pressure scale-height and $\Omega_K = \sqrt{\frac{GM_\star}{R^3}}$ the Keplerian rotation frequency (Bell et al. [1997]).
Equation (1.6) for the radial direction can be solved for Ω in dependence of $\rho(R)$ (or $P(R)$). It is common to assume a power-law shape in the cylindrical radial direction for the pressure, such that $P = P_0 \left(\frac{R}{R_0}\right)^n$ near a reference radius R_0 where $P = P_0$. This is motivated by the shape of the surface density profile obtained by the MMSN with $\Sigma(R) = \int_{-\infty}^{\infty} \rho(R,z)dz$. The result is a rotation frequency dependent on n and slightly below the Keplerian value.

1.1.3 Dust Dynamics

The dynamic of dust is critical to the understanding of protoplanetary disks and is given here for completeness and as motivation for further considerations. The hydrodynamic equations expressing the coupled system of gas and dust with gas density ρ_g, dust density ρ_d, gas velocity $\vec{v}_g$, dust velocity $\vec{v}_d$, central gravitational potential Φ_{grav}, as well as gas pressure P_g, friction time τ_s are

$$\partial_t \rho_g + \nabla \cdot (\rho_g \vec{v}_g) = 0 \tag{1.9}$$

$$\partial_t \rho_d + \nabla \cdot (\rho_d \vec{v}_d) = 0 \tag{1.10}$$

$$\partial_t (\rho_g \vec{v}_g) + \nabla \cdot (\rho_g \vec{v}_g \otimes \vec{v}_g) = -\nabla P_g - \rho_g \nabla \Phi_{grav} + \rho_d \frac{\vec{v}_d - \vec{v}_g}{\tau_s} \tag{1.11}$$

$$\partial_t (\rho_d \vec{v}_d) + \nabla \cdot (\rho_d \vec{v}_d \otimes \vec{v}_d) = -\rho_d \nabla \Phi_{grav} - \rho_d \frac{\vec{v}_d - \vec{v}_g}{\tau_s}. \tag{1.12}$$

The friction time (or stopping time) τ_s is governed by the friction between dust particles and gas. It depends on the dust grain size ratio by means of its radius a (assuming spherical grains) to the mean-free path of the gas λ_{free}. If this ratio is smaller than 9/4 the particle is said to be in the *Epstein* regime (Epstein [1924]). For larger ratios, the particle is in the *Stokes* regime (Stokes [1851]). Differentiating between the two regimes, the stopping time is

$$\tau_s = \begin{cases} \frac{\rho_s a}{\rho_g v_{th}} & , \frac{a}{\lambda_{free}} \leq \frac{9}{4} \text{ (Epstein)} \\ \frac{4a}{9\lambda_{free}} \frac{\rho_s a}{\rho_g v_{th}} & , \frac{a}{\lambda_{free}} > \frac{9}{4} \text{ (Stokes)} \end{cases}. \tag{1.13}$$

Here ρ_s is the density of the dust grain and $v_{th} \sim c_s$ is the mean thermal velocity. From the stopping time and the rotation frequency a useful quantity, namely the stokes number

$$St = \tau_s \Omega, \tag{1.14}$$

describing the degree of coupling between gas and dust can be defined.

Radial and Azimuthal Motion

Famously, Nakagawa et al. [1986] solved the system of Equations (1.9) - (1.12) in the regime of small Stokes numbers to find a steady state solution for gas- and dust velocities in cylindrical coordinates which, according to Youdin and Johansen [2007], can be expressed as

$$v_{g,R} = \frac{2\epsilon\tau_s}{(1+\epsilon)^2+\tau_s^2}\eta v_K, \tag{1.15}$$

$$v_{g,\varphi} = v_K - \left(1 + \frac{\epsilon\tau_s^2}{(1+\epsilon)^2+\tau_s^2}\right)\frac{\eta}{1+\epsilon}v_K, \tag{1.16}$$

$$v_{d,R} = -\frac{2\tau_s}{(1+\epsilon)^2+\tau_s^2}\eta v_K, \tag{1.17}$$

$$v_{d,\varphi} = v_K - \left(1 - \frac{\tau_s^2}{(1+\epsilon)^2+\tau_s^2}\right)\frac{\eta}{1+\epsilon}v_K, \tag{1.18}$$

where $\epsilon = \frac{\rho_d}{\rho_g}$ is the ratio of dust density to gas density, $v_K = \Omega_K R = \sqrt{\frac{GM}{R}}$ is the Keplerian rotation velocity and $\eta \sim (c_s/v_K)^2$ a dimensionless measure for rotation. From this solution, it becomes obvious that in the azimuthal direction, gas and dust neither move at the Keplerian value nor at the same speed. Given the fact that Equation (1.11) contains the pressure gradient in addition to gravitational- and centrifugal forces whilst Equation (1.12) does not, neither comes as a surprise when one assumes a decrease in pressure with radius. Due to the outward pressure gradient, the gas moves at sub-Keplerian velocities. The dust, on the other hand, would be stable on Keplerian orbits but does feel a headwind from the slower gas. Because of this, the dust moves closer in and again feels the headwind. The result is a consistently negative radial velocity for the dust in Equation (1.17). The radial velocity obtained for the gas is positive due to the acceleration from the dust (see Weidenschilling [1977]). As a result of the solution, all dust would fall onto the central star, while gas is transported away from it. According to Brauer et al. [2007] the effect peaks at $St = 1$ and decreases for lower as well as higher values. For reasonable disk parameters, Armitage [2020] calculates an example drift time for grain sizes of $\sim$ 20 cm in the order of 10^3 years at 5 AU, which is very short compared to the disk lifetime.
As a consequence of this rapid inflow, also known as radial drift barrier, the formation of planetesimals, which are of kilometer sizes, must take place on small time scales as well.

Vertical Motion

As pointed out in Birnstiel et al. [2016], dust moves towards higher gas pressure, which is also true in the vertical direction, where the highest gas pressure is, as outlined in Section 1.1.2, inferred to be in the mid-plane. The vertical dust velocity can be expressed as

$$v_{d,z} = -z\Omega St. \tag{1.19}$$

Similar to the radial and azimuthal velocity components, also the vertical one

depends on particle size through the Stokes number. Again, for reasonable disk parameters, Armitage [2020] gives an example for settling at 1 AU of 1.5×10^5 years, which is much shorter than the lifetime of the disk but larger than the corresponding time-scale for radial drift.

Dust Growth

In order to get from below μm sized dust particles to planets that are thousands of kilometers in diameter, many orders of magnitude must be overcome. Intuitively, some particles i and j need to collide in order to form a bigger particle. Independent of their size, the collisional time scale can be defined as

$$\tau_{\text{col}} = \frac{1}{n_j \sigma_{ij} \Delta v_{ij}}, \tag{1.20}$$

where n_j is the number density of particles j, σ_{ij} is the mutual cross-section and Δv_{ij} the relative velocity of i and j particles (Birnstiel et al. [2016]). As seen above, the velocity difference between dust species i and j due to sedimentation and radial drift depends on the difference in their Stokes numbers. In addition to those effects, Brownian motion and turbulence (Voelk et al. [1980], Ormel and Cuzzi [2007]) can also contribute to the relative velocity. While Brownian motion contributes significantly only for the smallest particles, the velocity difference due to turbulent motion as described by Ormel and Cuzzi [2007] peaks at $St = 1$. For larger and larger particles, radial drift and, finally, azimuthal drift become dominant (Birnstiel et al. [2011]).
The different outcomes from the collision of two particles are mainly sticking, bouncing, erosion, mass transfer, and fragmentation (Birnstiel et al. [2016]). Windmark et al. [2012a] and Windmark et al. [2012b] modeled the collisional outcome for different particle sizes with the result that masses add up most efficiently for smaller particles below sizes of 1 mm, independent of the size ratio. If both grains have comparable sizes, fragmentation starts at the cm-scale. At the meter and lower-kilometer scale, erosion and fragmentation dominate, and mass transfer is only possible from much smaller particles (see Blum [2018] for a review of laboratory experiments). This fact is also known as the fragmentation barrier. In combination with the radial drift barrier, it forms the meter-size barrier. The name "meter-size barrier" is somewhat misleading since this barrier covers decimeter to kilometer scales. Dust grains with sizes right below this barrier are termed pebbles.

1.1.4 Instabilities in Protoplanetary Disks

As sketched above, a PPD with laminar gas flow is relatively simple to describe mathematically in cylindrical coordinates. The picture becomes more complicated when dust is brought into the equation. Naturally, dust interacts with the gas, settles towards the mid-plane, and drifts towards the star, while the gas does not. Initially, it was believed that settling to the mid-plane increases dust density enough to form planetesimals through direct gravitational collapse (Safronov [1972]), but it was shown by Weidenschilling [1980] that this is not the case. Additionally, laminar gas tends to flow away from the star rather than towards it. As mentioned above, turbulent motion provides the possibility to transport angular momentum from inner to outer disk regions. The so-called route to turbulence is a long-standing problem in disk theory. Several mechanisms have been found which are capable of inducing turbulence in PPDs.
Below, a brief overview of subjectively selected mechanisms that are capable of producing vortices in the gas phase and those further concentrating dust until it is dense enough to collapse gravitationally, as well as the mechanism behind gravitational collapse itself, is given. For more details see e.g Klahr et al. [2018], Lyra and Umurhan [2019], Birnstiel et al. [2016] and Schreiber [2018] (and references therein).

Magnetorotational Instability

Although the importance of turbulent viscosity has been known, no unique mechanism to produce the required turbulence has been identified for a long time. A milestone was the discovery of the Magnetorotational Instability (MRI, Balbus and Hawley [1991]) which is capable of producing strong levels of turbulence[3]. As the name indicates, The MRI requires a magnetic field, interacting with charged particles in a differentially rotating disk. The instability is driven by the mutual angular momentum transfer between paired particles through the magnetic field in such a way that they radially deviate.
Although only a small ionization fraction $\chi \sim 10^{-12}$ suffices for the MRI to operate before Ohmic diffusion quenches the effect (Armitage and Kley [2019]), it is found that Ohmic diffusion, ambipolar diffusion, and the Hall effect result in so-called dead zones in the disk beyond $\sim$ 10 AU where the MRI is not active (Gammie [1996], Lesur et al. [2014]). In these dead zones (or Ohmic zones), hydrodynamic effects must be operating to provoke the necessary turbulence levels.

[3] In terms of the resulting α-viscosity (Shakura and Sunyaev [1973]), which is found as $\alpha \sim 10^{-2}$ (see Equation (1.2)).

Instabilities in Gas

The Rayleigh criterion is essential for understanding the stability of PPDs. It is the criterion of stability for a radially unstratified and differentially rotating cylinder. Following the derivation in Lyra and Umurhan [2019], the epicyclic frequency under these circumstances is

$$\kappa_{ep} = \Omega\sqrt{2(2-q)}, \tag{1.21}$$

where $q = -\frac{\partial \log \Omega}{\partial \log R}$. In case $q > 2$, κ_{ep} becomes complex and the flow unstable. In the case of Keplerian flow, $q = 1.5$, and the flow is stable. More often, the criterion for stability is expressed as a radially (cylindrical) outward increase of angular momentum $\frac{\partial L}{\partial R} > 0$.
When stratification is considered, buoyancy enters the equation in the form of the Brunt-Väisälä frequencies $\mathcal{N}_i^2 = -\frac{1}{\rho c_p}\partial_i P \partial_i S$ with $i \in R, z$, and heat capacity c_p to give (Rüdiger et al. [2002]) the Solberg-Høiland criteria

$$\mathcal{N}_z^2 + \mathcal{N}_R^2 + \kappa_{ep}^2 > 0, \tag{1.22}$$

$$\frac{\partial P}{\partial z}\left(\frac{\partial S}{\partial z}\frac{\partial L^2}{\partial R} - \frac{\partial S}{\partial R}\frac{\partial L^2}{\partial z}\right) < 0. \tag{1.23}$$

Here P is pressure and S entropy. According to Lyra and Umurhan [2019], these conditions also imply $\mathcal{N}_z^2 > 0$ and $\mathcal{N}_R^2 + \kappa_{ep}^2 > 0$, independently. The former is the Schwarzschild criterion, and the latter is the Rayleigh criterion extended with a radially buoyant contribution, which can be stabilizing or destabilizing as in the convective overstability (COS, Klahr and Hubbard [2014]), where thermal relaxation of a deflected parcel in a radial entropy gradient buoyantly amplifies epicyclic oscillations. The COS shows a resemblance with the subcritical baroclinic instability (SBI, Lesur and Papaloizou [2010]), which likewise depends on a convectively unstable stratification and thermal relaxation (Lyra [2014]). A disk, stable to the Solber Høiland criteria, can still be unstable to the SBI or the COS when cooling is brought into the equation.
Similarly, short cooling times are required for the vertical shear instability (VSI, Urpin and Brandenburg [1998]) to operate. In contrast to the COS, the VSI requires radial and vertical stratification, hence baroclinicity. From the vorticity equation, it can be derived that baroclinicity in a rotating system generates a thermal wind. In the vertical direction, this is tantamount to a variation in the rotation frequency as $R\partial_z\Omega^2 = \frac{(\nabla\rho\times\nabla P)_\varphi}{\rho^2}$, aka vertical shear. The VSI mechanism can best be understood when first the case without vertical shear is considered: a vertically displaced fluid parcel in such a (barotropic) environment, would, in the isothermal case, drift with constant velocity through the fluid and, with an adiabatic component, perform epicyclic oscillations in the radial direction - or analog oscillations in the vertical

- around its equilibrium state, radiate its kinetic energy away and finally settle at the initial state. In this case, no instability arises. In the presence of vertical shear, however, there is a narrow zone of instability between the neutral cases of vertical and the parabolic-shaped contour of constant angular momentum. A displacement into this region means gas enters a region of lesser angular momentum and will continue accelerating as the centrifugal disbalance continuously increases with its motion. This is a special violation of Rayleigh's criteria in Equations (1.21) and (1.22).
In contrast to the COS, which is most efficient at cooling times in the order of the orbital time-scale (Lyra [2014]), the VSI is most efficient in the isothermal case, since buoyancy acts in a purely stabilizing way (see Equation (1.22)). A critical value for the cooling time above which the VSI does not operate anymore has been assessed, e.g., by Lin and Youdin [2015] and Manger et al. [2021] as $\tau_{crit} \lesssim \frac{|\partial_z(R\Omega)|}{\mathcal{N}^2}$. Historically, the mechanism behind the VSI was first described by Goldreich and Schubert [1967] and Fricke [1968] in the context of differentially rotating stellar interiors. It was hence termed the Goldreich-Schubert-Fricke instability (GSF). The equivalence between VSI and GSF was first noted by Urpin [2003].
Since GSF and VSI are mathematically equivalent, there is no universally accepted distinction between the two. However, it makes sense to make a distinction since the spherical geometry of stars is very different from the approximately cylindrical disks. Together with the mechanism, responsible for cooling, this will give the distinction made in this thesis. In contrast to VSI, GSF has - at the time of writing this thesis - never been modeled using a global approach. This may be because the weakly compressible conditions in stellar interiors require special modeling techniques, or because initial conditions in spherical geometry are difficult to find. This thesis presents the first global GSF simulations in Chapter 3. Despite its relevance for stars, which is also briefly outlined in that Chapter, VSI in spherical geometry (GSF) may be of interest in disks to show that even when the assumption of cylindrical geometry does not hold, the instability can operate.

Streaming Instability

As outlined above, depending on Stokes number and dust to gas ratio, dust in PPDs not only drifts on rather short time-scales towards the central star but also tends to move to the disk mid-plane[4] and accumulates in the central pressure maxima of vortices in the gas. Once the dust-to-gas ratio is high enough - not too far from order unity - a process termed streaming instability (SI, Youdin and Goodman [2005], Johansen and Youdin [2007], Youdin and Johansen [2007]) sets in. The SI further increases the local dust density. If the dust becomes dense enough so it can collapse under its own self-gravity in this SI-moderated environment, is dependent on the strength of coupling to the gas as well as the shear counteracting collapse.

[4]Settling of dust to the mid-plane increases the dust to gas ratio at the mid-plane and, according to Equation (1.16) accelerates the gas there relative to higher regions. As a result, Kelvin-Helmholtz instability (KHI) occurs and re-mixes the dust. Consequently, dust can accumulate in the pressure maxima of KHI vortices, causing the same scenario as with other dust traps.

Gravitational Instability

When external masses, or in other words shearing forces, are *swindled* away, one obtains the well-known Jeans instability (Jeans [1902]), which states, that gas clouds larger than the Jeans-length $\lambda_{Jeans} = c_s\sqrt{\pi/G\rho}$ are unstable against gravitational collapse. When external masses are considered, rotation and angular momentum conservation become important. Although analytical derivations of the corresponding stability criterion for the case of uniform rotation were derived earlier (e.g. Safronov [1960] or Chandrasekhar [1961]), it is attributed to Toomre [1964], who was the first to consider Keplerian differential rotation. The corresponding instability is hence named after him as Toomre instability. The criterion for instability is usually expressed in terms of the Toomre parameter Q as

$$Q = \frac{\Omega c_s}{\pi G \Sigma} \overset{!}{<} 1. \tag{1.24}$$

Since the derivation of the stability criterion is done in the thin-disk limit, the Q-parameter encompasses the surface density Σ rather than the volume density. Again, c_s is the speed of sound, G is the gravitational constant and Ω as angular frequency is a proxy for the epicyclic frequency (see Equation (1.21)). This Toomre value indicates whether there can be a range of axisymmetric unstable perturbation for the radial wave numbers k. Large wave numbers (small scales) are stabilized by the pressure (Jeans-Criterion), and small wave numbers (large scales) by the angular momentum conservation (Rayleigh Criterion). The most unstable wave-number is $k_{crit} = \frac{\pi G\Sigma}{c_s^2}$, and the corresponding wavelength $\lambda_{crit} = \frac{2c_s^2}{G\Sigma}$. Intuitively, the speed of sound, and with it the pressure, opposes the density and shifts the most unstable cloud size to larger values.
As the Jeans-instability criterion, Toomre's criterion is valid for gases. A stability criterion for dust can be approximated when the sound speed in the dispersion relation obtained by Toomre is replaced with a characteristic velocity of dust (e.g. Safronov [1972], Goldreich and Ward [1973]). The corresponding dispersion is the same as obtained by Toomre but with gas sound speed replaced by "dust sound speed". It reads

$$\omega^2 = c_{dust}^2 k^2 - 2\pi G\Sigma|k| + \Omega^2 \overset{!}{<} 0. \tag{1.25}$$

In the absence of random dust motion, $c_{dust} = 0$, Goldreich and Ward [1973] compute the critical wavelength related to collapse in the axisymmetric case as $\lambda_c = \frac{4\pi G\Sigma}{\Omega^2}$. There is a close relation between the radius at which collapse takes place for a given, uniform density and the sphere of gravitational dominance a solid body of the corresponding mass would have in the gravitational field of a shearing mass. The latter is termed Hill sphere, and the corresponding radius and density are Hill radius and Hill density. The Hill radius is

$$r_{Hill} = d \times \sqrt[3]{\frac{m}{3M_\star}} \approx 0.69 \times d \times \sqrt[3]{\frac{m}{M_\star}}. \tag{1.26}$$

Analogously, the Hill density is defined as the uniform density corresponding to mass m as $\rho_{Hill} = \frac{9}{4\pi}\frac{M_\star}{d^3}$ (the definition used in Klahr and Schreiber [2020]). One might assume that collapse is certain once dust or pebbles locally reach this density. Klahr and Schreiber [2021], however, show that this is not the case. The reason is that shearing forces and the inherent random motion of dust and pebbles, as well as coupling to the gas, play an opposing role in the collapse. Klahr and Schreiber [2020] argue that at the scales relevant for collapse, diffusion caused by coupling to the gas is dominant and use a timescale argument to derive a critical length scale

$$l_c = \frac{1}{3}\sqrt{\frac{\delta}{St}}H. \tag{1.27}$$

Combined with the Hill density, this gives a second criterion that a pebble cloud embedded in gas must fulfill. Here, St is again the Stokes number, $H = c_s/\Omega$ the vertical pressure scale height, and $\delta = D/Hc_s =$ the dimensionless diffusivity, with underlying turbulent diffusion D. Combining the criteria, a minimal mass for a collapsing cloud at Hill density can be derived as $m = \frac{4\pi}{3}l_c^3\rho_{Hill}$. Klahr and Schreiber [2021] also deduce a formal pebble speed of sound, which is defined by the ratio of diffusivity to the stopping time, and give it as

$$a \approx \sqrt{\frac{\delta}{\delta + St}}c_s \approx \sqrt{\frac{\delta}{St}}c_s. \tag{1.28}$$

With this expression for the sound speed, the original dispersion relation in Equation (1.25) can be used to derive the stability criterion for pebbles as

$$Q_{dust} = \sqrt{\frac{\delta}{St}}\frac{Q}{Z} \overset{!}{<} 1. \tag{1.29}$$

Here, Q is Tommre's parameter for gas and $Z = \Sigma_{gas}/\Sigma_{dust}$ the metallicity (Klahr and Schreiber [2021]). The diffusion-limited collapse described by this criterion is of special importance in the context of this thesis since it gives the context of simulations analyzed in Chapter 4.

1.1.5 Planetesimal Formation

As the smallest objects, bound by self-gravity - thus objects that have just overcome the meter-size barrier (see Section 1.1.3, "Dust Growth") - understanding the formation of planetesimals is crucial to understanding the formation process of planets. By nature, they are small and can today be observed only in our own solar system in the form of asteroids and Kuiper-belt objects. Observations of primordial Asteroids (Delbo' et al. [2017], Delbo et al. [2019]) and Kuiper belt objects (Johnston [2019]) show that there exists a preferred size scale on which these objects form. This preferred scale can be used as a benchmark for the modeling capacities of numerical simulations describing the formation of the corresponding planetesimals. Historically, the size distribution of planetesimals is assumed to follow a power-law distribution (Morbidelli and Vokrouhlický [2003], Johansen et al. [2015], Simon et al. [2017], Schäfer et al. [2017]). However, numerical simulations by Polak and Klahr [2023] reproduce a preferred mass scale and find the mass distribution of objects around this scale to follow a Gaussian profile, at least at 1 AU distance from a solar mass star.

1.2 Explicit and Implicit Method

Although analytical methods are still a vital part of theoretical astronomy, numerical methods have become increasingly indispensable, and oftentimes, computational resources determine the pace at which theory can progress[5]. Because of this, access to efficient computer codes dedicated to solving the problem at hand is essential for theoretical astronomers. There are many classes of problems in modern astrophysics that one has to tackle computationally; hydrodynamic ones, which generally require the solution of the Navier-Stokes-Equations (NSE), are of special importance. Fundamentally, the NSE express a fluid's evolution through temporal and spatial derivatives, expressing mass, momentum, and energy conservation. Mathematically, they give the set of non-linear Partial Differential Equations (PDEs), defined in Equations (1.3)-(1.5). In order to make the NSE solvable on a computer, they have to be discretized in space and time.
Below, a brief comparison of the time explicit and implicit methods is given. This acts as an introduction to the main project within this thesis, the development of a globally implicit hydrodynamics solver. More details on hydrodynamic methods can be found in numerous textbooks with a focus on astrophysical applications to engineering ones and, finally, applied mathematics, e.g. Bodenheimer et al. [2006], Ferziger and Perić [2002], Chung [2002], Toro [2009] and LeVeque [2002].

[5] A historical example of this is the discussion on the interaction of pebbles with gas and the turbulence within. It was unclear whether pebbles damp-, or trigger turbulence in the gas and if direct gravitational collapse of pebbles in disks is possible. Finally, a series of numerical studies (Johansen et al. [2006a], Johansen et al. [2006b]) showed that turbulence leads to concentration on certain scales.

Spatial Discretization

Different techniques exist and are actively used in astrophysics for the spatial discretization of the NSE. The Finite Difference method (FDM), Finite Volume method (FVM), and Finite Element Method (FEM), as well as spectral type methods, are grid-based methods - also named *Eulerian* - methods. Each of these methods has different strengths and weaknesses, be it in the spatial order recoverable with the method or the complexity of the domain model to their conservation properties. As a rule of thumb, one might say that methods built on the differential (or strong) form of the NSE, that is, the FDM as well as (most) spectral methods, are especially useful where a high spatial convergence order is desirable. On the other hand, methods built on the integral (or weak) form of the NSE, the FVM, and FEM, are most useful when complex geometries or boundary conditions are needed.
The second class of spatial methods is given by the *Lagrangian* or grid-free methods. Here, the trajectories of simulation particles, the so-called super-particles, are traced over time. The most prominent Lagrangian method is the Smoothed Particle Hydrodynamics (SPH) method. It is especially useful where a sharp definition of boundaries is not meaningful and/ or a large range of spatial scales has to be resolved. There are also codes mixing SPH and FVM or FEM methods like, e.g., the GIZMO code (Hopkins [2015]) or the AREPO code (Springel [2010]). The GIZMO code is used to conduct the simulations analyzed in Chapter 4 of this thesis.

Temporal Discretization Approaches

There are two fundamental approaches to the temporal discretization of the NSE. The differences between them can best be seen in the example of the linear one-dimensional first-order PDE, also known as the transport equation

$$\partial_t q + v \partial_x q = 0, \tag{1.30}$$

with some scalar variable q, as well as time independent velocity v in x direction. When the considered domain is divided into cells of equal finite width Δx, one can use the FDM with the first order upwinding scheme, also called the donor-cell scheme, to express Equation (1.30) in semi-discrete form for some cell j inside the domain as

$$0 = \partial_t q_j + v_j \begin{cases} \frac{q_j - q_{j-1}}{\Delta x} & , v_j > 0, \\ \frac{q_{j+1} - q_j}{\Delta x}, & , v_j < 0 \\ 0 & , \text{else} \end{cases} . \tag{1.31}$$

Here, the upwinding scheme favors the direction from which information arrives at the considered cell.
In tight analogy, the gradient in temporal direction can be discretized for some

time-step n and time difference Δt, where the analog scheme is called Euler-scheme, and the direction of information propagation is fixed in positive t-direction $\partial_t q = (q^n - q^{n-1})\Delta t$. It is a convention to write this as $\partial_t q = (q^{n+1} - q^n)\Delta t$, to illustrate that information at n is known and that at $n+1$ has to be found.

1.2.1 Numerical Stability Example

In order to continue with the temporal discretization of the transport operator, a *support point* (SP, Stützstelle) needs to be selected. This can be given either by the present (known) time step or by the next (unknown) time step. In the former case, one obtains the so-called explicit method, the *forward Euler method*, which (now considering positive velocities only) leads to

$$\frac{q_j^{n+1} - q_j^n}{\Delta t} + v_j \frac{q_j^n - q_{j-1}^n}{\Delta x} = 0 \tag{1.32}$$

$$\Leftrightarrow q_j^{n+1} = q_j^n - \Delta t \times v_j \frac{q_j^n - q_{j-1}^n}{\Delta x}. \tag{1.33}$$

Since all pieces of information at the last time-step are known, the algebraic equation for q^{n+1} can be readily solved.
When the unknown time-step is chosen as SP one obtains the *backward Euler scheme* with the expression

$$\frac{q_j^{n+1} - q_j^n}{\Delta t} + v_j \frac{q_j^{n+1} - q_{j-1}^{n+1}}{\Delta x} = 0 \tag{1.34}$$

$$\Leftrightarrow \begin{pmatrix} 1 + \frac{v_j \Delta t}{\Delta x} \\ -\frac{v_j \Delta t}{\Delta x} \end{pmatrix} \begin{pmatrix} q_j^{n+1} \\ q_{j-1}^{n+1} \end{pmatrix} = q_j^n. \tag{1.35}$$

Since this can not be solved for q_j^{n+1} without knowledge about q_{j-1}^{n+1}, one has to formulate the whole considered domain as a linear algebraic system to find a solution as

$$\underbrace{\begin{pmatrix} \ddots & -c & & \\ 0 & 1+c & -c & \\ & 0 & 1+c & \dots \\ & & & \ddots \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} \vdots \\ q_j^{n+1} \\ q_{j-1}^{n+1} \\ \vdots \end{pmatrix}}_{\vec{q}^{n+1}} = \underbrace{\begin{pmatrix} \vdots \\ q_j^n \\ q_{j-1}^n \\ \vdots \end{pmatrix}}_{\vec{q}^n}, \tag{1.36}$$

with $c = v_j \frac{\Delta t}{\Delta x}$. This approach is termed *implicit* since the solution vector is given implicitly by $A\vec{q}^{n+1} = \vec{q}^n$. For an intuitive comparison, the explicit Equation (1.33) can be rewritten as

$$\vec{q}^{n+1} = \underbrace{\begin{pmatrix} \ddots & c & & \\ 0 & 1-c & c & \\ & 0 & 1-c & \dots \\ & & & \ddots \end{pmatrix}}_{B} \vec{q}^{n}. \tag{1.37}$$

As mentioned before, Equations (1.36) and (1.37) hold for positive v_j. For the case of negative v_j, one finds equivalent expressions with the entries of sub- and super-diagonal switched and a change in sign for all c (c itself also becomes negative). This makes it obvious that A is a non-singular matrix and is, hence, invertible in both cases. For B, one can infer the condition, that all eigenvalues have to be smaller than unity, since otherwise the solution - and errors with it - would grow over time (Smith [1978]). This is the case for B only when $c < 1$. One can show that, indeed, for $c > 1$, Equation (1.37) has no unique solution other than the trivial one. For this reason, implicit methods are unconditionally stable, while explicit methods are stable under the condition $c < 1$, where c is known as the Courant- or Courant-Friedrichs-Lewy (CFL) number. In the literature, the term "CFL number" is often used in the context of the associated stability condition, and the term "Courant" number is synonymous with a safety factor limiting the CFL number to a certain value below one. This distinction is not made in the following and the term Courant number, as used in this thesis, is always simply the result of the expression $C = v\frac{\Delta t}{\Delta x}$. A more rigorous analysis of (un-) conditional stability is done with the von Neumann stability analysis, which reveals exponential growth of amplitudes in the explicit case for c larger unity.

1.2.2 Numerical Dissipation Example

Making the Ansatz of a discrete plane wave $q_j^n \sim e^{i(k\Delta x j - \omega \Delta t n)}$ (for the explicit part following Durran [2010]), similar to the von Neumann stability analysis, one can write for Equation (1.33)

$$e^{i(k\Delta x j - \omega\Delta t(n+1))} + (c-1)e^{i(k\Delta x j - \omega\Delta t n)} - ce^{i(k\Delta x(j-1) - \omega\Delta t n)} = 0, \tag{1.38}$$

$$\Leftrightarrow e^{-i\omega\Delta t} = 1 - c(1 - e^{-ik\Delta x}), \tag{1.39}$$

$$\Leftrightarrow |A|^n e^{-i\omega_r \Delta t} = 1 - c(1 - e^{-ik\Delta x}). \tag{1.40}$$

Here the substitution $|A|^n = e^{\omega_i n \Delta t}$ with $\omega = \omega_r + i\omega_i$ has been made in the last step such that the sign of $|A|^n - 1$ decides if errors are damped over time or grow exponentially. Splitting this into real- and complex parts and using Euler's identity, gives

$$|A|^n \cos(\omega_r \Delta t) = 1 - c(1 - \cos(k\Delta x)) \quad \text{Real part,} \tag{1.41}$$

$$|A|^n \sin(\omega_r \Delta t) = c \sin(k\Delta x) \quad \text{Imaginary part.} \tag{1.42}$$

Squaring both parts and adding yields

$$(|A|^n)^2 = 1 - 2c(1-c)\underbrace{(1 - \cos(k\Delta x))}_{0 \leq \ldots \leq 1}. \tag{1.43}$$

Under the assumption of a non-complex solution, this also yields the Courant criterion $c_{max} = 1$. The last term on the right-hand side can only take values smaller than unity since $k\Delta x = 2\pi\Delta x/\lambda$ and $0 < \Delta x < 2\lambda$ is the resolution limit. Additionally, one can see that the right-hand side and thus the damping of amplitudes is minimal for $c = 1$ and in the allowed range maximal for $c = 0.5$. Below this value for the Courant number, the damping decreases again. This damping is known as numerical diffusion.
Conducting the same analysis for the implicit expression in Equation (1.34), one obtains

$$(1+c)e^{i(k\Delta x j - \omega\Delta t(n+1))} - e^{i(k\Delta x j - \omega\Delta t n)} - ce^{i(k\Delta x(j-1) - \omega\Delta t(n+1))} = 0, \tag{1.44}$$

$$\Leftrightarrow (1 + c - ce^{-ik\Delta x})e^{-i\omega\Delta t} = 1, \tag{1.45}$$

$$\Leftrightarrow |A|^n (1 + c - ce^{-ik\Delta x})e^{-i\omega_r \Delta t} = 1. \tag{1.46}$$

Separating in real and complex parts as before gives

$$|A|^n \left[(1 + c - c\cos(k\Delta x))\cos(\omega_r \Delta t) - c\sin(k\Delta x)\sin(\omega_r \Delta t)\right] = 1, \tag{1.47}$$
$$|A|^n \left[(1 + c - c\cos(k\Delta x))\sin(\omega_r \Delta t) - c\sin(k\Delta x)\cos(\omega_r \Delta t)\right] = 0. \tag{1.48}$$

Squaring and adding the components leads to

$$(|A|^n)^2 = \frac{1}{1 + 2c(1 + c)(1 - \cos(\Delta x k))}. \tag{1.49}$$

As mentioned before, this never exceeds unity for arbitrary c. In addition to this, one can see that $|A|^n$ approaches its optimal value of one for $c \to 0$. This means, that numerical diffusion in the examined implicit method monotonically decreases with smaller time-step sizes.
Likewise, the dispersion relation for the explicit scheme can be obtained by dividing Equations (1.41) and (1.42). Based on this, Durran [2010] gives the phase-speed $v_{ps} = \frac{\omega_r}{k}$ in the limit of good resolution for the explicit scheme as

$$\frac{\omega_r}{k} \approx v_j \left(1 - \frac{(k\Delta x)^2}{6}(1 - c)(1 - 2c)\right). \tag{1.50}$$

Consequently, no change on v_j is imposed when $c = 1$ or $c = 1/2$ and increasing values when $c \to 0$. The dispersion relation for the implicit scheme can be directly read from the imaginary part in Equation (1.48) as

$$\omega_r = \frac{1}{\Delta t}\tan^{-1}\left[\frac{\sin(k\Delta x)}{1/c + 1 - \cos(k\Delta x)}\right] \tag{1.51}$$
$$\approx \frac{1}{\Delta t}\left[ck\Delta x - \frac{1}{6}(k\Delta x)^3 c(2c^2 + 3c + 1)\right], \quad \text{(small } \Delta x\text{)} \tag{1.52}$$
$$\Rightarrow \frac{\omega_r}{k} \approx v_j \left[1 - \frac{(k\Delta x)^2}{6}(1 + c)(1 + 2c)\right]. \tag{1.53}$$

As for diffusion, the analyzed implicit scheme shows monotonically decreasing dispersion with smaller Courant numbers for $c > 0$ but is never zero for finite Δx. The implicit method has a decelerating effect. This contrasts the explicit scheme, where diffusion decreases with smaller time steps but dispersion increases, and the optimal combination is at $c = 1$. In the analyzed first-order forward-/ backward-Euler, first-order donor-cell scheme, the implicit method is for every $0 < c < 1$ more diffusive and shows higher dispersion than its explicit counterpart as depicted in Figure (1.1).

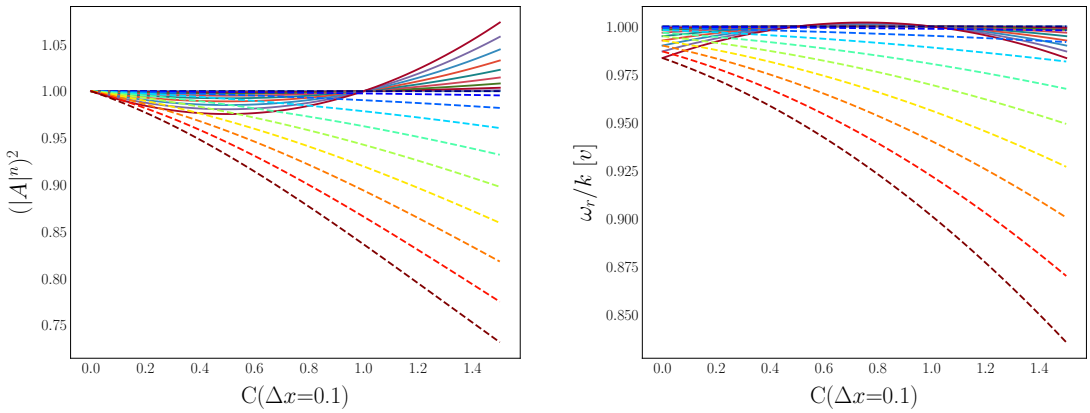


Figure 1.1: Diffusion (left) and dispersion (right) in comparison for explicit (solid line) and implicit (dashed line) forward-/ backward-Euler scheme with first order donor cell. Different colors correspond to different values of k from zero (light blue) to pi (dark red). x-axis is given by the Courant-number $C = v\frac{\Delta t}{\Delta x}$ for fixed $\Delta x = 0.1$. Solid-/ dashed lines in the same color are computed for equal parameters.

1.2.3 Applications of Implicit Methods

The simplistic analysis above illustrates the three main properties of implicit schemes a physicist who chooses a method might be interested in. The first is unconditional stability, which allows for arbitrarily large Courant numbers and time steps. The second is their monotonically decreasing dissipation towards lower time steps, which still is, in all cases, larger than that of the explicit scheme. The third is that a large system of equations needs to be inverted to find a solution. The last point oftentimes necessitates a large amount of computational resources and makes the implementation of implicit schemes difficult. One may, very rightfully, ask, under which conditions it makes sense to use an implicit scheme.
Indeed, the situation displayed in Figure 1.1 is kind of a worst-case scenario from the point of the implicit method since it describes the modeling of a transient flow in the absence of external forces. An obvious use case for implicit methods is to find stationary solutions to the NSE, also known as the modeling of steady flows. In such flows, the evolution of transient modes is not of interest and their diffusion, if considered at all, is desirable. Additionally, the flow usually has to be integrated over long time scales, making explicit methods with their time-step limitation oftentimes unfeasible.
A second case, where implicit methods are of interest, is when the Courant criteria for the modes of interest are distinct from the most restricting criteria in the system. As mentioned above, the Courant criterion can be motivated by the spectral norm of the matrix in Equation (1.37) being smaller unity. In analogy, this

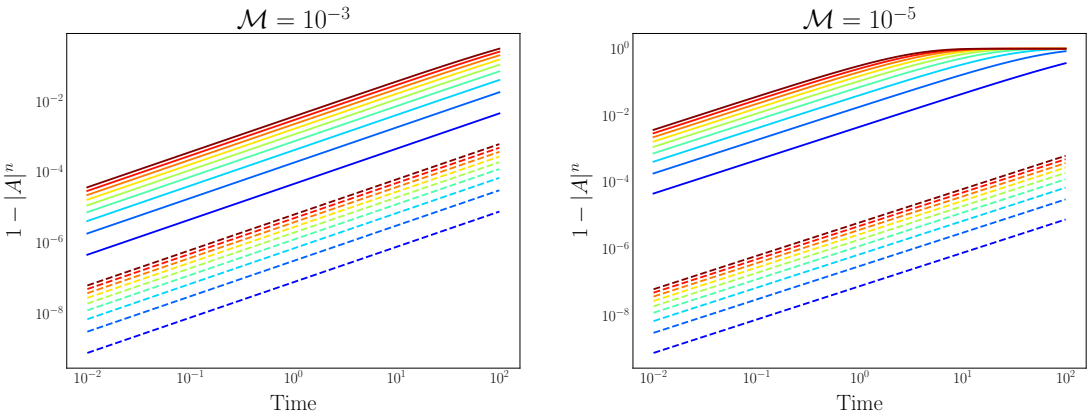


Figure 1.2: The ordinate shows the change to an initial dimensionless amplitude of one $(1 - |A|^n)$ at a time step n. Depicted is the comparison of time-integrated numerical diffusion at Mach numbers 10^{-3} (left) and 10^{-5} (right) between an explicit scheme (forward-Euler, solid lines) and implicit scheme (backward-Euler, dashed lines). Values are obtained using Equations (1.43) and (1.49), respectively. The Courant numbers for both schemes are chosen as $C = 0.25$ but for the explicit scheme with respect to the sound speed and for the implicit scheme with respect to the advection velocity. Calculation is done using $\Delta x = 10^{-3}$. For both Mach-numbers, the implicit scheme shows lower diffusion at integrated time. Colors represent the wave number.

can be utilized when considering elliptic extensions to the hyperbolic expression in Equation (1.30). Such an extension transfers the equation to a mixed hyperbolic-parabolic expression. Since the solution to a parabolic partial differential equation is not given by propagating waves, the time step restriction for the mixed system is not necessarily limited by the Courant condition[6]. As an example, the diffusion term $\nabla(D\nabla q)$ is examined here, which, in one-dimensional, Cartesian coordinates, and for a constant diffusion coefficient D can be written as $D\partial_x^2 q$, or discretized and including the time step Δt as

$$D\Delta t\frac{q_{j+1}-2q_j+q_{j-1}}{(\Delta x)^2} \rightarrow \frac{D\Delta t}{(\Delta x)^2}\begin{pmatrix} \ddots & & & & & \\ \cdots & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & 1 & -2 & \cdots & \\ & & & & \ddots & \end{pmatrix}\begin{pmatrix} \vdots \\ q_{j+1} \\ q_j \\ q_{j-1} \\ \vdots \end{pmatrix}. \quad (1.54)$$

The spectral norm of the matrix is $|\rho| = 2$, and the resulting time-step restriction $\Delta t < \frac{(\Delta x)^2}{2D}$ can be much stricter than the Courant condition[7]. A Comparison of the velocity-limited time step at Courant number C and the diffusion-limited time step gives

$$\frac{C\Delta x}{v} > \frac{(\Delta x)^2}{2D} \quad (1.55)$$

$$\Leftrightarrow v < \frac{2DC}{\Delta x}. \quad (1.56)$$

In this case, that is when the velocity of interest $v < \frac{2DC}{\Delta x}$, one might want to use an implicit scheme.
Another case where one might not want to resolve the largest eigenvalue of the characteristic matrix, also known as the largest characteristic, also known as the fastest supported velocity, is when pressure is taken into the equation. The - in the truest sense - characteristic velocity exerted by pressure changes is the speed of sound c_s. It can be derived in a similar way as above when the Euler-equations are considered, and the vectors $\vec{q}$ contain velocity, pressure, and density. This is done oftentimes in the literature, e.g., in Toro [2009], LeVeque [2002] in continuous space. Under conditions when the speed of sound is much faster than the velocities of interest, the Mach number is low, and implicit schemes become attractive in terms of computational time and numerical dissipation. A comparison of the numerical diffusion

[6]The Courant criterion, by its definition shown above, limits the time step to the largest characteristics of a hyperbolic system. This is the propagation speed of the fastest propagating wave within that system.

[7]When more than one-dimension is modeled, the condition becomes stricter with a factor related to the dimensionality $1/N_{dim}$.

obtained for the explicit- and implicit schemes, described above, at two different Mach-numbers, is shown in Figure 1.2. The numerical diffusion for the implicit scheme is much lower than for the explicit scheme when time steps are integrated and realistic Courant numbers are chosen. The implicit scheme outperforms the explicit one because much fewer time steps are needed to integrate into the depicted time. The comparison assumes that sound waves are not of interest since they are not resolved in the implicit method. Conditions, where $v \ll c_s$ arise when the fluid under consideration is weakly compressible and large changes in pressure cause small changes in density. In astrophysics, this is the case, e.g., in the interiors of stars, and because of this, implicit methods are more prominent in stellar astrophysics.
Often, explicit methods are used as the basis for a solver and combined with implicit methods for single terms, which would otherwise make the time-step size very small, e.g., diffusion terms like in Equation 1.54. Implicit methods are also more prominent, where the inversion of the defining matrix is no prohibitive hurdle. This is the case in one-dimensional problems, where all entries in the resulting (sparse) system matrix are very close to the main diagonal, and direct inversion procedures are fast. In this case, the additional computational effort of inverting a matrix for the implicit solution is only a factor a few larger than the explicit calculation. This is not the case in two-/ and three-dimensional problems, where the bandwidth of the system matrix is much larger and direct methods scale poorly. In astrophysics, globally implicit solvers modeling the full NSE in more than one dimension are rare, and there is no textbook solution for building *the best* solver. In fact, the performance of linear solvers is often highly problem-dependent. The same holds true for techniques to bring the NSE to a solvable (linear) form and for the discretization schemes applied.

1.3 Relevance to this Thesis

Section 1.1 of this chapter gives an introduction to the physical processes relevant in the context of this thesis, while Section 1.2 gives the motivation for the usage of an implicit method. The latter sets the stage for Chapter 2, where the numerical methods needed to develop an implicit code are presented. The developed code, MATRICS, is then used to model the GSF and COS, introduced in Section 1.1.4, in a spherical geometry largely supported by pressure. Kinematic and thermal diffusion are modeled, making the considerations in Section 1.2.3 very relevant to Chapter 3. Chapter 4 analyzes the outcome of pebble cloud collapse simulations in the shearing gravitational field of a solar mass star at distances that are equivalent to the asteroid belt (2.3 AU) and the Kuiper belt (25 AU). The broader context of these collapse simulations is introduced in Section 1.1.4 and narrowed down in Section 1.1.5.

2 The MATRICS Code

This chapter presents the novel, fully implicit, compressible hydrodynamic solver MATRICS. The development of MATRIC is the main project within this thesis. The focus is on the extensibility and flexibility of the developed code and the reduction of computational complexity. Both of these are achieved by a (semi) matrix-free solution approach as outlined in Section 2.3. With this method, it is easily possible to modify equations already implemented in the code or add new ones while simultaneously using the advantages of the method's implicitness.
This chapter's content is - with minor changes - taken from Meyer et al. [in production]. The first section, Section 2.1 summarizes the numerical methods and algorithms used in this chapter. It also puts this chapter into context relative to the actual code implementation. Section 2.2 outlines the equations solved in the code and describes the spatial- and temporal discretization of these equations. Also outlined is how the transition from the equations on a mesh to a system of equations is made. Section 2.3 describes the solution procedure of the code and gives a short overview of the integrated algorithms. The implementation of boundary conditions is treated in some detail. Last, Section 2.4 demonstrates the code's functionality and correctness by showing the solution to different test problems.

2.1 Overview

As mentioned, this chapter gives an introduction to the MATRICS code. It is by no means a Technical Design Report and does not outline the complete functionality of the solver. Instead, it constitutes a minimal working example of a combination of methods implemented in the code. This combination is not chosen randomly. It represents the empirically observed most stable combination of methods. An example is the fully simultaneous solution procedure for all equations summarized in a single system of equations as described in Section 2.2.5. The operator splitting approach is also implemented in the code but not presented below.
This not only keeps the chapter to a digestible length but also shows a ground truth for the development of an at least 2D- globally implicit solver. As a warning note, it shall be mentioned here that although implicit methods are formally unconditionally stable according to von Neumann's analysis, this is far from the truth in practice. The unpleasant truth is, that every implementation has to make trade-offs between the stability of the incorporated methods and the computational time needed for the solution procedure. To complicate matters further, this trade-off is generally problem-dependent. With this in mind, MATRICS is more of a test bench code than a rigidly defined set of methods. In this chapter, only the methods actively used in this thesis are explained and summarized below.

The outlining of implementation details is kept to a minimum - not to say it is ignored - within this chapter and the focus is on the description of the needed methods and approaches from a numerically interested physicist's point of view. Listings of the most important algorithms are given in the appendix and referred to in the text to avoid making the code appear too enigmatic. Some general facts about the implementation of MATRICS are summarized in the appendix Section A.1.

Summary of Methods

Because of the outstanding importance of (angular) momentum conservation in astrophysics, the conservative formulation of the NSE is used. Here, the momentum equations rely on momentum instead of velocity as a primitive variable, and the energy equation relies on internal energy instead of specific internal energy. The finite-volume method (FVM) is chosen because it provides flexibility to the implemented grid structures. The FVM, in turn, favors the use of a staggered grid, which includes the advantageous property of prohibiting momentum pressure odd-even decoupling Harlow and Welch [1965]. As the interpolation method to cell faces, the upwinding method is chosen. Cartesian, cylindrical, and spherical coordinate systems are implemented in 1D, 2D, and 3D-axisymmetry. Time integration is done with the first-order Backward Euler method (BDF-1) or the second-order 3-4-1 Backward Euler method (BDF-2). Section 2.2 gives details on these methods.
Instead of solving one system per equation, all equations are solved simultaneously in one large system. Because of the nonlinearity of the NSE and consequently also of the corresponding system of equations, Newton's method is implemented. This method requires multiple inversions of the Jacobian matrix, making the need for an efficient underlying linear solver even more imperative. Since the exact determination of all components making up the Jacobian Matrix is a highly nontrivial task, the right preconditioned restarted Generalized-Method of Residuals (GMRES) is implemented in a matrix-free way. In addition, the possibility of using the Incomplete Lower-Upper factorization Technique (ILUT) is provided. The implementation by Guennebaud et al. [2010] is integrated and the explicitly needed Jacobian matrix elements are calculated only approximately. In case ILUT is not used, no matrix elements are computed.

2.2 Discretization

This section provides a comprehensive overview of the fundamental components and numerical techniques employed in the MATRICS code. First, a detailed description of the NSE in their conservative formulation in a Cartesian, cylindrical, and spherical coordinate system is given. This is followed by details on the implementation of the finite volume method and the discretization on a staggered grid[1]. An overview of the implemented upwinding schemes, as well as time-stepping methods, is given. The section concludes by explaining the transition from the discretized equations to a linear algebraic system of equations required for the implicit solution of the NSE.

2.2.1 The Equations solved

The NSE in the conservative formulation and differential form are given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0, \tag{2.1}$$

$$\frac{\partial \vec{m}}{\partial t} + \nabla \cdot (\vec{m} \otimes \vec{v}) = -\nabla P + \nabla \cdot \hat{\vec{\tau}} - \rho \nabla \Phi_g, \tag{2.2}$$

$$\frac{\partial \mathcal{E}^{total}}{\partial t} + \nabla \cdot \left(\mathcal{E}^{total} \vec{v}\right) = -\nabla \cdot (P \vec{v}), \tag{2.3}$$

with density ρ, velocity vector $\vec{v}$, linear-momentum vector $\vec{m}$, pressure P, viscous stress tensor $\hat{\vec{\tau}}$, total internal energy density $\mathcal{E}^{total}$ and gravitational potential Φ_g. To close the system one has to relate momentum to velocity and pressure to the conserved variables either with an appropriate equation of state in the (weakly) compressible case or a pressure-Poisson equation in the incompressible case. The implemented default option is to use the caloric equation of state $P = (\gamma - 1)\mathcal{E}$ with $\mathcal{E} = \rho e$. This is a reasonable approximation for low Mach number flows. In this case, the energy Equation (2.3) simplifies to

$$\frac{\partial \mathcal{E}}{\partial t} + \nabla \cdot (\mathcal{E} \vec{v}) = -P \nabla \vec{v} \quad . \tag{2.4}$$

Section 2.4.2 shows a case where an equation of state including the kinetic term $P = \left(\mathcal{E}^{total} - \frac{1}{2}\rho \vec{v} \cdot \vec{v}\right)(\gamma - 1)$, together with Equation 2.3 has to be used.
Since the system is solved in either 1D or 2D for Cartesian,- cylindrical- or spherical coordinates or 3D axisymmetry for cylindrical- or spherical- coordinates only, derivatives in φ-direction arising in the divergence, gradient, and the tensor-divergence vanish. Further, the transformation $\theta = \theta' - \frac{\pi}{2}$ is imposed in the spherical case. This allows for an intuitive formulation of the discretized equations with $\theta = 0$ at

[1] At this point, I would like to thank Dr. A. A. Hujeirat for his suggestions to use the conservative formulation of the NSE with the finite volume method in conjunction with a staggered grid.

the equatorial plane. The resulting equations describing the gradient, divergence, and tensor divergence operators can be found in the appendix to this chapter (see also, e.g., Bird et al. [2006] or Kuo and Acharya [2012]).
The outer product $(m \otimes v) = (v \otimes m) = \rho(v \otimes v)$ is independent of the used coordinate system, but its divergence is not. Kley [1998] has shown that for the conservation of angular momentum, it makes sense to exchange the equation for linear momentum in φ-direction with the expression for angular momentum since it has the advantage of vanishing source terms in the φ-component. In the following, the linear momentum $m_i = \rho \cdot v_i$ is distinguished from the angular-momentum

$$\begin{aligned} m_\varphi = \rho v_\varphi \rightarrow l = \rho \underbrace{v_\varphi R}_{\hat{v}_\varphi} \quad &\text{(Cylindrical)}, \\ m_\varphi = \rho v_\varphi \rightarrow l = \rho \underbrace{v_\varphi r \cos\theta}_{\hat{v}_\varphi} \quad &\text{(Spherical)} \quad . \end{aligned} \tag{2.5}$$

The viscous stress tensor appearing in Equation (2.2) is defined as

$$\hat{\vec{\tau}} = \underbrace{\mu \left(\vec{\nabla}\vec{v} + (\vec{\nabla}\vec{v})^T \right)}_{\tau^{(1)}} - \underbrace{\left(\frac{3}{2}\mu - \kappa \right) (\vec{\nabla} \cdot \vec{v})\delta}_{\tau^{(2)}}, \tag{2.6}$$

with shear viscosity μ, diagonal unit tensor δ, and bulk viscosity κ, which are assumed zero in the following (Newtonian fluid). The second term on the right-hand side is determined by the velocity divergence, which also vanishes in the incompressible case. In components, the first term on the right-hand side is symmetric and given in the appendix, where the derivatives in z, θ, and φ-direction are set to zero.

2.2.2 Finite Volume Method

With the Finite Volume Method, the spatial domain on which the NSE are solved is expressed as a finite number of sub-cells termed control volumes (CV) (Ferziger and Perić [2002]). The primitive variables are defined as averages on the cell's volume, and Gauss' theorem is used to express the arising volume integrals for the advection term in the form of fluxes through each CV's surface. The fluxes are then summed to obtain the total flux of every quantity in or out of each CV. By definition, the flux of a quantity through an interface is this quantity at the interface times the velocity orthogonal to the interface. Because of this, it makes sense to calculate and store the velocity components directly at the interface while scalar quantities are stored at cell centers. A grid expressing this approach is called a staggered grid.
The staggered grid has the advantages of prohibiting the decoupling of pressure gradients and velocities as well as prohibiting the need for velocity interpolations to cell faces for flux calculations. In the 3D-axisymmetric case, where also v_φ is

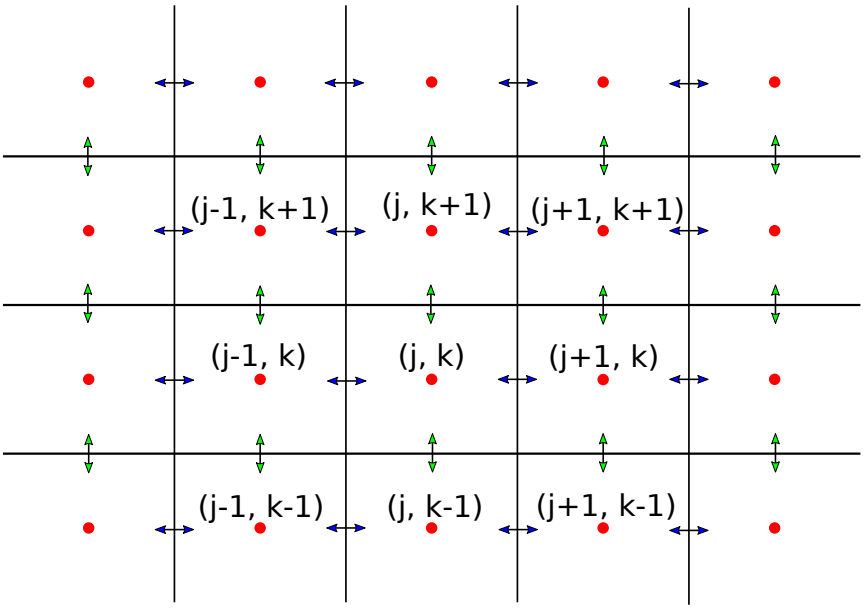


Figure 2.1: Illustration of staggered grid. Scalar properties are defined at the locations of the red dots in the center of each cell. The velocity's x, R, or r-component is located at the interfaces marked with blue arrows, and the y, z, or θ-component of the velocity is calculated at the interfaces marked by green arrows. Counting is indicated by the index k in vertical or polar direction and j in horizontal or radial direction. The location of variables at cell centers is expressed with integer indices and at cell faces with the integer $\pm\frac{1}{2}$ notation.

calculated- but no according derivatives are needed, v_φ and l are treated as scalar quantities and are located at cell centers, too. The schematic setup of the staggered grid is shown in Figure 2.1 where the positions of cell averages of the scalar quantities are indicated by red dots and those of vectorial components by arrows.

2.2.3 Discretization and Interpolation

For spatial discretization, variables defined at cell centers and cell faces are distinguished. In the following, the generic variable $q \in [\rho, \mathcal{E}, m, n, l]$ is used as placeholder, where m is the linear-momentum in x, R and r-direction (j-direction) and n is the linear-momentum in y, z or θ-direction (k-direction). l is the angular momentum, as defined in the previous section. The advective term is

$$\nabla \cdot (q\vec{v}) \,|_{j,k} = \frac{u_{j+\frac{1}{2},k} \,[q]_{j+\frac{1}{2},k} - u_{j-\frac{1}{2},k} \,[q]_{j-\frac{1}{2},k}}{x_{j+\frac{1}{2},k} - x_{j-\frac{1}{2},k}} + \frac{v_{j,k+\frac{1}{2}} \,[q]_{j,k+\frac{1}{2}} - v_{j,k-\frac{1}{2}} \,[q]_{j,k-\frac{1}{2}}}{y_{j,k+\frac{1}{2}} - y_{j,k-\frac{1}{2}}}, \tag{2.7}$$

where $[q]_{j\pm\frac{1}{2},k}$ is the interpolated (upwinded) value of q to the respective interface between cells j and $j \pm 1$. Equally, $[q]_{j,k\pm\frac{1}{2}}$ is the upwinded value of q to the interface between k and $k+1$. The velocities u and v are computed from their linear-momentum counterparts, obtained from the momentum equations as

$$u_{j\pm\frac{1}{2},k} = \frac{m_{j\pm\frac{1}{2},k}}{\overline{\rho}_{j\pm\frac{1}{2},k}}, \tag{2.8}$$

$$v_{j,k\pm\frac{1}{2}} = \frac{n_{j,k\pm\frac{1}{2}}}{\overline{\rho}_{j,k\pm\frac{1}{2}}}. \tag{2.9}$$

Here, the directly known values of $m_{j\pm\frac{1}{2},k}$ and $n_{j,k\pm\frac{1}{2}}$ are used together with the density values, which are linearly interpolated as

$$\overline{\rho}_{j\pm\frac{1}{2},k} = \frac{\rho_{j,k} + \rho_{j\pm1,k}}{2}, \tag{2.10}$$

$$\overline{\rho}_{j,k\pm\frac{1}{2}} = \frac{\rho_{j,k} + \rho_{j,k\pm1}}{2}. \tag{2.11}$$

In Cylindrical coordinates, the advection term is

$$\nabla \cdot (q\vec{v}) \,|_{j,k} = \frac{R_{j+\frac{1}{2},k} u_{j+\frac{1}{2},k} \,[q]_{j+\frac{1}{2},k} - R_{j-\frac{1}{2},k} u_{j-\frac{1}{2},k} \,[q]_{j-\frac{1}{2},k}}{\frac{1}{2}\left(R^2_{j+\frac{1}{2},k} - R^2_{j-\frac{1}{2},k}\right)} + \frac{v_{j,k+\frac{1}{2}} \,[q]_{j,k+\frac{1}{2}} - v_{j,k-\frac{1}{2}} \,[q]_{j,k-\frac{1}{2}}}{z_{j,k+\frac{1}{2}} - z_{j,k-\frac{1}{2}}}, \tag{2.12}$$

and in spherical coordinates (see e.g. Shadab et al. [2019], Wang and Johnsen [2017])

$$\nabla \cdot (q\vec{v})\,|_{j,k} = \frac{r^2_{j+\frac{1}{2},k} u_{j+\frac{1}{2},k} \left[q\right]_{j+\frac{1}{2},k} - r^2_{j-\frac{1}{2},k} u_{j-\frac{1}{2},k} \left[q\right]_{j-\frac{1}{2},k}}{\frac{1}{3}\left(r^3_{j+\frac{1}{2},k} - r^3_{j-\frac{1}{2},k}\right)} + \frac{1}{r_{j,k}} \frac{r_{j,k+\frac{1}{2}} \cos\theta_{j,k+\frac{1}{2}} v_{j,k+\frac{1}{2}} \left[q\right]_{j,k+\frac{1}{2}} - r_{j,k-\frac{1}{2}} \cos\theta_{j,k-\frac{1}{2}} v_{j,k-\frac{1}{2}} \left[q\right]_{j,k-\frac{1}{2}}}{\sin\theta_{j,k+\frac{1}{2}} - \sin\theta_{j,k-\frac{1}{2}}} \quad . \tag{2.13}$$

In the staggered grid approach, the cells for the momenta in j- and k- direction are shifted half a cell in the respective direction and the advection equations take a slightly different form. For the momentum m, in j direction, the expression in Cartesian coordinates is

$$\nabla \cdot (m\vec{v})\,|_{j-\frac{1}{2},k} = \frac{\overline{u}_{j,k} \left[m\right]_{j,k} - \overline{u}_{j-1,k} \left[m\right]_{j-1,k}}{x_{j,k} - x_{j-1,k}} + \frac{\overline{v}_{j-\frac{1}{2},k+\frac{1}{2}} \left[m\right]_{j-\frac{1}{2},k+\frac{1}{2}} - \overline{v}_{j-\frac{1}{2},k-\frac{1}{2}} \left[m\right]_{j-\frac{1}{2},k-\frac{1}{2}}}{y_{j-\frac{1}{2},k+\frac{1}{2}} - y_{j-\frac{1}{2},k-\frac{1}{2}}}, \tag{2.14}$$

where the values of $\overline{u}$ and $\overline{v}$ are defined as

$$\overline{u}_{j,k} = \frac{1}{2} \frac{m_{j-\frac{1}{2},k} + m_{j+\frac{1}{2},k}}{\rho_{j,k}}, \quad \text{and} \tag{2.15}$$

$$\overline{v}_{j-\frac{1}{2},k+\frac{1}{2}} = \frac{1}{2}\left[\frac{n_{j-1,k+\frac{1}{2}}}{\frac{1}{2}\left(\rho_{j-1,k} + \rho_{j-1,k+1}\right)} + \frac{n_{j,k+\frac{1}{2}}}{\frac{1}{2}\left(\rho_{j,k} + \rho_{j,k+1}\right)}\right] \quad . \tag{2.16}$$

The procedure for the momentum in the k direction is equivalent, as are the expressions in cylindrical- and spherical- Coordinates.
The upwinding method is equivalent for scalar variables and momentum components. Implemented in the code are the (constant interpolation-) first-order accurate donor cell (DC) scheme

$$q_{j-\frac{1}{2}} = \begin{cases} q_{j-1} & , v_{j-\frac{1}{2}} > 0 \\ q_j & , v_{j-\frac{1}{2}} < 0 \end{cases}, \tag{2.17}$$

as well as the generalized upstream biased κ-scheme

$$
\begin{aligned}
q_{j-\frac{1}{2}} &= q_c + \underbrace{\frac{1}{4}\left[(1-\kappa)\left(q_c - q_u\right) + (1+\kappa)\left(q_d - q_c\right)\right]}_{\sigma^{\kappa}_{j-\frac{1}{2}}}, \\
(q_d, q_c, q_u) &= \begin{cases} (q_j, q_{j-1}, q_{j-2}), & v_{j-\frac{1}{2}} > 0 \\ (q_{j-1}, q_j, q_{j+1}), & v_{j-\frac{1}{2}} < 0, \end{cases}
\end{aligned}
\tag{2.18}
$$

where, e.g., for $\kappa = -1$ the second order Linear Upwind Scheme (LUS) (Price et al. [1966]) and for $\kappa = \frac{1}{2}$ the third order accurate Quadratic Upwind Interpolation scheme (QUICK) (Leonard [1979]) is recovered. $\sigma^{\kappa}_{j-\frac{1}{2}}$ is the inferred slope at the $j - \frac{1}{2}$ interface.
The velocities for upwinding (v or u) are taken at the respective interfaces of the variable cells. It becomes obvious that one has to take care of the non-linearities present, especially in the equations of momenta since the condition for the direction in which upwinding is conducted depends on the result of the upwinding procedure itself. This issue, as well as the general implicit solution procedure, are addressed in the next Section. From the interpolation methods in Equations (2.17) and (2.18), only the DC scheme is total variation diminishing (TVD). To prevent the occurrence of spurious oscillations in the vicinity of discontinuities (see Section 2.4.2), various slope-limiters (see, e.g., Zhang et al. [2015] and references therein) are implemented. The final interpolation at the left interface of a cell j is then given as

$$
\begin{aligned}
[q]_{j-\frac{1}{2}} &= q^{\mathbf{DC}}_{j-\frac{1}{2}} - \frac{1}{2}\Phi(r_{j-\frac{1}{2}})\left(q^{\mathbf{DC}}_{j-\frac{1}{2}} - q^{\kappa\textbf{-scheme}}_{j-\frac{1}{2}}\right) && (2.19) \\
&= q^{\mathbf{DC}}_{j-\frac{1}{2}} + \Phi(r_{j-\frac{1}{2}})\sigma^{\kappa}_{j-\frac{1}{2}}. && (2.20)
\end{aligned}
$$

The resulting flux $f_{j-\frac{1}{2}} = [q]_{j-\frac{1}{2}} v_{j-\frac{1}{2}}$ is given by this interpolation, multiplied with the velocity at that position. The superbee limiter reads

$$
\Phi(r_{j-\frac{1}{2}}) = \max\left(0, \min(1, 2r_{j-\frac{1}{2}}), \min(2, r_{j-\frac{1}{2}})\right), \tag{2.21}
$$

with the ratio of successive gradients

$$
r_{j-\frac{1}{2}} = \frac{\sigma^{\kappa}_{j-\frac{3}{2}}}{\sigma^{\kappa}_{j-\frac{1}{2}}} \quad . \tag{2.22}
$$

Usually one would express Equation (2.20) as function of the left state at an interface $q^{L}_{j-\frac{1}{2}}$ and the corresponding right state at the same interface $q^{R}_{j-\frac{1}{2}}$ as $[q]_{j-\frac{1}{2}} = f(q^{L}_{j-\frac{1}{2}}, q^{R}_{j-\frac{1}{2}})$. In this notation, the left and right states correspond to the upwind

and downwind states at the interface. The superscripts are omitted in Equation (2.20) since either $q^L_{j-\frac{1}{2}}$ or $q^R_{j-\frac{1}{2}}$ is used. This depends on the direction of the flow and never both at the same time are used, such that $f(\cdot,\cdot)$ itself is an upwinding operator. The obtained scheme is similar to the MUSCL scheme (van Leer [1979]) and equivalent for $\kappa = \frac{1}{2}$.

2.2.4 Time-Stepping

Temporal discretization is much simpler compared to spatial discretization, yet its implications are much more complicated. Implement in MATRICS is the first-order accurate Euler scheme (BDF-1)

$$\frac{\partial q}{\partial t} \Rightarrow \frac{q^{n+1} - q^n}{\Delta t}, \tag{2.23}$$

and the second order accurate 3-4-1 Euler scheme (BDF-2)

$$\frac{\partial q}{\partial t} \Rightarrow \frac{3q^{n+1} - 4q^n + q^{n-1}}{2\Delta t}, \tag{2.24}$$

(e.g Hairer et al. [1993]) where the first time step of every simulation run is always carried out using the BDF-1 scheme. This workaround can be overcome by implementing the trapezoidal version of the BDF-2 scheme (TR-BDF-2), a one-step process. The implementation of TR-BDF-2 is left for future work. Different modes for the time-step size calculation are implemented. The first is an a priori configured constant time step and the second is an exponentially growing time step

$$\Delta t^{n+1} = \begin{cases} \alpha \cdot \Delta t^n & , \Delta t^n \leq \Delta t_{max} \\ \Delta t^n & , \text{else} \end{cases} \tag{2.25}$$

with α being the growth factor, typically is chosen as $1 < \alpha < 1.1$. Δt_{max} is a pre-defined upper limit for the time step. The third option is a dynamic time-step proposed by Dorfi [1997], where

$$\Delta t^{n+1} = \begin{cases} \frac{1}{2}\Delta t^n & , s > s_{max} \\ \Delta t^n & , \frac{1}{2}s_{max} \leq s \leq s_{max} \\ \frac{3}{2}\Delta t^n & , s < \frac{1}{2}s_{max} \end{cases} \tag{2.26}$$

with

$$s = \frac{|\vec{x}^{n+1} - \vec{x}^{n}|}{|\vec{x}^{n}|}. \tag{2.27}$$

$\vec{x}^n$ is the vector containing all variables at time-step n and $s_{max} \approx 0.1$ typically. As Δt^0, typically a time-step that is one order of magnitude lower than the expected advection velocity is selected. Alternatively, for equilibrium initial conditions the time-step is chosen $\Delta t \sim 10^2 \Delta t_{\mathbf{explicit}}$ of the explicit time-step, defined via the speed of sound. Sometimes it is best to find a time step by trial and error. In addition, the possibility to limit all time-step sizes to a Courant number of one, given by the advection velocity ($\mathrm{CFL_{adv}}$) is provided. Depending on the size of the initial time-step, this ignores the condition imposed by the speed of sound (CFL_{cs}).
One has to note here, that both Courant numbers are calculated by the code as CFL= $\frac{\max(u_{\max}, v_{\max})\Delta t}{\Delta x}$. The only possibility to distinguish between CFL_{cs} and $\mathrm{CFL_{adv}}$ is the size of Δt itself. When Δt is such that $\frac{c_s \Delta t}{\Delta x} > 1$, sound-waves are filtered (Section 2.4.1) and the sound-speed is not represented in $\max(u_{\max}, v_{\max})$. The code internal CFL number is thus

$$\mathrm{CFL} = \begin{cases} \mathrm{CFL_{adv}} & \text{if } \Delta t > \Delta t_{\mathrm{explicit}} \\ \mathrm{CFL_{cs}} & \text{else} \end{cases} . \tag{2.28}$$

2.2.5 System Construction

In the implicit method, the fluxes and source terms are evaluated depending on the yet unknown properties at the next time level $n+1$. For a single linear equation in 1D, this is straightforward, e.g., the continuity equation in semi-discrete form for a time-independent velocity vector becomes

$$\rho^{n+1} + \nabla \cdot \left(\rho^{n+1}\vec{v}\right) \Delta t = \rho^{n} \quad \text{(BDF-1)}, \tag{2.29}$$

$$\frac{3}{2}\rho^{n+1} + \nabla \cdot \left(\rho^{n+1}\vec{v}\right) \Delta t = \underbrace{\frac{4}{2}\rho^{n} - \frac{1}{2}\rho^{n-1}}_{\text{known}} \quad \text{(BDF-2)} \quad . \tag{2.30}$$

When spatially discretized with the methods outlined in the previous section, one can utilize two different formulations for the fully discretized NSE. The operator form for every cell, denoted as

$$\mathcal{L}_{\rho}(\rho^{n+1}) = rhs, \tag{2.31}$$

or the form of a system with the systems matrix A_ρ

$$A_\rho \rho^{n+1} = rhs, \tag{2.32}$$
$$\Rightarrow A_\rho = \alpha I + \Delta t B_\rho \quad . \tag{2.33}$$

Here, I is the identity matrix, and B_ρ is the matrix defined by the advection term. B_ρ depends on the utilized upwinding scheme. For the BDF-1 scheme, $\alpha = 1$, and for BDF-2 $\alpha = \frac{3}{2}$. The subscript ρ indicates that the system represents the continuity equation. Although only the continuity equation is used as an example, equivalent formulations can be found for the other equations in operator-form $\mathcal{L}_\mathcal{E}$, $\mathcal{L}_m$, $\mathcal{L}_n$, $\mathcal{L}_l$ as well as in matrix-form $A_\mathcal{E}$, $A_m(m)$, $A_n(n)$ and A_l. It should be noted that the matrices A_q and B_q are generally nonlinear.
In the 1D case, the matrices A_q are tridiagonal for DC and penta-diagonal for LUD and QUICK, as one can easily verify by examining the stencil of the respective schemes. The two-dimensional and 3D-axisymmetric cases are more complicated since one needs to decide on the ordering of j and k when constructing the system matrix as well as the systems vector $\mathcal{L}_q$. It makes sense to order all j for every k or all k for every j. In both cases, far-off-diagonal elements are introduced into the system matrices. Hujeirat and Rannacher [1998] use the first approach because of the radial dominance of the considered flows and the resulting advantage of keeping the radial coupling terms close to the main diagonal in their defect-correction solution approach. Although this is of limited importance in the matrix-free approach, the same ordering is chosen since it is advantageous to have larger coupling terms close to the main diagonal for some of the preconditioners implemented (see Section 2.3.3).
Further complexity arises from the need to solve multiple interdependent equations concurrently. Different strategies have been explored to tackle this challenge, including sequential approaches, where each equation is solved one after the other, and simultaneous approaches, where all equations are collectively formulated into a single system. Various combinations of these approaches have been investigated in prior work (Hujeirat [2005], Hujeirat and Rannacher [1998]). Although the implementation in MATRICS allows for flexibility in incorporating different combinations of sequential and simultaneous parts, the focus in this chapter is exclusively on the fully simultaneous case where for each cell, represented by indices j and k, a vector of length N (number of equations solved) is utilized to store the solved variables. This is the method that is numerically the most stable. For the NSE the vector of a single cell is then

$$\vec{cell}_{j,k} = \begin{pmatrix} \rho_{j,k} \\ \mathcal{E}_{j,k} \\ m_{j,k} \\ n_{j,k} \\ l_{j,k} \end{pmatrix} \in \mathbb{R}^N, \quad \mathcal{L}_{j,k} = \begin{pmatrix} \mathcal{L}_\rho(\rho,\mathcal{E},m,n,l)_{j,k} \\ \mathcal{L}_\mathcal{E}(\rho,\mathcal{E},m,n,l)_{j,k} \\ \mathcal{L}_m(\rho,\mathcal{E},m,n,l)_{j,k} \\ \mathcal{L}_n(\rho,\mathcal{E},m,n,l)_{j,k} \\ \mathcal{L}_l(\rho,\mathcal{E},m,n,l)_{j,k} \end{pmatrix} \quad . \tag{2.34}$$

This fills K lines, each of length J as

$$\vec{line}_k = \begin{pmatrix} cell_{0,k} \\ cell_{1,k} \\ ... \\ cell_{J-2,k} \\ cell_{J-1,k} \end{pmatrix} \in \mathbb{R}^{N \times J}, \quad \vec{\mathcal{L}}_k = \begin{pmatrix} \mathcal{L}_{0,k} \\ \mathcal{L}_{1,k} \\ ... \\ \mathcal{L}_{J-2,k} \\ \mathcal{L}_{J-1,k} \end{pmatrix}, \tag{2.35}$$

which in turn make up the final vector

$$\vec{x} = \begin{pmatrix} line_0 \\ line_1 \\ ... \\ line_{K-2} \\ line_{K-1} \end{pmatrix} \in \mathbb{R}^{N \times J \times K}, \quad \mathcal{L} = \begin{pmatrix} \mathcal{L}_0 \\ \mathcal{L}_1 \\ ... \\ \mathcal{L}_{K-2} \\ \mathcal{L}_{K-1} \end{pmatrix} \quad . \tag{2.36}$$

The consequence of such a multivariate systems vector with coupled variables is that the structure of the system matrix changes from a line-diagonal matrix to a block-diagonal matrix since for every cell (j, k), there exists now an $N \times N$ block instead of a single entry. The system to solve for $\vec{x}^{n+1}$ has the form

$$A(\vec{x}^{n+1})\vec{x}^{n+1} = \vec{x}^{n}. \tag{2.37}$$

It is noteworthy at this point that the arising system is very sparse, contains far off-diagonal elements, is non-symmetric, non-linear, and not necessarily positive definite or diagonally dominant but also non-singular and diagonalizable in real space. These properties make it a) possible to solve the system and b) particularly difficult to do so in an efficient way.
One may notice that the construction for the system matrix is used in the example above instead of for the Jacobian matrix of the system. This is done because the definition of the system matrix is more intuitive and with the definition $A = \frac{\partial \mathcal{L}_q}{\partial x}$ and $J = \frac{\partial (\mathcal{L}_q - rhs)}{\partial x}$ the matrices are equivalent for the continuity-, energy- and angular momentum equations.

2.3 Solution Procedure

To solve the non-linear, large, and sparse system mentioned above, one needs to bring the system to a linear form first (Equation (2.38)). This can be done by a kind of Newton's method or via a defect correction approach (Auzinger [1987]). Alternatively, the non-linear system can be solved with a dedicated method for non-linear systems like, e.g., the Full Approximation Scheme (FAS) based on the multigrid method (van Henson [2003]). In MATRICS, the subjectively more intuitive approach of applying Newton's method first and then applying a linear solver iteratively to obtain a solution is chosen. In the following section, the overall solution method is briefly outlined. Afterward, details on the different modules that are incorporated are given.

2.3.1 General Approach

First, the system summarizing all NSE on the complete solution domain in Equation (2.37) is linearized with Newton's method. In this approach, the Jacobian matrix of the residual function for the system is utilized to calculate approximations of the system's solution vector. Except for the non-linearity, the system matrix's other properties - especially the sparsity and invertibility - are inherited to the Jacobian matrix. To ensure distinguishability, the non-linear system is referred to in operator form as $\vec{\mathcal{L}}$ and the linearized version as $\vec{L}$. The expression "linearized" refers to the fact that in the advection term of the momentum-equations, the transition (in semi-discrete form)

$$\left(\vec{m}^{n+1} \otimes \vec{v}^{n+1}\right) \Rightarrow \left(\vec{m}^{i} \otimes \vec{v}^{i-1}\right), \tag{2.38}$$

is made, using i as the Newton iteration step index (see Section 2.3.2). This holds equivalently for other terms implemented, e.g., the centrifugal force. Further, the non-linear residual function $\vec{\mathcal{R}} = \vec{\mathcal{L}} - \vec{b}$ and the linearized residual function $\vec{R} = \vec{L} - \vec{b}$ where $\vec{b}$ is the right hand side vector (see Equation (2.31)) are distinguished. The Jacobian J_R is always calculated from $\vec{R}$.
Because of the large size and the high degree of sparsity of J_R, direct dense solution methods such as Gauss elimination are not feasible. Luckily, dedicated solution methods for sparse linear systems exist as direct-sparse methods (LU, QR, Cholesky, etc.), iterative methods (Jacobi, Gauss-Seidel, SOR, etc., and Krylov methods), and multigrid methods (Barrett et al. [1994]). The first class of direct methods is only attractive for the one-dimensional case where J_R is (block)-diagonal and only matrix elements close to the main diagonal exist. Since in MATRICS, the more general 2D case where also far-off diagonal elements are present is treated, the focus is on iterative methods, especially Krylov subspace methods (Sections 2.3.3 and 2.3.3).
These methods find a solution iteratively by spanning for the residual function $r_0 = rhs - Ax_0$ the Krylov-subspace $\mathcal{K}_m = \{Ar_0, A^2r_0, ..., A^{m-1}r_0\}$, and hence only

rely on matrix-vector products to find the solution. This is particularly interesting because the matrix-vector product can be expressed in operator form as described in Section 2.2.5. This allows for the solution of the linear system in a matrix-free way, which is not possible when incorporating the Jacobi or Gauss-Seidel methods - or any derivatives of them - since they explicitly rely on individual matrix components. At this point, it is noteworthy that using a matrix-free approach is useful for computing the matrix-vector product on the fly and offers the possibility of saving storage. An even more important aspect is, that for Newton's method, it is advantageous to have the Jacobian matrix-vector product as exact as possible to guarantee good convergence behavior[2] and, hence, have the Jacobian matrix calculated exactly. This task is difficult to achieve when the matrix is expressed by components and very easy to achieve when the matrix-vector product is approximated by a finite difference approach.

2.3.2 Newton's Method

Methods building on the Newton method for linearization and using a Krylov-subspace method for the solution of the consecutive equation are called Newton-Krylov methods. The multivariate Newton method can be expressed as

$$\begin{aligned} \vec{x}^{i+1} &= \vec{x}^{i} + \left[\frac{\partial \vec{R}^{i}}{\partial \vec{x}^{i}}\right]^{-1} \cdot \vec{\mathcal{R}}^{i} \\ &= \vec{x}^{i} + \underbrace{\left[J_R(\vec{x}^{i})\right]^{-1} \cdot \vec{\mathcal{R}}^{i}}_{\vec{\mu}} . \end{aligned} \tag{2.39}$$

Here, μ is the correction per iteration step and i is the iteration step. Dropping the superscript i, the linear system to be solved once per every Newton-iteration step is then

$$J_{R^i}(\vec{x})\mu = \mathcal{R} \equiv \mathcal{L}(\vec{x}) - \vec{b} \quad . \tag{2.40}$$

The need for the calculation of the Jacobi-matrix here is quite obvious. The calculation can be done in three different ways, the first being manually calculating the coefficients and writing them in a sparse matrix. This can be computationally extremely fast but comes at the cost of the manual determination of each non-zero matrix element making the development difficult and the approach much less flexible. The second approach is automatic differentiation, which is relatively fast but

[2]The computation with an inexact Jacobian matrix is also possible. The method is then named Defect-correction (DC). When the DC approach is chosen, one would require an additional layer of iteration to converge to Newton's method. When this additional layer of iteration is not implemented in matrix-based approaches, one risks the solver's stability.

requires intimate knowledge about the sparsity pattern of the resulting matrix and imposes some implementation constraints. Both approaches have the disadvantage of having to store the complete and exact Jacobian matrix with all its elements. The prohibitive point, though, is not the storage of the Jacobian matrix but the computation of its elements. The exact computation of each and every matrix element can arguably be regarded as impossible in practice. At the same time, matrix-based methods have the advantage of being able to make use of efficient point methods.
The third method is the finite difference method, which does not require storing - or computing - individual matrix elements but only a single vector. In this method, the Jacobian can not be calculated stand-alone but only as matrix-vector product $J_R\vec{x}$. The biggest advantage of the finite difference method is that the Jacobian matrix-vector product can be approximated to good accuracy without much effort. The finite-difference approach is chosen in MATRICS. The first-order matrix-free vector product in dependence on the linearized operator L can be written in this approach as

$$J_{R^i}(\vec{x^i})\vec{\mu} = \frac{L(\vec{x^i} + \epsilon\vec{\mu}, \vec{x^{i-1}}) - L(\vec{x^i}, \vec{x^{i-1}})}{\epsilon}, \tag{2.41}$$

using the machine-precision epsilon ϵ_m via $\epsilon = \epsilon_m/||\vec{\mu}||$ as well as the Euclidean norm $||\cdot||$. Strictly speaking, this formulation yields Broyden's method (the secant-method) for finite ϵ, but both methods are equivalent in practice when solved to machine precision. The difficulty is to find the correction μ by iteratively finding a solution to the inversion problem

$$\frac{L(\vec{x^i} + \epsilon\vec{\mu}) - L(\vec{x^i})}{\epsilon} = \mathcal{L}(\vec{x^i}) - \vec{b}. \tag{2.42}$$

Section 2.3.3 describes methods for finding the solution for μ. The term "matrix-free" in the context of this work, does not merely refer to an implementation detail of the method in terms of programming. In applied mathematics, this term refers to the fact the desired matrix-vector product is evaluated through a vector-valued function rather than a matrix-vector multiplication and thus describes the method itself. In this case, this function is given by the right-hand side in Equation (2.41), while the left-hand side in that equation describes the matrix-based method.
In MATRICS, $\vec{x^{i-1}}$ and $\vec{x^i}$, and consequently the result of $L(\vec{x^i}, \vec{x^{i-1}})$, are support points, calculated once per Newton iteration step. $L(\vec{x^i} + \epsilon_m\vec{\mu}, \vec{x^{i-1}})$ is evaluated multiple times in every Newton step to find $\vec{\mu}$ such that Equation (2.42) is satisfied (see Section 2.3.3). In a matrix-based method, the equivalent to this form of updating would be to recalculate the Jacobian matrix after every linear solver iteration step, based on the value of the last iterate. In matrix-free methods, this is included by construction, and all variables and interpolations described in Section 2.2.3 are evaluated at the most recent linear-solver step of the most recent Newton-iteration

step. In this respect, matrix-based methods are more flexible since they allow the Jacobian matrix to be evaluated once per Newton step (Newton-Raphson method), or only once per time step (yielding a DC approach no matter how exactly the Jacobian matrix is calculated). This flexibility is sacrificed in MATRICS so that it does not become dependent on the difficult construction of the Jacobian matrix. The listings in the appendix Section A.4.3 show the necessary steps to construct the Jacobian matrix explicitly, as implemented in MATRICS for the purpose of preconditioning. The corresponding matrix-free functions are listed in the appendix Section A.4.4.
Independent of its matrix-free or matrix-based formulation, Newton's method is based on an updating step as given by Equation (2.39) and may not converge for non-linear problems using large advective Courant-numbers (e.g. $CFL_{adv} \gtrsim 2$ for Sods shock tube in Section 2.4.2). To counter this, the possibility of using damping (e.g., Nowak and Weimann [1992]) is provided. Here, instead of Equation (2.39), the update is performed as

$$\vec{x}^{i+1} = \vec{x}^{i} + \lambda_{\text{damp}}\vec{\mu}, \tag{2.43}$$

where $0 < \lambda_{\text{damp}} \leq 1$ is the damping factor. $\vec{\mu}$ is calculated without change to the un-damped case using Equation (2.42). The implementation of Newton's method in MATRICS is listed in the appendix Section A.4.1.

2.3.3 Linear Solvers

Since Equation (2.41) is used in a matrix-free way, a matrix-free solver is needed to solve for μ. The only possibilities are either to use an appropriate Krylov-subspace method directly or a geometric method (Schwarz type- or geometric multigrid methods) built around one. In the following, a brief and qualitative overview of the solvers implemented is given, with the goal of presenting the possible options for a solution pathway instead of giving a detailed description of the individual solvers. For this, the reader is referred to Saad [2003] and references therein as well as to the individual references within this section.

Krylov Methods

Krylov subspace methods obtain a solution by iteratively evaluating matrix-vector products and consequently spanning a Krylov subspace. A popular and very efficient choice for elliptical problems is the Conjugate Gradient (CG) method, which unfortunately can not be used since it is applicable only for symmetric and positive definite systems, which, for the hyperbolic system here (the discretized NSE), is not the case. This is intuitive since the upwind procedure, outlined in Section 2.2.3, favors one direction. This favoring of a spatial direction directly translates to the shape of the Jacobian matrix. One should note that although MATRICS operates in

a matrix-free way, the laws of linear algebra still need to be respected. For the more general non-symmetric case, the Biconjugate Gradient Stabilization (BiCGSTAB) or the Generalized Method of Residuals (GMRES) can be applied (Saad and Schultz [1986]).
Although BiCGSTAB has a small scaling factor, it makes use of two matrix-vector calculations per iteration as opposed to GMRES which has a scaling factor that grows with the number of iterations but makes use of only one matrix-vector product per iteration. Additionally, convergence is not guaranteed for BiCGSTAB, while for a system of size $N \times N$, GMRES becomes a direct method after N iterations. Monotonic convergence is guaranteed for GMRES (see Saad [2003] for details). Both methods are implemented in MATRICS. The implementation of the GMRES algorithm is listed in the appendix Section A.4.2. The implementation of BiCGSTAB is based on the work by Guennebaud et al. [2010] in the `C++ eigen` package.
It is a well-known fact that convergence of both methods can be accelerated in terms of needed iterations and numerical stability a great deal by using a suitable preconditioner (van der Vorst [2003], Pearson and Pestana [2020]). Since GMRES only needs one matrix-vector product and, hence, one preconditioning cycle per iteration and has guaranteed convergence, it is used as the standard solver throughout this work. A matrix-free, right preconditioned version of restarted GMRES is implemented by making use of Equation (2.41) and paying attention to the flexibility to use different preconditioners in the implementation.

Preconditioners

One of the implemented and mainly used preconditioners in this work is the Incomplete Lower Upper factorization with Threshold technique (ILUT), where the required matrix coefficients are calculated only approximately a priori on paper and are evaluated only once per time-step. As mentioned, an exact Jacobian matrix in components is difficult to compute, but an approximation to it is relatively easy to find. Since ILUT is of an approximative nature by definition, an approximation matrix computation imposes no further restriction. To compute the matrix components for ILUT, the advection terms are treated as if they would arise from the DC scheme.
The most important components, which should be recovered in a good approximation, are the diagonal entries arising from the time derivative since they ensure the nonsingularity of the matrix and the pressure gradient term. The latter is incorporated into the Jacobian matrix through the equation of state. Despite its great acceleration properties, there are two main drawbacks to the usage of ILUT. The first is its inherent sequentiality when computed, which, for larger systems, becomes an obvious issue. The second drawback is its dependence on individual components of the Jacobian matrix, which makes a complete matrix-free operation mode impossible and reduces the flexibility of the solver significantly.
To prohibit this, implement operation modes of GMRES and BiCGSTAB without preconditioning (Identity preconditioner) and with an approximate Jacobian precon-

ditioner are implemented. Here, only the advection term of each equation is considered, and the advection is treated as if it was done by first-order upwinding. In this case, the complete preconditioner becomes a prefactor $\frac{1}{1+\frac{\Delta t}{\Delta x}|v_{x,jk}|+\frac{\Delta t}{\Delta y}|v_{y,jk}|}$. Although this preconditioner reduces the GMRES iterations in the test cases only by about 10%[3], its low computational cost justifies its usage nevertheless. Although there exists a variety of more dedicated matrix-free preconditioners such as polynomial preconditioners (see e.g. Choquet [1995]) or multigrid preconditioners (Hackbusch [1985], Bastian et al. [2019]), their implementation is beyond the scope of this thesis and subject of future work.

2.3.4 Boundary Conditions

The most common choices for boundary conditions (BCs) in astrophysics are periodic-, reflective- and in- or outflow boundaries as well as sometimes constant boundaries. In the implementation in MATRICS, each boundary consists of two cells to model, e.g., inflow problems to second order. The representation of the boundary scheme in x- or radial direction is shown in Figure 2.2 for the inner (left) and outer (right) end of the simulation domain. The blue arrows indicate velocities belonging to the boundary cells and the black ones to the outermost domain cell. The vertical arrows represent the vertical velocity components, and the horizontal ones represent the radial components. The boundary scheme in k-direction is equivalent with the only difference of being rotated by 90°.

A detailed description of every type of BC is not given here since it can be found elsewhere. Nevertheless, a few peculiarities about the implementation of BCs with implicit methods that apply likewise to Dirichlet and von Neumann boundary conditions have to be addressed. The first is the treatment of the innermost velocity component of the domain, represented by the horizontal black arrow on the left-hand side of Figure 2.2. For reflective BCs, this value is always kept at zero despite being inside the solution domain. There are three ways to realize this. The first is to regularly solve the whole domain as described in Section 2.3.2 and to overwrite the value at every boundary update. The second option is to remove the solution procedure component, e.g., drop the respective index from Equation (2.42) altogether. The third option is to keep the component but set the value of μ_{BC} to zero and $x_{\mathrm{BC}}^{i+1} = x_{\mathrm{BC}}^{i}$ for all time-steps, thus keeping the velocity value constant to the initial value.

The first option has been shown to decrease the stability of the solution method substantially since it causes a decoupling of the used value for the boundary velocity from the solved one. This also influences the other variables' values since they are calculated as if flow through the boundary domain would be present. This is nothing but an error. The second option is technically difficult to implement and neglects the coupling to the boundary velocity, which also destabilizes the solution

[3]This is very problem dependent. In unfavorable conditions, the number of iterations may not be reduced at all.

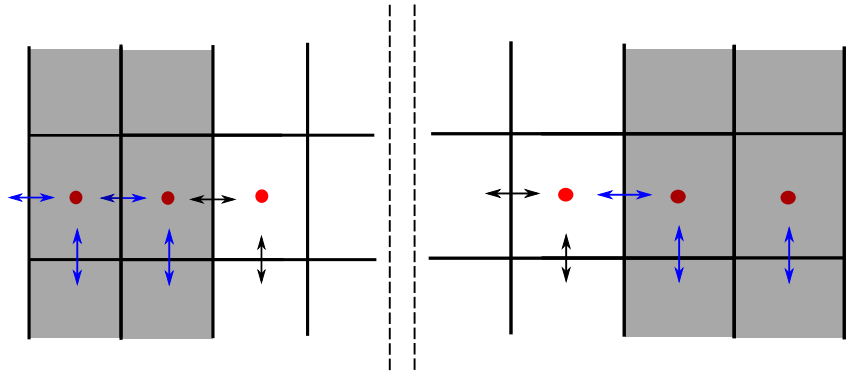


Figure 2.2: Boundary (grey) cells and last cell in the domain in the respective direction for opposite sides of the domain. This depiction is equivalently valid in Cartesian-, cylindrical- and spherical coordinates for both $j-$ and $k-$directions. The blue arrows indicate the position of velocities belonging to the boundary cells, and the black arrows are those belonging to the outermost domain cell.

procedure. The third method does not come with these disadvantages since, on the one hand, the velocity is kept constant as a result of the solution procedure itself, and coupling to the boundary velocity is included. A more detailed description of this approach can be found in Keil [2010].
An additional difficulty is positioning the boundary update in the solution procedure. One may update the boundaries once after every time step, once after every Newton iteration step, or once after every linear solver step (GMRES or BiCGSTAB iteration step). For a qualified choice, not only computational cost has to be considered. In the cases where not only the domain values depend on the BCs but also the BCs depend on domain values - as is the case for reflective boundary conditions - an additional cross dependency is introduced to the solution method when the BCs are updated inside the Newton- or the linear-solver steps. In principle, this must be resolved through an additional iteration loop. In practice, this was not observed to be necessary, and no noticeable difference in computational speed was noted. On the other hand, the difference in the solver's consequent stability is quite large. Because of this, the BC update is done inside every linear solver step.
Furthermore, one must consider the upwinding procedure's dependence on the BCs. While the DC scheme requires information only from one boundary cell, LUD, and QUICK require two boundary cells when the flow comes out of the boundary. Lastly, incorporating the boundary conditions in the Jacobian matrix is essential. This is already included in the realization through boundary update inside the linear solver step but is not necessarily for matrix-based methods and may cause substantial nu-

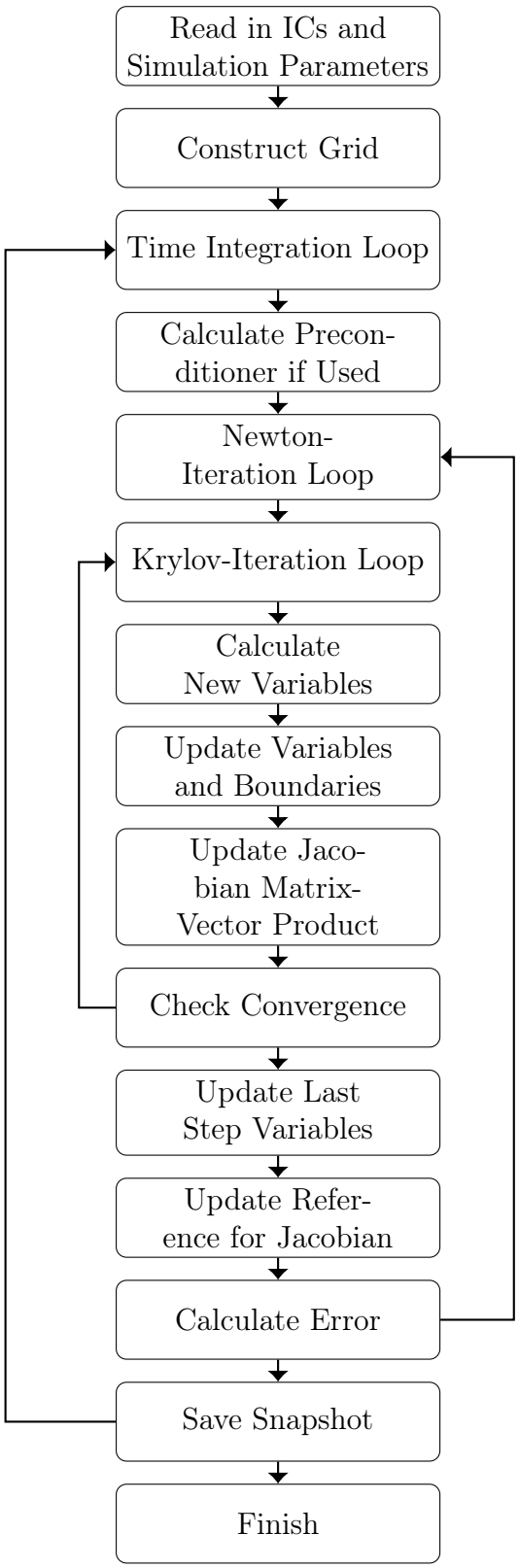


Figure 2.3: Schematic depiction of the main steps taken in the solution procedure of MATRICS.

merical limitations if not done correctly. In principle, one must consider boundary effects when ILUT is utilized and matrix components are computed. In practice, this was found negligible when the Courant number of the flow is not too large.
To summarize this section, a visual representation of the main steps and loops of the MATRICS solver routine is given in Figure 2.3. The expression "Update Reference Jacobian" in step 12 describes the process of updating the vector $L(\vec{x}^i, \vec{x^{i-1}})$ in Equation (2.41). Similarly, step 9, "Update Jacobian Matrix-Vector Product," is equivalent to the evaluation of the full right-hand side expression in Equation (2.41). This requires the preceding update of $\vec{x}^{i+1} = \vec{x}^i + \epsilon_m \vec{\mu}$, making an additional variable update inside the Newton-iteration step superfluous.

2.4 Test Problems

This section shows test cases for every component of the MATRICS code. In Cartesian coordinates, one-dimensional tests are performed to 1) illustrate the ability of the code to resolve sound waves when the Courant number is chosen sufficiently small, and to diffuse these waves out when a higher time-step is chosen, and 2) to show that analytic results for a physical problem can be reproduced. For the latter, the well-studied shock tube problem by Sod [1978] is performed. Contrary to its usual application, the focus is not on demonstrating the shock-capturing capabilities of the numerical method but on demonstrating the code's capability to deal with shocks when the Mach number is larger than unity.
To test the implementation of viscosity and the equations in cylindrical coordinates in 3D axisymmetry, as well as the boundary scheme, the flow between two rotating concentric cylinders, known as Taylor-Couette (TC) flow, is simulated. For the TC flow, analytic studies describing the onset of instability exist. With the reproduction of the analytic results for a one-dimensional solar wind problem, the spatial order of the schemes as implemented is proven to be second-order. The implementation in spherical coordinates alongside the implementation of gravity is tested with simulations of the vertical shear instability (VSI), where analytical growth rates are reproduced. The last test problem is of special interest in the context of this work since it is close to the intended use case of the code.

2.4.1 Linear Sound Wave

First, the compressible 1D Euler Equations in Cartesian coordinates, considering thermal energy only with the caloric equation of state

$$P = (\gamma - 1)\mathcal{E} = (\gamma - 1)\rho e, \tag{2.44}$$

are tested for an ideal gas with $\gamma = 1.4$ and specific internal energy e. A constant density profile $\overline{\rho} = 1$ that is distorted by a sinusoidal profile $\rho_0'(x) = 10^{-4} \cdot \sin(x)$ with small amplitude gives the initial condition for density $\rho_0(x) = \overline{\rho} + \rho_0'(x)$. In the adiabatic case, density and pressure are related via

$$P = K\rho^{\gamma}, \tag{2.45}$$

where the constant K is set to unity. From Equations (2.44) and (2.45), the specific internal energy is calculated and the internal energy $\mathcal{E}_0(x) = \rho_0(x)e_0(x)$ initialized. With the adiabatic speed of sound $c_s = \sqrt{\gamma\frac{P}{\rho}} = \sqrt{\gamma(\gamma-1)e}$ one obtains the initial fluid velocity for the sound wave and close the set of initial conditions with the momentum

$$m_0(x) = c_{s,0}(x) \cdot \rho_0(x) \quad . \tag{2.46}$$

Periodic boundary conditions are applied and the integration domain is set to $x \in [0, 2\pi)$ with 100 grid points. The continuity, momentum, and energy equations are solved simultaneously, and different time-steps and hence different CFL numbers are tested. Figure 2.4 shows the time evolution of the density amplitude for different CFL numbers where the LUD-scheme for spatial- and the BDF-1 scheme for temporal integration is used. ILUT preconditioned GMRES, restarted after 20 iterations is chosen as the linear solver.

The first feature one may notice is the sinusoidal oscillations present which are small in amplitude and can be explained with the setup of the initial conditions, where Equation 2.46 is used to calculate momentum at $x_{j-\frac{1}{2}} = \frac{x_j + x_{j-1}}{2}$. Internal computation the code is done as, $m(x_{j-\frac{1}{2}}) = \frac{m(x_j)+m(x_j-1)}{2} \neq m(\frac{x_j+x_{j-1}}{2})$. One can work out the difference in momentum easily to obtain $\mathcal{O}(m_{0,ics} - m_{0,code}) = 10^{-5}$, being in perfect agreement with the order of the oscillations observed here.
The noteworthy phenomenon is the overall trend observable for the different CFL numbers, where density fluctuation diminishes. This is to be expected as outlined in Section 1.2.2. The decrease is clearly super-linear for the CFL$>$ 1, where the averaging out of the fluctuations for CFL$\approx$ 12 is reached faster than for CFL$\approx$ 1.2. In Figure 2.5, the density fluctuations are plotted at $t = 50$, where the damping of the density fluctuations is even more obvious.

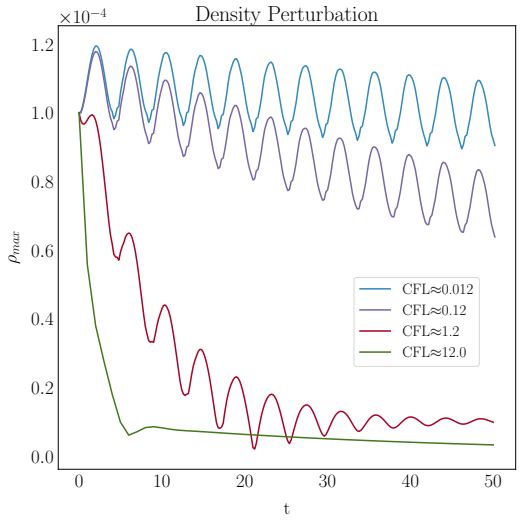


Figure 2.4: Evolution of the density perturbation in the sound wave test for different CFL numbers using 100 grid points, solving with the LUD- and BDF-1 schemes.

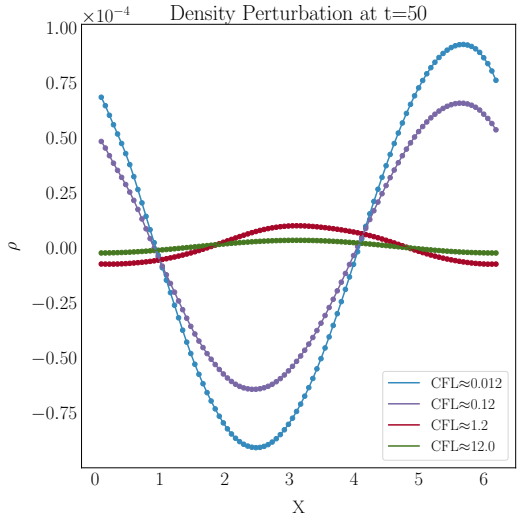


Figure 2.5: Density perturbation at $t = 50$ in code units for the sound wave test, where time integration is done at different CFL numbers. The simulation is done with 100 cells, 2nd order upwinding, BDF-1 time integration, and periodic boundary conditions.

With this, the code's ability to resolve sound waves and operate in the CFL>1 regime, where sound waves are diffused out with time, in dependence on the time-step size, is proven.

2.4.2 Sods Shock Tube

Although shock modeling is not a primary concern in this work, shocks can be used as a stress test to illustrate the limitations of the code. Sods shock tube problem (SSTP) is a well-studied problem with a known analytic solution (here taken from Toro [2009]). Since a shock is by definition a high Mach number flow, kinetic energy can not be neglected relative to thermal energy and has to be included in the energy equation as

$$\frac{\partial \mathcal{E}^{total}}{\partial t} + \nabla \cdot \left(\mathcal{E}^{total}\vec{v}\right) = -(P\nabla \cdot \vec{v} + \vec{v} \cdot (\nabla P)), \tag{2.47}$$

where the total energy is $\mathcal{E}^{total} = \mathcal{E} + \frac{\rho}{2}\vec{v} \cdot \vec{v}$, giving the ideal gas equation of state for compressible flows

$$P = \left(\mathcal{E}^{total} - \frac{1}{2}\rho\vec{v} \cdot \vec{v}\right)(\gamma - 1) \quad . \tag{2.48}$$

Reconstruction of the quadratic velocity term is done in accordance with Equation (2.38) as

$$\begin{aligned} \vec{v} \cdot \vec{v} \rightarrow & \frac{u^i_{j-\frac{1}{2},k} + u^i_{j+\frac{1}{2},k}}{2} \frac{u^{i-1}_{j-\frac{1}{2},k} + u^{i-1}_{j+\frac{1}{2},k}}{2} \\ & + \frac{v^i_{j,k-\frac{1}{2}} + v^i_{j,k+\frac{1}{2}}}{2} \frac{v^{i-1}_{j,k-\frac{1}{2}} + v^{i-1}_{j,k+\frac{1}{2}}}{2} \\ & + w^i_{j,k} w^{i-1}_{j,k}, \end{aligned} \tag{2.49}$$

i refers to the current Newton-iteration step and $i-1$ to the previous one. v is the velocity in the vertical direction, and w is the velocity perpendicular to the simulation plane. In this case of a one-dimensional problem, only the first component (velocity u in x-direction) is considered. The difference between the total equation of state and the thermal equation of state used to model the SSTP can be seen in Figure 2.6 from where it becomes obvious that the kinetic term can not be neglected when no other entropy generating terms such as von Neumann-Richtmyer(-Landshoff) viscosity (VonNeumann and Richtmyer [1950], Landshoff [1955]) is employed.

The initial conditions for the SSTP for density, thermal energy, and momentum on the domain $x \in [0, 1]$ are

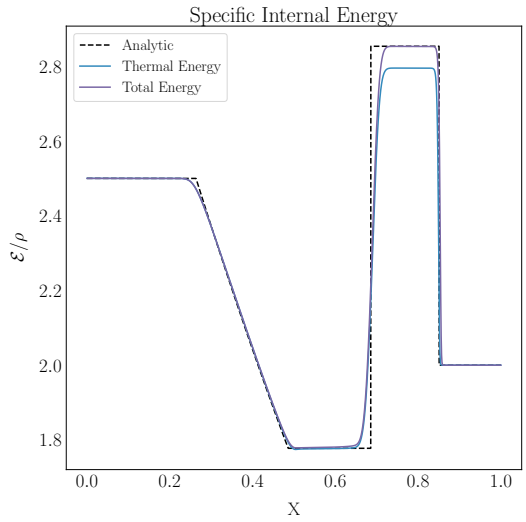


Figure 2.6: Comparison of the specific internal energy using total and thermal energy in the SSTP. 1000 grid cells are used, and integration is done using the QUICK and BDF-1 schemes at a CFL number ~ 0.5 where GMRES with ILUT preconditioning is used as the solver.

$$\rho(x,0) = \begin{cases} 1.0 & ,x <= 0.5 \\ 0.125 & ,x > 0.5 \end{cases}, \tag{2.50}$$

$$\mathcal{E}^{total}(x,0) = \begin{cases} \frac{1.0}{\gamma-1} & ,x <= 0.5 \\ \frac{0.1}{\gamma-1} & ,x > 0.5 \end{cases}, \tag{2.51}$$

$$m_x(x,0) = 0 \quad . \tag{2.52}$$

The Euler equations are solved with the equation of state in Equation 2.48 and $\gamma = 1.4$ and time is integrated to $t = 0.2$ using the BDF-1 scheme. The comparison of internal energy and velocity between the non-slope limited QUICK, the slope-limited QUICK, and the DC schemes are shown in Figure 2.7 where, as a reference, also the analytic solution from Toro [2009] is shown.

With the fully simultaneous implicit approach, the SSTP can be modeled with Courant numbers > 1 as shown in Figure 2.8 for simulations with slope limited QUICK using Courant numbers of up to 4. 1000 domain-cells are used in all cases. The cases with CFL numbers below two are solved using the standard Newton method and the CFL=4 case is obtained using Newton's method with a damping factor of $\lambda_{\text{damp}} = \frac{1}{2}$. On the one hand, the choice of $\lambda_{\text{damp}} < 1$ extends the convergence radius of Newton's method but, on the other hand, can result in a higher number of iterations needed for convergence. It can be seen in Figure 2.8 that, as a function of Courant number, the numerical diffusion of the numerical scheme grows.

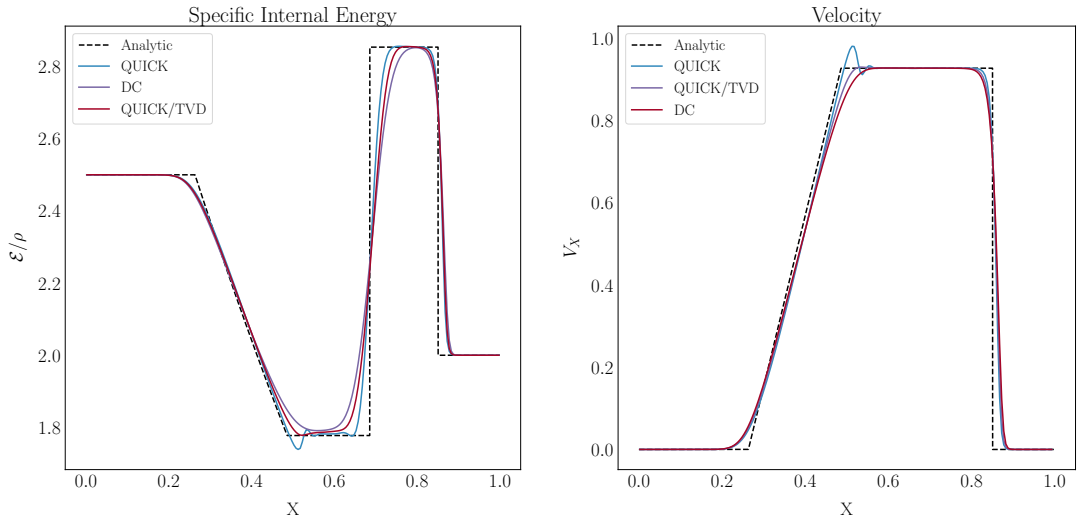


Figure 2.7: Specific internal energy (top) and velocity (bottom) for SSTP with 200 grid cells at time $t = 0.2$, integrated with BDF-1. The spatial scheme is indicated in the legend.

Table 2.1: Comparison of average L_1 errors with respect to density for the slope-limited version of QUICK and LUD, respectively, for different spatial resolutions.

Δx	QUICK $\langle L_1(\rho)\rangle$	LUD $\langle L_1(\rho)\rangle$
1/100	0.2392	0.2485
1/200	0.0414	0.0427
1/400	0.0101	0.0100
1/800	0.0029	0.0029

This effect becomes especially prominent beyond $\mathrm{CFL}_{\mathrm{adv}} > 1$ indicating that this is a reasonable constraint to self-impose.

This becomes even more obvious when looking at Figure 2.9, where instead of QUICK, the DC scheme and an otherwise equal set up is used to reach a higher Courant number of $\mathrm{CFL}_{\mathrm{adv}} = 14.5$. The stronger diffusion compared to the smaller Courant number can clearly be seen and is also present for more dedicated shock capturing methods, as, e.g., in Figure 6 by Fraysse and Saurel [2019], where a 10 times higher spatial resolution is used with different time-step sizes. The effect appears to be the same as for sound waves. This means that not only sound waves are diffused out when $\mathrm{CFL}_{\mathrm{cs}} > 1$, but also higher amplitude flows when $\mathrm{CFL}_{\mathrm{adv}} > 1$. The condition $\mathrm{CFL}_{\mathrm{adv}} \leq 1$ is thus imposed in all following runs.

An order evaluation of the slope-limited schemes is given in Table 2.1, using the average of the L_1 loss function over the whole domain as the measure for the spatial error. For the density, this average is defined as

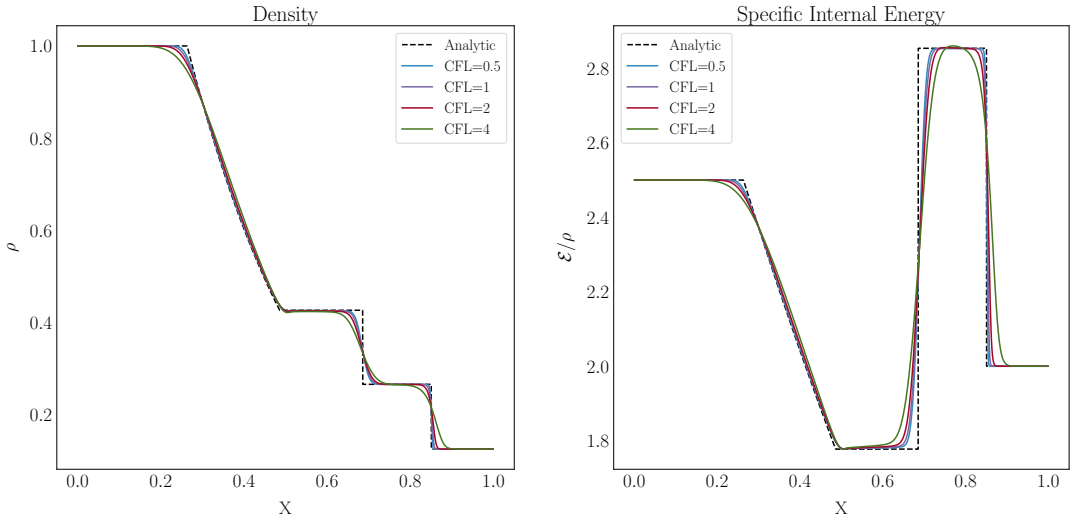


Figure 2.8: Density and Specific internal energy for SSTP with 1000 grid cells at time $t = 0.2$, integrated with backward Euler and different CFL numbers. The slope-limited version of QUICK is used.

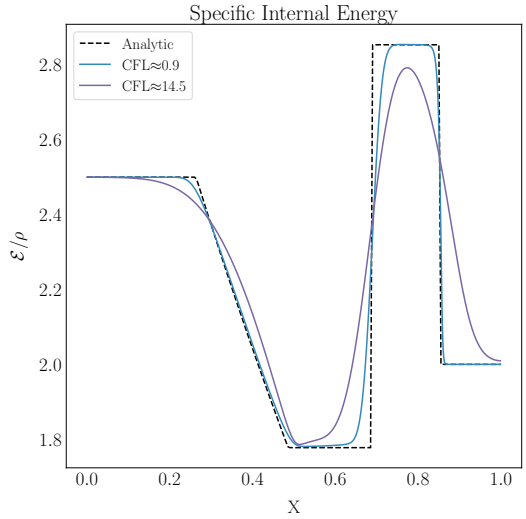


Figure 2.9: Specific internal energy density simulation of the SSTP with 1000 cells for two different CFL numbers. A total equation of state is used in both cases with first-order upwinding and the BDF-1 scheme. GMRES is selected as the solver.

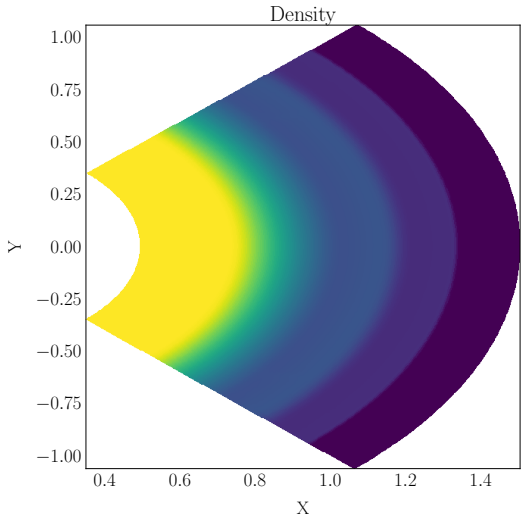


Figure 2.10: 3D-axisymmetric solution of Sods shock tube integrated with spherical coordinates to a dimensionless time of 0.2. 200^2 cells, an advective CFL number of 0.5, the slope-limited QUICK scheme, and the BDF-1 scheme are used. GMRES is selected as the solver.

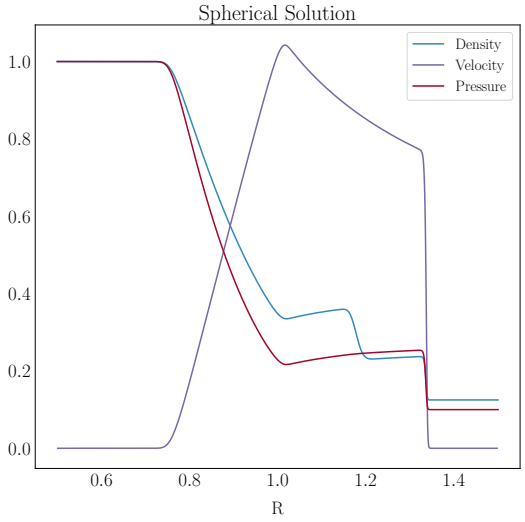


Figure 2.11: Solution of Sods shock tube integrated in spherical coordinates to a dimensionless time of 0.2. 200 cells, an advective CFL number of 0.5, the slope-limited QUICK scheme, and the BDF-1 scheme are used. GMRES is selected as the solver.

$$\langle L_1(\rho)\rangle = \frac{1}{N_x}\sum_{0}^{N_x}\left|\left(\rho_{\text{Simulated}} - \rho_{\text{Analytic}}\right)\right|, \tag{2.53}$$

where N_x is the count of cells. The obtained slope in log-log-space is with ~ -1.7 close to minus two as one would expect for the combination of the respective schemes and the superbee limiter, but due to the first order component at the discontinuity is not quite -2. The solution to Sod's shock tube, using the same initial conditions - except a shift by 0.5 in the positive r-direction - integrated using a spherical coordinate system in 3D-axisymmetry is shown in Figure 2.10 for density. Figure 2.11 shows profiles in the radial direction for density, velocity, and pressure. The results for density, velocity as well as pressure are very similar to those obtained by Balsara et al. [2020], who use a spherical geodesic mesh to model the problem as well as to Omang et al. [2006] who solve the problem using a Lagrangian method.

2.4.3 Taylor-Couette Flow

The flow between two concentric and independently rotating cylinders is called Taylor-Couette flow (TC). It is generally characterised by the radius ratio $\eta = \frac{r_\mathrm{i}}{r_\mathrm{o}}$, the gap width $d = r_\mathrm{o} - r_\mathrm{i}$ as well as the Reynolds numbers $Re_{\mathrm{i/o}} = \frac{\Omega_{\mathrm{i/o}} r_{\mathrm{i/o}} d}{\nu}$ and aspect ratio $\Gamma = \frac{h}{d}$ for the height of the simulation domain h as well as inner radius r_i and outer radius r_o, respectively. Based on these properties Andereck et al. [1986] describe different flow regimes where the laminar base-flow (Couette flow) has the well-known analytical solution

$$\begin{aligned} v_\varphi(r) &= A \cdot r + \frac{B}{r}, \\ A &= \Omega_\mathrm{i}\frac{\Omega_\mathrm{o}/\Omega_\mathrm{i} - \eta^2}{1-\eta^2}, \\ B &= \Omega_\mathrm{i} r_\mathrm{i}^2 \frac{1 - \Omega_\mathrm{o}/\Omega_\mathrm{i}}{1-\eta^2} \quad . \end{aligned} \tag{2.54}$$

The transition from Couette- to Taylor-vortex flow depends on the rotation relation between the inner and outer cylinders. In the case of a stationary outer cylinder, the Rayleigh criterion is violated, and the flow's stability depends on viscosity's stabilizing effect, which is usually expressed in terms of the Reynolds number. Since the original expression derived by Taylor [1923] (Schrimpf et al. [2021]) of the critical Reynolds number (Re_{crit}) in the small gap limit

$$Re_{crit} = 41.2 \cdot \sqrt{\frac{\eta}{1-\eta}}, \tag{2.55}$$

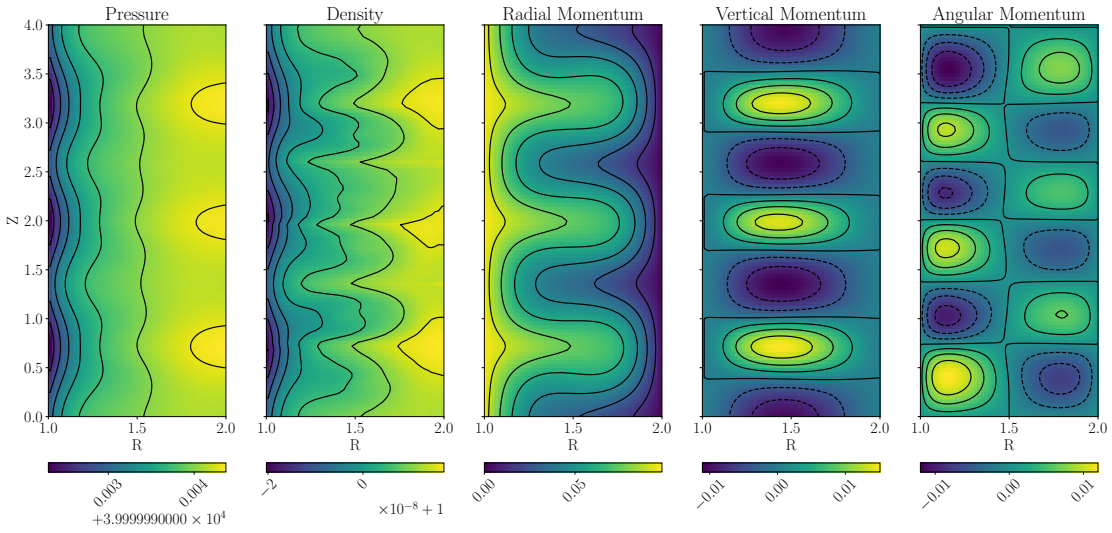


Figure 2.12: Contour lines of pressure, angular, radial, and vertical momentum together for the Taylor Vortex flow after 1.5 viscous time scales. The plots are created from the test run with a resolution of 100×200 and an inner rotation of $\Omega_\mathrm{i} = 0.1$ as well as the radius ratio $\eta = 0.5$ and aspect ratio $\Gamma = 4$ using periodic boundary conditions in the vertical direction.

many other approximations of the Taylor number as stability criterion have been calculated and/or experimentally obtained (e.g., by Snyder [1968], Recktenwald et al. [1993a] and Esser and Grossmann [1996]) for different radius-ratios. The aim of the test case, performed in this section, is to reproduce the critical Reynolds numbers found in the literature to verify the implementation of axisymmetric cylindrical coordinates using viscosity and centrifugal terms, employing reflective and periodic boundary conditions. For simplicity, the inner radius is fixed at $r_\mathrm{i} = 1$ and the kinematic viscosity is set to $\nu = 10^{-3}$. The wall of the outer cylinder is kept at rest ($\Omega_\mathrm{o} = 0$) and only the rotation Ω_i of the inner cylinder is varied.
As initial conditions, the angular momentum is set according to Equation (2.54), and the other momentum components are set to zero plus a small random fluctuation $\sim \mathcal{O}(10^{-5})$. From the equilibrium of pressure and centrifugal forces

$$\rho \frac{v_\varphi^2}{r} = -\nabla P, \tag{2.56}$$

together with the thermal equation of state in Equation (2.44), the specific internal energy $e = e(r)$ is obtained by using $\rho = \rho_0 = 1$ as

$$e(r) = C_1 + \frac{\frac{A^2 r^2}{2} + 2AB\log(r) - \frac{B^2}{2r^2}}{\gamma - 1}, \tag{2.57}$$

with $\gamma = 1.4$. When choosing $C_1 = 10^5$, the modeled flow is weakly compressible. This marks a difference from the theoretical descriptions, which assume full incompressibility, but from the experiments, density fluctuations are observed in the order of the numerical accuracy, which is 10^{-8} of the initial value of unity. No quantitative difference is observed when compressibility is reduced through a further increase in the initial specific internal energy. The results are comparable even when the continuity equation is deactivated altogether (resulting in constant density without enforcement of divergence freedom).
Reflective boundaries are enforced in the radial direction where Ω_i, and hence $v_{\varphi,\mathrm{i}}$, is kept fixed. The NSE (2.1) and (2.2), combined with the energy Equation (2.4) are solved, ignoring viscous heating. In the vertical direction, periodic boundary conditions are chosen. They are most suited to model the case of an infinitely long cylinder, as is aimed for. An aspect-ratio $\Gamma = 4$ is set up, giving a cell aspect ratio of $\Gamma_{cell} = 2$ and a radius ratio of $\eta = 0.5$. Simulations are conducted with resolutions 20×40, 50×100, 100×200, and 200×400 to find a good compromise between accuracy and computational time to conduct further runs with.
Following Recktenwald et al. [1993a], the onset of instability for the chosen radius ratio can be expected with $\Omega_\mathrm{in} \approx 0.68$, corresponding to $Re_{crit} \approx 68$. A slightly larger value of $\Omega_\mathrm{i} = 0.1$ is chosen and a time-step of $\Delta t = 0.2$ is enforced such that it can be used for all runs at $\mathrm{CFL}_\mathrm{adv} < 1$, which depending on resolution is $10^4 - 10^6$ that of the CFL number one would obtain when considering the sound speed. Time integration is done using the BDF-2, and spatial integration is done using the QUICK scheme. Typical results for this configuration's contours of pressure, density, and momentum components are shown in Figure 2.12 where pressure and density have roughly the same shape. The difference is due to the fact that the GMRES accuracy is chosen to be 10^{-8} and hence in the order of the observed density changes.
Figure 2.13 shows the temporal evolution of the maximal velocity inside the domain to $t = 1.5\tau_{vis}$ of the viscous time-scale $\tau_{vis} = \frac{d}{\nu}$. One can clearly see the steep growth for all resolutions starting at $0.4\tau_{vis} < t < 0.75\tau_{vis}$ with the lowest resolution starting slightly earlier. The growth phase is followed by a phase of relaxation at $0.75\tau_{vis} < t < \tau_{vis}$ and finally convergence to equilibrium at $t > \tau_{vis}$. The evolution tracks for the different resolutions converge relatively quickly and thus. It is thus proceeded with the 50×100 resolution.

Figure 2.14 shows the test for different aspect ratios at the chosen radial resolution and constant cell aspect ratio of $\Gamma_{cell} = 2$. Once again, one can see the solution converge for larger aspect ratios, as expected since the case of an infinitely long cylinder is approached. A clear outlier is the $\Gamma = 2$ case, where the velocity growth sets in at a later time, and the observed overshoot shortly before relaxation

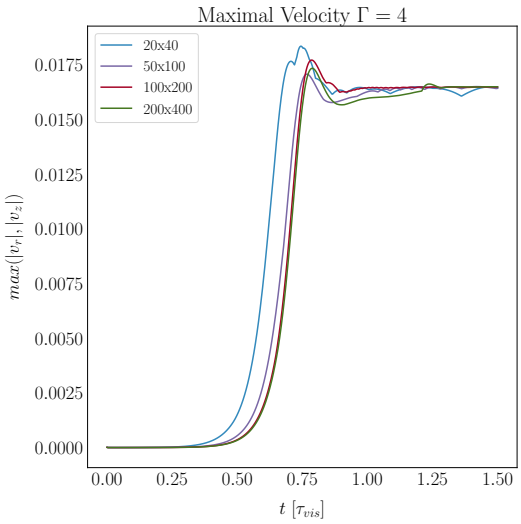


Figure 2.13: Temporal evolution of maximal velocity in the simulation domain for constant aspect ratio and different resolutions in the Taylor-Couette test problem. The QUICK and BDF-2 schemes are used and ILUT preconditioned GMRES is selected as solver.

is much higher. Both effects may be due to the supposed uneven count of modes in the simulation domain. A too-small domain aspect ratio, as well as a too-small resolution, artificially destabilizes the flow and is not suited for the determination of the critical Reynolds number. Likewise, a larger domain aspect ratio with higher resolution is computationally much more expensive.

As a compromise, the aspect ratio of $\Gamma = 4$ with a resolution of 50×100 seems sufficient for the main runs. These are conducted for the radius ratios $\eta = \{0.3, 0.5, 0.7, 0.8, 0.9, 0.975\}$. Multiple runs are conducted for each η to obtain the respective critical Reynolds number Re_{crit} approximately through manual iterations. Re_{crit} is obtained for each radius ratio by choosing Ω_i in the vicinity of the value predicted by Recktenwald et al. [1993a], integrating to $t = 2\tau_{vis}$ and checking if considerable growth of the maximal velocity inside the domain, can be observed. Ω_i is varied until the onset of instability is sufficiently constrained.
The results of the simulations are depicted in Figure 2.15, where the theoretical predictions by Recktenwald et al. [1993a] and Taylor [1923] are also plotted. The blue area is constrained by the criterion explained above. A much smaller resolution of 20×40 is chosen for the orange area. Integration is done to $t \approx 1000\tau_{vis}$ and it is checked at the end if the velocity inside the domain has vanished, which below the orange area is the case and above it is not. The red crosses mark the radius ratios of the conducted simulations. As one can see, the runs with qualified guesses about the required resolution and aspect ratio to model the case of an infinitely long cylinder (blue) are in perfect agreement with theoretical predictions by Recktenwald et al. [1993a] and for the gap width approaching the small gap limit also with those by

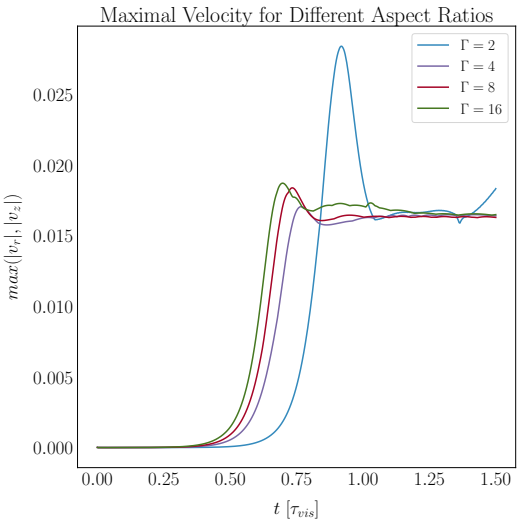


Figure 2.14: Temporal evolution of maximal velocity in the simulation domain for resolution and different aspect ratios in the Taylor-Couette test problem. The QUICK and BDF-2 schemes are used and ILUT preconditioned GMRES is selected as solver. The resolution is selected as 50×100.

Taylor [1923]. As expected from the first tests, the smaller resolution runs (orange) exhibit a more unstable behavior while still being in agreement with theory.

2.4.4 Vertical Shear Instability

The Vertical Shear Instability (VSI, Urpin and Brandenburg [1998], Nelson et al. [2013]) is believed to be a driver of turbulence in protoplanetary discs (PPDs) appearing in vertically stratified PPDs with a radial variation in its temperature profile and sufficiently short cooling time. For the modeling, initial conditions following Nelson et al. [2013] and Manger and Klahr [2018] are set up in equilibrium such that centrifugal forces exactly counter pressure and gravitational forces in the radial direction and pressure counters gravity in the vertical direction. The midplane ($z = 0$) density and temperature are assumed to follow the radial profile

$$\rho(R, Z = 0) = \rho_0 \left(\frac{R}{R_0} \right)^p , \tag{2.58}$$

$$T(R, Z = 0) = T_0 \left(\frac{R}{R_0} \right)^q \quad . \tag{2.59}$$

Combining these equations with the balance of forces one can work out the density and rotation profiles using the isothermal pressure density relation $P = c_s^2 \rho$ to

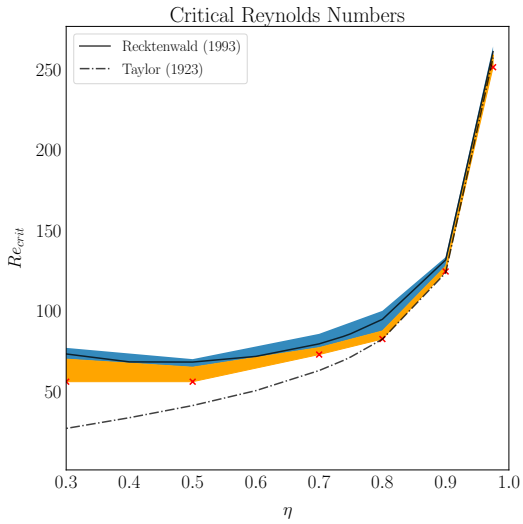


Figure 2.15: Experimentally derived ranges for the critical Reynolds numbers in the Taylor-Couette test problem for different radius ratios using high-resolution simulations to 2 viscous time scales (blue) and low-resolution simulations to 2000 viscous time scales (orange) in comparison to predictions by Recktenwald et al. [1993b] (solid line) and Taylor [1923] (dotted line). The orange area marks the range where instability is obtained exclusively from the lower-resolution runs, and no instability for the higher-resolution runs was found. The red crosses mark the positions of the lowest value of the critical Reynolds number for the obtained radius ratios.

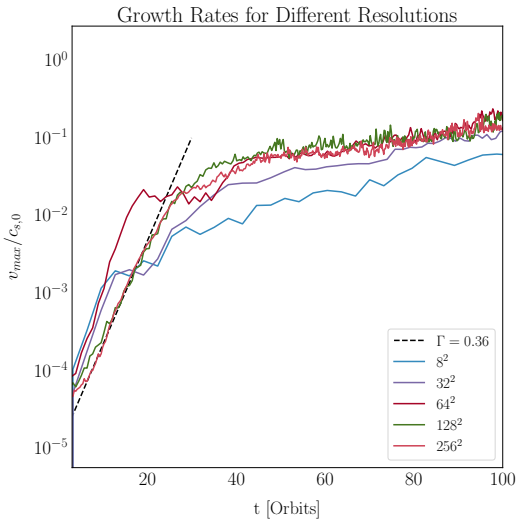


Figure 2.16: VSI growth rates for different resolutions as a function of the number of orbits covered compared to $\Gamma = 0.36$ growth. A fit to the 128^2 and 256^2 runs yields a growth rate of $\Gamma = 0.32$ and for the two lower resolution runs $\Gamma = 0.4$. In all cases, the QUICK and BDF-2 schemes are used.

$$\rho(R, Z) = \rho_0 \left(\frac{R}{R_0}\right)^p \exp\left[\frac{GM}{c_s(R)^2}\left(\frac{1}{r} - \frac{1}{R}\right)\right], \tag{2.60}$$

$$\Omega(R, Z) = \Omega_K(R)\left[1 + q - \frac{qR}{r} + \frac{Rc_s(R)^2 \cdot (q+p)}{GM}\right]^{1/2}, \tag{2.61}$$

with Keplerian rotation $\Omega_K = \sqrt{\frac{GM}{R^3}}$, spherical radius $r = \sqrt{R^2 + Z^2}$, gravitational constant G, central mass M and sound speed $c_s(R)^2 = c_0^2 \left(\frac{R}{R_0}\right)^q$. The angular momentum is set as $L(R, Z) = \rho(R, Z) \cdot \Omega(R, Z)R^2$ and a small random fluctuation in the order of 10^{-5} is added on it. Radial and vertical momenta are set to zero, and parameters equivalent to Manger et al. [2020] and Klahr et al. [2023] are chosen as $q = -1$, $p = -1.5$, and $GM = 1$.
A quadratic simulation box is defined in spherical coordinates using the pressure scale height ($H = \frac{c_s}{\Omega}$) as size. The box is located at one scale height above the midplane and radially centered at $R_0 = 1$. Spherical coordinates with reflective no-slip boundary conditions are used as was done in a similar way by Klahr et al. [2023]. Equations (2.1) and (2.2) are solved, without viscosity. The system is closed with the isothermal pressure density relation $P(R) = \rho(R)c_0^2 \left(\frac{R}{R_0}\right)^q$ using $c_0 = 0.1$. The energy Equation (2.3) is deactivated in the runs and the time-step is chosen such that the convective Courant number does not exceed unity. The initial time-step for the low resolution runs is $dt = 1$ and for the high resolution runs a factor of 5

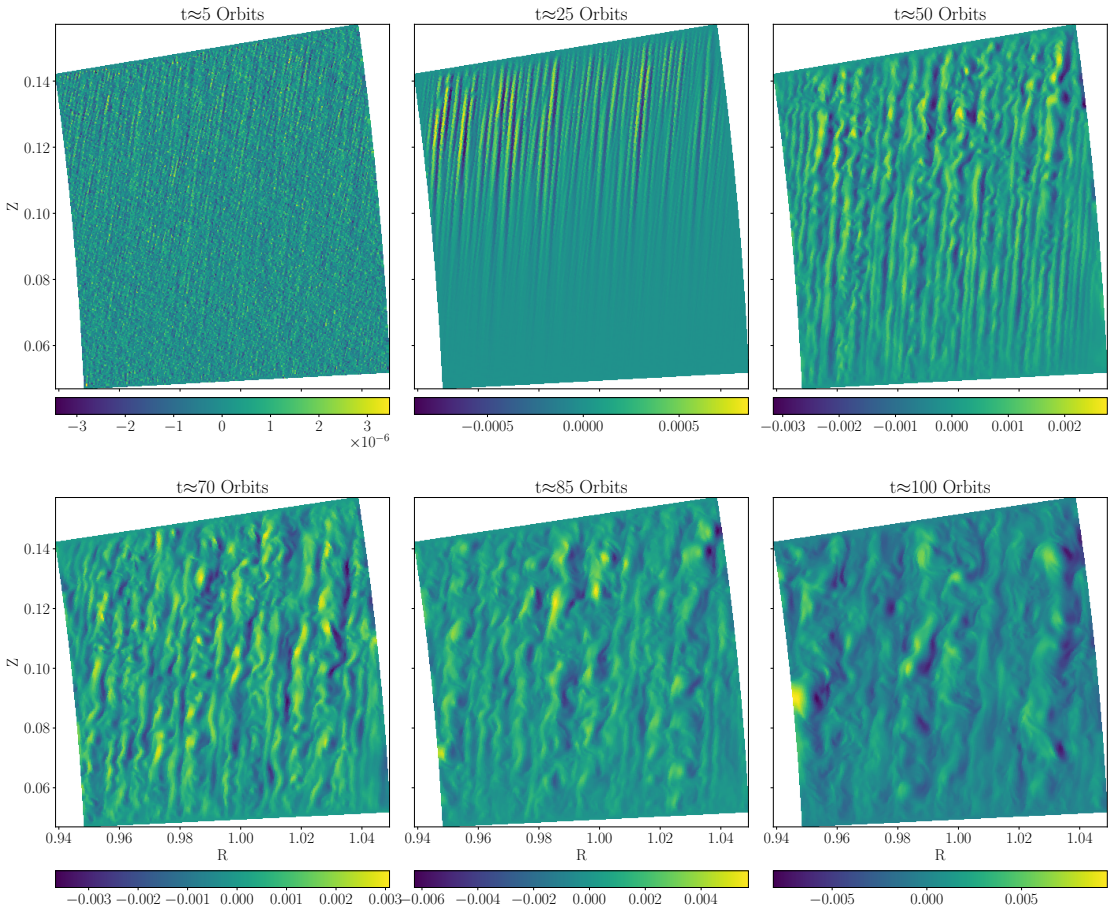


Figure 2.17: Visualization of the vertical velocity component in the 256^2 resolution run for the VSI at different orbital times. The QUICK and BDF-2 schemes are utilized and GMRES is selected as solver.

lower. Runs with different resolutions ranging from 8^2 to 256^2 are conducted.
The two lower resolution runs appear to have a slightly steeper growth rate of $\Gamma \approx 0.4$ than the two higher resolution runs with $\Gamma \approx 0.32$. In addition, settlement to steady state with $v_{max}/c_{s,0} = 0.1$ can be observed for all runs, which is equal to that found by Manger et al. [2020]. Figure 2.17 shows the field of the θ-component of the velocity for six different orbital times for the highest resolution run (256^2). The expected VSI typical pattern is building up in the first few orbits and decaying into larger scales with time. The formation of vortices starting at less than 50 orbits similar to those in Klahr et al. [2023], is observed. The formation of nonlinear patterns is additional proof of the correctness of the implementation of Newton's method.
The results agree with previous work using the well-tested PLUTO code and analytical considerations that validate the implementation in spherical coordinates, including central gravity.

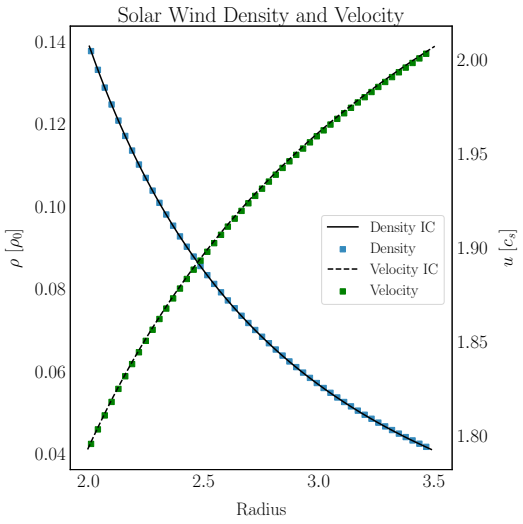


Figure 2.18: Semi-analytic solution (black lines) and simulated data points (squares) for the solar wind test problem at $t = 2.5$. BDF2 is used for temporal integration alongside the LUD scheme for spatial discretization. The simulation was conducted with 200 cells, and every fourth was used for plotting. Density corresponds to the left- and radial velocity to the right axis.

2.4.5 Parker's Solar Wind

Last, the order of the spatial reconstruction in spherical coordinates is verified. Similarly to Balsara et al. [2020] a simple model of a stationary solar wind, introduced by Parker [1965], is used. The model expresses the balance of pressure-gradient-, central gravitational- and inertial force. The radial component of the stationary continuity equation in spherical coordinates is with radial velocity u

$$\frac{1}{r^2}\frac{\partial\left(r^2\rho u\right)}{\partial r} = 0 \tag{2.62}$$

$$\Leftrightarrow -u\frac{\partial\rho}{\partial r} = \rho\frac{\partial u}{\partial r} + \frac{2\rho u}{r} \tag{2.63}$$

$$\Rightarrow \rho = \rho_0\left(\frac{u_0}{u}\right)\left(\frac{r_0}{r}\right)^2, \tag{2.64}$$

where the last step is obtained by the separation of variables and integration. The momentum equation using the described equilibrium assumptions is

$$\underbrace{\frac{1}{r^2}\frac{\partial\left(r^2\rho u^2\right)}{\partial r}}_{=u\rho\frac{\partial u}{\partial r}\text{ (using Equation (2.63))}} = -\frac{\partial P}{\partial r} - \rho\frac{\partial\Phi_g}{\partial r}. \tag{2.65}$$

Table 2.2: Comparison of average L_1-errors (see Equation 2.53) and resulting spatial order with respect to density for the slope-limited LUD ($\kappa = -1$) for different spatial resolutions.

Δx	LUD $\langle L_1(\rho) \rangle$	$\mathcal{O}(\Delta x)$
1/100	1.37×10^{-5}	—
1/200	3.2×10^{-6}	2.1
1/400	8.14×10^{-7}	1.975

Adopting a polytropic equation of state $P = K\rho^\gamma$ with $\partial_r P = \gamma\rho^{\gamma-1}\partial_r\rho$, dividing Equation (2.65) by ρ, and radially integrating yields

$$\frac{u^2 - u_0^2}{2} = -\frac{\gamma}{\gamma - 1}\left(\rho^{\gamma-1} - \rho_0^{\gamma-1}\right) + GM\left(\frac{1}{r} - \frac{1}{r_0}\right). \tag{2.66}$$

Equation (2.45) can then be used to find a density-independent formulation of the momentum equation. Setting all constants except the polytropic index to unity i.e $\rho_0 = r_0 = 1$, one obtains $u_0 = c_s = \sqrt{\gamma}$ and arrives at

$$\frac{u^2}{2} + \frac{\gamma}{\gamma - 1}u^{1-\gamma}r^{2-2\gamma} - \frac{GM}{r} - \frac{\gamma}{\gamma - 1} - \frac{\gamma}{2} + \frac{3}{2} = 0. \tag{2.67}$$

Together with Equation (2.64) and the polytropic equation of state, this closes the system.
The continuity equation and the momentum equation are solved in spherical coordinates and radial direction only, using the BDF-2 scheme and the slope-limited κ-scheme with $\kappa = -1$ (LUD). Similar to Balsara et al. [2020], the domain $r \in [2, 3.5]$ is chosen. A numerical solution to Equation 2.67, is found using `Pythons SymPy` package (Meurer et al. [2017]) and used to calculate the ICs for radial velocity and density. The adiabatic index is set to $\gamma = 1.4$ and gravity to $GM = 1$. Equation (2.67) admits two solutions in the chosen radial domain, one corresponding to a radially decreasing velocity and the other to a radially increasing velocity. Since the former is unphysical, the latter solution is selected. In this solution, the velocity is supersonic throughout the domain.
The outer boundary is chosen as free-flow (in and outflow allowed with continuative or zero-gradient approximation), and the inner boundary is constant in time. After some initial numerical relaxation, no transient time evolution is expected. At sufficiently large times, all information about the initial conditions (except those that are engraved in the constant boundary) is lost, and the problem is completely defined by the conditions imposed on the constant boundary. Integrate is done, using a convective CFL number of 0.5, to the dimensionless time of $t_{end} = 2.5$, corresponding to two sound crossing times. The results for the error are shown in Table

2.2, where the implemented scheme can be seen to be second order. Here, the error is expressed in terms of the average L_1 loss function over the whole domain. The obtained density and radial velocity, alongside their semi-analytic values, are shown in Figure 2.18.

2.5 Conclusion

In this chapter, the novel globally implicit, time-unsplit, and versatile upwinding-based finite-volume hydrodynamic solver MATRICS is presented. The spatial discretization schemes are outlined, and the time integration methodology is explained. The general solution procedure with the incorporated linear system solvers, which rely on a matrix-free formulation of the system, is presented. The matrix-free approach, through the Newton-Krylov method, guarantees easy expandability and easier implementation of boundary conditions in the code, which has traditionally been difficult for implicit methods. With tests in different implemented coordinate systems and dimensionalities, the operability, accuracy, and correctness of the code and the implementation of the physical mechanisms considered are proven.
Section 2.4.1 proves that the code damps sound waves at high Courant numbers but resolves them at lower ones. Section 2.4.2 shows that the method is able to resolve shocks without producing spurious oscillations close to second order and that, similar to the damping of sound waves, shocks can be damped at high advective Courant numbers. Furthermore, Sod's test in spherical coordinates is conducted and reproduces results from the literature. With 3D axisymmetric simulations in cylindrical coordinates using resolutions high enough to model the considered physics correctly and including kinematic viscosity, the applicability of the method is illustrated. Moderate computational effort suffices to model the complex example of the Taylor-Couette flow in Section 2.4.3. In Section 2.4.4, the isothermal growth rate of the VSI in 3D-axisymmetric spherical coordinates is reproduced successfully at a time step beyond that needed to resolve sound waves. The expected physical results are obtained, which don't rely on resolving sound waves. Finally, in Section 2.4.5, a one-dimensional test in spherical coordinates is performed to prove our method's second-order accuracy and its correctness in spherical coordinates.
The next step in the development process is to improve the parallelization of the solver, where the focus in the short term is on `OpenMP` parallelization. In addition to that, future versions of the code will be built on more dedicated matrix-free preconditioners. Initial tests suggest that a combination of geometric multigrid and GMRES is promising.

3 The Goldreich-Schubert-Fricke Instability in Spherical Geometry

As the introduction mentions, the Goldreich-Schubert-Fricke instability, henceforth GSF, follows a similar analytical description as the VSI. Analogous to the VSI in protoplanetary disks, it is believed to be a driver of turbulence and angular momentum transport in the radiative zone of stellar interiors. In contrast to PPDs, stars are mainly supported by pressure, not rotation. Consequently, the Mach number of flows is much lower than in PPDs. Furthermore, the assumption of optically thin Newtonian cooling does not hold inside stars' optically thick environment, and thermal diffusion is responsible for thermal relaxation. A third difference is geometry. Typically, initial conditions for VSI studies are assessed as marginally stable equilibrium solution to the coupled system of vertical- and radial-momentum equations in cylindrical coordinates. This relies on the assumption of temperature being constant on cylinders, which does not hold in stellar interiors, where the temperature is almost constant on spherical shells.
Following a short introduction in Section 3.1, this chapter derives a simplified solar model considering gravity, rotation, and pressure in spherical coordinates (Section 3.2). The parametrizations used are chosen to be as close to the cylindrical prescriptions used in VSI studies as possible. At the same time, the spherical nature of the environment is preserved. The strength of rotation support in the environment is introduced with a free parameter. Section 3.3 describes the equations solved and the initial conditions chosen within this chapter in detail. Section 3.4 presents numerical results for the isothermal and thermally relaxed evolution of the GSF and compares them to analytical findings.
The aim of the chapter is to give numerical proof that the GSF can operate in the pressure-supported environment of stellar interiors, where thermal relaxation is mediated by diffusive processes. Numerical modeling in an environment like this is particularly difficult because the Mach number of the flow in a pressure-supported environment is generally low and the effect under study, namely the GSF, is very susceptible to viscosity (Caleo et al. [2016]). The former necessitates very small time steps in explicit methods, which is numerically expensive and, as illustrated in Section 1.2.2, increases numerical diffusion. Implicit methods are not bound to the Courant criterion enforced by the speed of sound and a time step that is oriented at the Mach number can be chosen. As a result, numerical diffusion is minimized (see Figure 1.2).
Additionally, implicit methods are not restricted in time step size through diffusion operators, which also limit the time step size in explicit methods. This is particularly important in this chapter since low Prandtl numbers are, by definition, the ratio

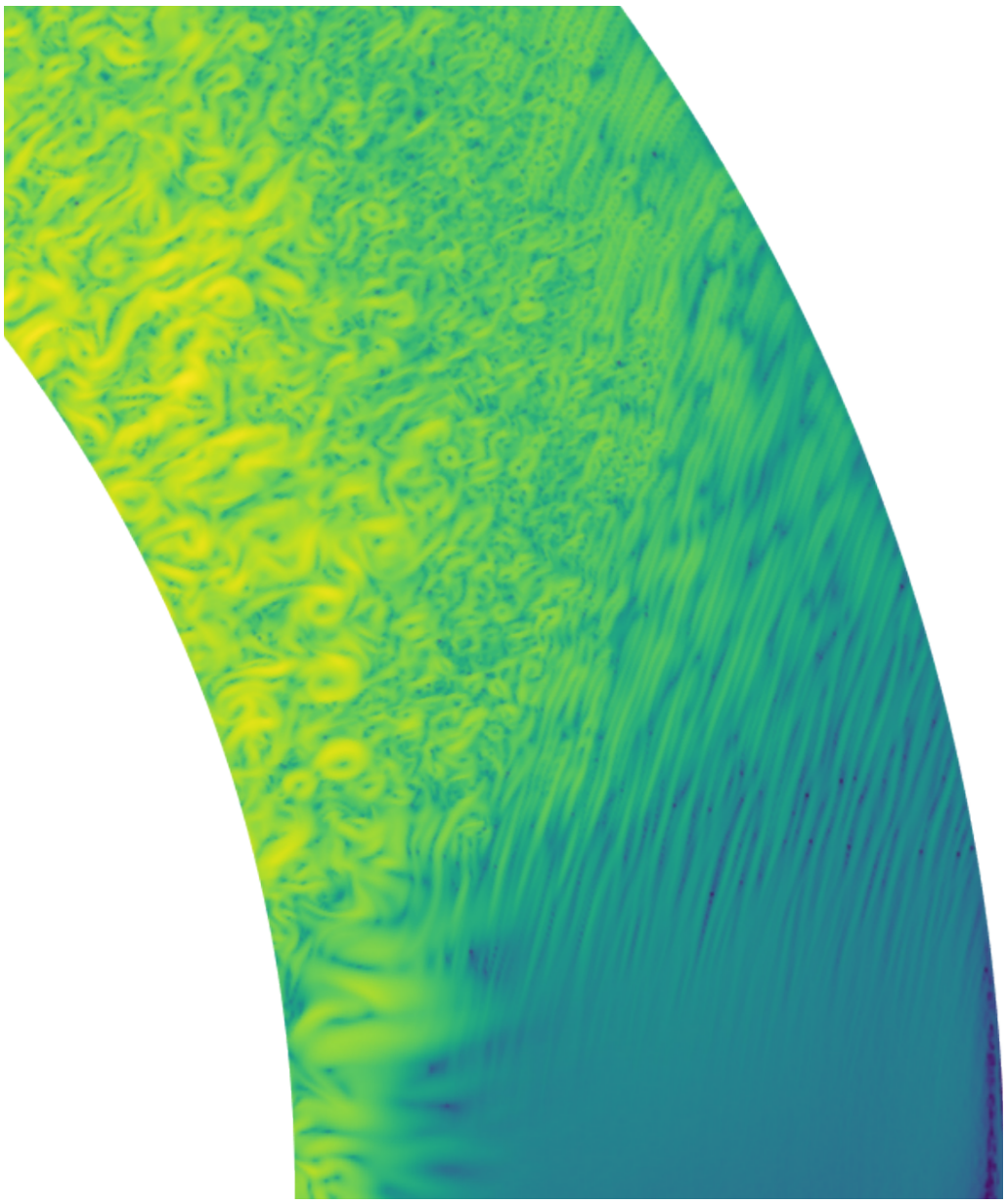


Figure 3.1: Modeled flow pattern in a stellar radiative zone in terms of $\sqrt{v_r^2 + v_\theta^2}$, starting from a slightly perturbed state of marginal stability between gravity, pressure, and rotation after $\sim$ 1 year. In the spherical radial direction, 2.7 pressure-, and 1.5 density scale heights are resolved. Simulation is carried out with the MATRICS code using $200_r \times 512_\theta$ cells and vastly super-explicit time steps ($\Delta t \approx 430 \Delta t_{explicit}$). The convective overstability forms in regions with strong shear close to the equator. At higher latitudes, different evolutionary stages of the Goldreich-Schubert-Fricke instability are visible. Inner radial regions at higher latitudes appear turbulent. Simulation is carried out at a Prandtl number of $Pr = 10^{-4}$. The full image can be found in Figure 3.8a.

of high thermal diffusion coefficients to lower kinematic viscosity coefficients. Time steps that are restricted by the thermal diffusion coefficient (see Equation (1.54) also cause higher numerical diffusion, and are thus adulterating the effectively operating Prandtl number. With its operability in 3D axisymmetric spherical coordinates and its extensibility to additional terms, the MATRICS code is not only by its nature of being globally implicit but also by the geometry and usability the perfect choice for the modeling tasks within this chapter. Figure 3.1 gives an impression of the topic in this chapter. The figure shows not only a visually appealing depiction of the GSF and COS but also a look-ahead at the main results within this chapter.

3.1 Introduction

The GSF (Goldreich and Schubert [1967], Fricke [1968]) is an axisymmetric instability arising under baroclinic conditions, implying the presence of vertical shear, and when buoyancy is not strong enough to stabilize a fluid parcel displacement against its marginally stable position. Despite being theoretically first described more than five decades ago, the role it plays in the transport of angular momentum (Aerts et al. [2019]) as well as chemical mixing (Siess et al. [2004]) in stellar interiors is not yet fully understood. Historically, GSF has been examined under the assumption of negligible viscosity (James and Kahn [1971], Korycansky [1991]), and stellar evolution codes (e.g., Paxton et al. [2013]) are relying on parametrization building on this assumption (Heger et al. [2000]).
On the one hand, asteroseismic observations reveal a discrepancy in the inferred angular momentum transport from these models (Cantiello et al. [2014]); on the other hand, theoretical considerations show the high susceptibility of GSF to background viscosity (Caleo et al. [2016], Caleo and Balbus [2016]). Because of this, more recent numerical studies analyze the effect of different Prandtl numbers ($Pr = \nu/\chi$) - the ratio of kinematic viscosity ν to thermal diffusivity χ -, and find the instability to only occur in regions of very strong shear (Barker et al. [2019], Caleo et al. [2016]). Adjacent to the simulations conducted in these studies, which rely on a local Boussinesq approximation (Barker et al. [2019], Barker et al. [2020], Dymott et al. [2023]), fluid dynamics simulations of the GSF solving the full non-linear set of NSE have not been carried out so far. This work presents the very first 3D-axisymmetric global GSF simulations carried out in a pressure-supported environment, inspired by the upper layer of the solar radiative zone.

3.2 Semi-Hydrostatic Solar Model With Weak Rotation

This section introduces a simplified solar model. The aim is to find a prescription close to hydrostatic equilibrium that allows for a stable configuration with rotation. The prescription of a model like this is unintuitively complicated since a (rigidly) rotating body can not simultaneously be in thermal and hydrostatic equilibrium. This is the von Zeipel paradox, which on the spatially global scale in stellar interiors is resolved by the famous Eddington-Sweet circulation (Eddington [1929], Sweet [1950]). When modeling the solar interior, either this fact is embraced and hydrostatic equilibrium conditions are set up - enforcing the presence of a thermal wind - or the hydrostatic equilibrium is relaxed to allow for a rotation configuration without the presence of a geostrophic- or thermal wind. The calculations outlined in this section follow the latter approach.

3.2.1 Temperature and Density Parametrization

As is done in many VSI studies, an equilibrium solution can be found in the isothermal case by equating gravitational-, centrifugal- and pressure forces in cylindrical-radial as well as gravitational- and pressure forces in the vertical direction (e.g. Nelson et al. [2013], Manger et al. [2020]). Similar to *the VSI approach*, a radial parametrization for temperature is chosen, but it is set to vary with spherical radius r instead of cylindrical R against a reference radius r_0, at which the temperature takes a reference value T_0, with slope q as

$$T(r) = T_0 \left(\frac{r}{r_0} \right)^q . \tag{3.1}$$

The corresponding isothermal speed of sound is

$$c_s^2(r) = c_0^2 \left(\frac{r}{r_0} \right)^q . \tag{3.2}$$

Additionally, a spherically radial stratification of density with slope p is chosen as

$$\rho(r, \theta) = \rho_0(\theta) \left(\frac{r}{r_0} \right)^p . \tag{3.3}$$

3.2.2 Hydrostatic Equilibrium

The equilibrium of gravitational and pressure forces can be computed based on the parametrization described above. Throughout this chapter, self-gravity is neglected. When, in the first step, rotation is also neglected, hydrostatic equilibrium in the spherical-radial direction is

$$\partial_r \Phi_g = \frac{1}{\rho(r,\theta)} \partial_r P(r,\theta) \tag{3.4}$$

$$\Leftrightarrow -\frac{GM}{r} = \frac{c_0^2}{\rho(r,\theta)} \left(\frac{r}{r_0}\right)^q \left[\partial_r \rho(r,\theta) + \frac{q}{r}\rho(r,\theta)\right] \tag{3.5}$$

$$= c_0^2 \left(\frac{r}{r_0}\right)^q \left[\frac{1}{\rho(r,\theta)} \partial_r \rho(r,\theta) + \frac{q}{r}\right]. \tag{3.6}$$

The isothermal equation of state $P = c_s(r)^2 \rho(r,\theta)$ has been used and the assumptions $q \neq -1$ and $p \neq -1$ are made. GM is assumed constant. When additionally the density parametrization in Equation (3.3) is enforced, this becomes

$$-\frac{GM}{r^2} = \frac{c_0^2}{r} \left(\frac{r}{r_0}\right)^q (p+q). \tag{3.7}$$

The solution requires $p + q = -\frac{GM}{c_0^2 r} \left(\frac{r}{r_0}\right)^{-q}$. This, in turn, implies that either p or q or both vary with radius, but this invalidates the derivation above since it was implicitly assumed that $p \neq p(r)$ and $q \neq q(r)$ when the pressure derivation is formed. This paradoxical situation can be resolved when the derivation is done with q or p being a function of radius. Without further justification, $p = p(r)$ is chosen, and the temperature profile is kept as in Equation (3.1) with a constant value for q. A density profile that is based on Equation (3.3) then reads

$$\rho(r,\theta) = \rho_0(\theta) \left(\frac{r}{r_0}\right)^{p(r)} \tag{3.8}$$

$$\Rightarrow \partial_r \rho(r,\theta) = \frac{\rho_0(\theta)}{r} \left(\frac{r}{r_0}\right)^{p(r)} \left[r \log\left(\frac{r}{r_0}\right) \partial_r p(r) + p(r)\right]. \tag{3.9}$$

When Equation (3.9) is combined with Equation (3.6), one obtains the first order linear ordinary differential equation

$$-\frac{GM}{r^2} = \frac{c_0^2}{r} \left(\frac{r}{r_0}\right)^q \left[r \log\left(\frac{r}{r_0}\right) \partial_r p(r) + p(r) + q\right]. \tag{3.10}$$

The solution has a rather cumbersome form but can be evaluated analytically in

a straightforward way to

$$p(r) = \underbrace{\frac{1}{\log\left(\frac{r}{r_0}\right)}\left[\frac{GM}{c_0^2(q+1)r}\left(\frac{r}{r_0}\right)^{-q} - q\log(r) + p_0\right]}_{\equiv p_{h,eq}(r)}. \tag{3.11}$$

p_0 is the integration constant. The expression is not valid at $r = r_0$. As Equation (3.9) also relies on $\partial_r p(r)$, one also needs

$$\begin{aligned}\partial_r p_{h,eq}(r) = {} & r^{-2}\log^2\left(\frac{r}{r_0}\right)\left[r\left(q\log(r) - q\log\left(\frac{r}{r_0}\right) - p_0\right)\right] \\ & - r^{-2}\log^2\left(\frac{r}{r_0}\right)\left[\frac{GM}{c_0^2(q+1)}\left(\frac{r}{r_0}\right)^{-q}\left((q+1)\log\left(\frac{r}{r_0}\right) + 1\right)\right].\end{aligned} \tag{3.12}$$

Of course, a hydrostatic equilibrium solution can be found more intuitively with much less calculus, when the prescription in Equation (3.8) is not enforced and density is treated as an arbitrary function. No claim is made here that the chosen prescription is optimal[1]. Still, it is consistent with what is done in the cylindrical case and can be evaluated analytically. It also allows for intuitive control of the strength of rotation support, as shown below.

3.2.3 Equilibrium With Rotation

In a cylindrical coordinates system, rotation acts through the centrifugal force in the **coordinate**-radial direction, only. In a spherical coordinate system, this is not the case. Since a non-advecting equilibrium solution is sought, Coriolis forces are neglected. Using the same convention as in Chapter 2, where the angle θ is taken with respect to the equatorial plane, the centrifugal acceleration is

$$\Omega^2(r,\theta)R\vec{e}_r = \Omega^2(r,\theta)r\cos\theta\left[\cos\theta\vec{e}_r - \sin\theta\vec{e}_\theta\right]. \tag{3.13}$$

The gravitational acceleration acts in the spherical radial direction as $\nabla\Phi_g = -\frac{GM}{r^2}$. The pressure force acts in the radial and polar directions and is expressed in components as

[1] A parametrization of exponential form would likely lead to a much simpler expression for the hydrostatic equilibrium solutions $p_{h,eq}(r)$ and $\partial_r p_{h,eq}(r)$ as the shape of the ODE for hydrostatic equilibrium implies.

$$\partial_r P = c_0^2 \left(\frac{r}{r_0}\right)^q \left[\frac{\partial \rho(r,\theta)}{\partial r} + \rho(r,\theta)\frac{q}{r}\right] \vec{e}_r, \tag{3.14}$$

$$\frac{\partial_\theta P}{r} = c_0^2 \left(\frac{r}{r_0}\right)^q \frac{1}{r}\frac{\partial \rho(r,\theta)}{\partial \theta}\vec{e}_\theta. \tag{3.15}$$

The equilibrium of forces in the spherical radial direction thus reads

$$\Omega^2(r,\theta) r \cos^2\theta + \frac{GM}{r^2} + c_0^2 \left(\frac{r}{r_0}\right)^q \frac{1}{\rho(r,\theta)} \left[\frac{\partial \rho(r,\theta)}{\partial r} + \rho(r,\theta)\frac{q}{r}\right] = 0, \tag{3.16}$$

and can be evaluated to give the rotation profile as

$$\Omega(r,\theta) = \frac{1}{\cos\theta}\left\{\frac{GM}{r^3} + \frac{c_0^2}{r^2}\left(\frac{r}{r_0}\right)^q \left[q + p(r) + r\log\left(\frac{r}{r_0}\right)\partial_r p(r)\right]\right\}^{\frac{1}{2}}. \tag{3.17}$$

The polar equilibrium of forces

$$-\Omega^2(r,\theta) r\cos\theta\sin\theta = c_0^2\left(\frac{r}{r_0}\right)^q \frac{1}{r}\frac{1}{\rho(r,\theta)}\frac{\partial \rho(r,\theta)}{\partial \theta}, \tag{3.18}$$

evaluates to

$$\frac{1}{\rho(r,\theta)}\frac{\partial \rho(r,\theta)}{\partial \theta} = \tan\theta B(r) \tag{3.19}$$

$$\Leftrightarrow \frac{1}{\rho_0(\theta)}\frac{\partial \rho_0(\theta)}{\partial \theta} = \tan\theta B(r), \tag{3.20}$$

$$\Rightarrow \rho_0(\theta) = \rho_{0,0}\cos^{-B(r)}(\theta). \tag{3.21}$$

Here the substitution

$$B(r) = -\left\{\frac{GM}{c_0^2 r^2}\left(\frac{r}{r_0}\right)^{-q} + \frac{1}{r}\left[q + p(r) + r\log\left(\frac{r}{r_0}\right)\partial_r p(r)\right]\right\}, \tag{3.22}$$

is made. This assumes that $p = p(r) \neq p(r,\theta)$, an assumption, that might not be desired in all cases. The reason for this is that Equation (3.17) directly depends on the cosine. Since in solar interiors, shellular rotation is often assumed (e.g., Zahn [1992], Meynet and Maeder [1997]), one would like to find p so that the dependence on the cosine vanishes. This is only possible when p also depends on the inclination.

3.2.4 Controlling Rotation

The considerations above describe equilibrium solutions in radial- and polar directions for the rotational- and density profiles under the condition of a spherically radial prescription for the pressure. The defining parameter is the radial slope of the density profile $p(r)$ and its derivative $\partial_r p(r)$. With the choice $p(r) = p_{h,eq}(r)$, the rotation in Equation (3.17) vanishes. Since the density can be assumed to decrease throughout stellar interiors with radius and rotation can not be stronger than gravity[2], $p(r)$ can be chosen in the bounds $p_{h,eq}(r) < p(r) < 0$. A value of $p(r)$ closer to $p_{h,eq}(r)$ implies a setup that is influenced more by pressure, and $p(r)$ closer to zero implies a setup that is more strongly influenced by rotation.
It makes sense to choose $p(r)$ as a superposition of $p_{h,eq}(r)$ and a small and positive distortion $\delta_p(r)$ such that the above conditions are fulfilled. In this case, one can write

$$p(r) = p_{h,eq}(r) + \delta_p(r), \tag{3.23}$$
$$\partial_r p(r) = \partial_r p_{h,eq}(r) + \partial_r \delta_p(r), \tag{3.24}$$

and the rotational profile becomes

$$\Omega(r,\theta) = \frac{c_0}{r\cos\theta}\left(\frac{r}{r_0}\right)^{q/2}\left[\delta_p(r) + r\log\left(\frac{r}{r_0}\right)\partial_r\delta_p(r)\right]^{\frac{1}{2}}. \tag{3.25}$$

Likewise, density, pressure, and entropy can be expressed with Equations (3.23), (3.24), (3.22), (3.21), (3.8) and (3.2) as

$$\rho(r,\theta) = \rho_{0,0}\cos^{D(r)}(\theta)\left(\frac{r}{r_0}\right)^{p_{h,eq}(r)+\delta_p(r)}, \tag{3.26}$$
$$P(r,\theta) = \underbrace{c_0^2\rho_{0,0}}_{P_0}\cos^{D(r)}(\theta)\left(\frac{r}{r_0}\right)^{p_{h,eq}(r)+\delta_p(r)+q}, \tag{3.27}$$
$$S(r,\theta) = P(r,\theta)\rho^{-\gamma}(r,\theta) \tag{3.28}$$
$$= \underbrace{c_0^2\rho_{0,0}^{1-\gamma}}_{S_0}\cos^{-D(r)(\gamma-1)}(\theta)\left(\frac{r}{r_0}\right)^{(p_{h,eq}(r)+\delta_p(r)+q)(1-\gamma)}. \tag{3.29}$$

Here, the substitution $D(r) = \frac{c_0^2}{r}\left(\frac{r}{r_0}\right)^q\left[\delta_p(r) + r\log\left(\frac{r}{r_0}\right)\partial_r\delta_p(r)\right]$ has been used.

[2]This would result in the rotational profile in Equation (3.17) being a complex-valued function and the consequent violation of the first Solberg-Høiland criterion when pressure is still present or the violation of the Rayleigh criterion when pressure is assumed zero.

3.3 Methodology

The Compressible Navier-Stokes-Equations (NSE) are solved in 3D-axisymmetry, in a conservative form

$$\partial_t \rho + \nabla \cdot (\rho \vec{v}) = 0, \tag{3.30}$$
$$\partial_t \vec{m} + \nabla \cdot (\vec{m} \otimes \vec{v}) = -\nabla P - \rho \nabla \Phi_{grav} + \nabla \cdot (\mu(\nabla \vec{v} + (\nabla \vec{v})^T), \tag{3.31}$$
$$\partial_t E + \nabla \cdot (E \vec{v}) = -P \nabla \cdot \vec{v} + \chi \rho \nabla \cdot \nabla T, \tag{3.32}$$

using the MATRICS code. Here, the thermal energy is defined as $E = \rho T$ and the momentum as $\vec{m} = \rho(v_r, v_\theta, r\cos\theta v_\varphi)^T$. Φ_{grav} is the gravitational potential, $\mu = \rho\nu$ the dynamical viscosity with kinematic viscosity ν. χ is the thermal diffusivity.

3.3.1 Code Units and Schemes

The conditions under which the GSF is studied here are aimed at being at least close to the upper regions of the solar radiative zone. It thus makes sense to choose the code units close to solar units. The rotation is fixed to a value in the vicinity of solar rotation in the upper radiative zone $\Omega_0 = \frac{2\pi}{(30 \text{ days})}$, the mass to the solar mass $M_\odot$ and the radius as the solar radius $R_\odot$. The resulting unit velocity is $1\Omega_0 R_\odot \approx 1.7$ km/s. The mean solar density is chosen as reference density $\rho_0 \sim 1$ g/cm$^3 = 0.17 M_\odot / R_\odot^3$ and $r_0 = 0.7 R_\odot$ as reference radius.
The MATRICS code is used throughout the numerical studies in Section 3.4. In all cases, temporal integration is done with the second-order accurate BDF-2 scheme and spatial upwinding with the QUICK scheme. GMRES is chosen as the linear solver. Reflective boundary conditions are employed, and the time step is chosen so that sound waves are not resolved.

3.3.2 Initial Conditions

Initial conditions are set up in four different simulation boxes located at four different inclinations $\theta \in \{0, \pi/4, \pi/6, \pi/8, \pi/12, \pi/2 - 0.08\}$ inside a shell in spherical geometry. The reference radius is chosen as $r_0 = 0.7$ and the reference speed of sound $c_0 = 0.5$. The gravitating factor $GM = 1$ is taken as constant throughout the simulation domain, and self-gravity is neglected. The temperature slope is fixed at $q = -1.5$. These parameters don't match those that one would expect in solar interiors. Especially the speed of sound and gravitational constant are vastly underestimated[3]. However, the chosen set of parameters is less intense to model and sufficient for this chapter's academic study of the GSF. The specific choice of parameters is because it enlarges the fluid's compressibility and thus its susceptibility to rotation while keeping the ratio GM/c_0^2 as a proxy for the strength of gravity

[3]One would expect $GM \approx 45 \times 10^3$, $c_0 = 100$, and $q \approx -0.6$.

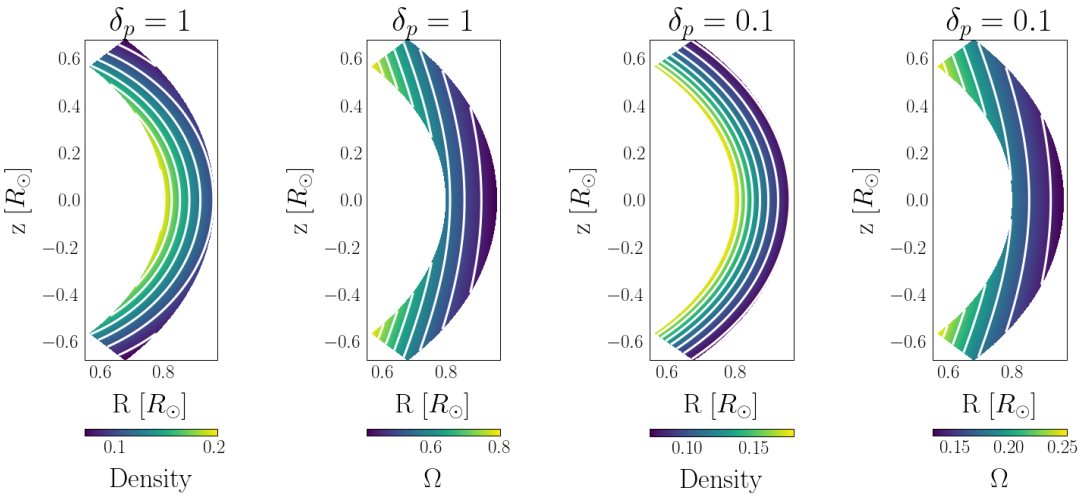


Figure 3.2: Density and rotation maps with respective iso-lines (white) for the prescription described in Section 3.2.3. Two different disturbance values δ_p as in Equation (3.33) are shown. The iso-lines in density indicate the oblateness of the object under examination.

relative to the pressure almost constant.
Since the rotational profile in Equation (3.25) is only valid if $\delta_p(r) + r \log\left(\frac{r}{r_0}\right) \partial_r \delta_p(r) > 0$, the minimal radius in the simulation domain is chosen as $r_{min} = r_0 + 0.1$. The radial extend is chosen to map one spherical-radial pressure scale height. The same numerical value is then chosen in the polar direction (in radians). As driving for rotation, $\delta_p(r)$ is chosen as

$$\begin{aligned} \delta_p(r) &= \delta_{p,0}, \\ \partial_r \delta_p(r) &= 0, \end{aligned} \tag{3.33}$$

where throughout this work $\delta_{p,0} = 1$ is enforced. The justification for this is that the chosen prescription makes evaluation of the rotation- and density- profile easy. With the described set of parameters, $p_{h,eq}(r)$ in Equation (3.11) can be evaluated to be $-5 \lesssim p_{h,eq}(r) \lesssim -3$ so its absolute value is larger than $\delta_{p,0}$ throughout the domain. This is a necessary condition, as otherwise, the pressure would increase with the radius.
Figure 3.2 shows the profiles for density and rotation (corresponding density profiles extending to the north-pole can be found in the Appendix Figure B.1). A small velocity perturbation in v_r and v_θ in the order of 10^{-5} is applied since otherwise, the initial conditions are observed to maintain their state of marginal stability - as expected. The simulation boxes for the initial conditions are depicted in Figure 3.3.

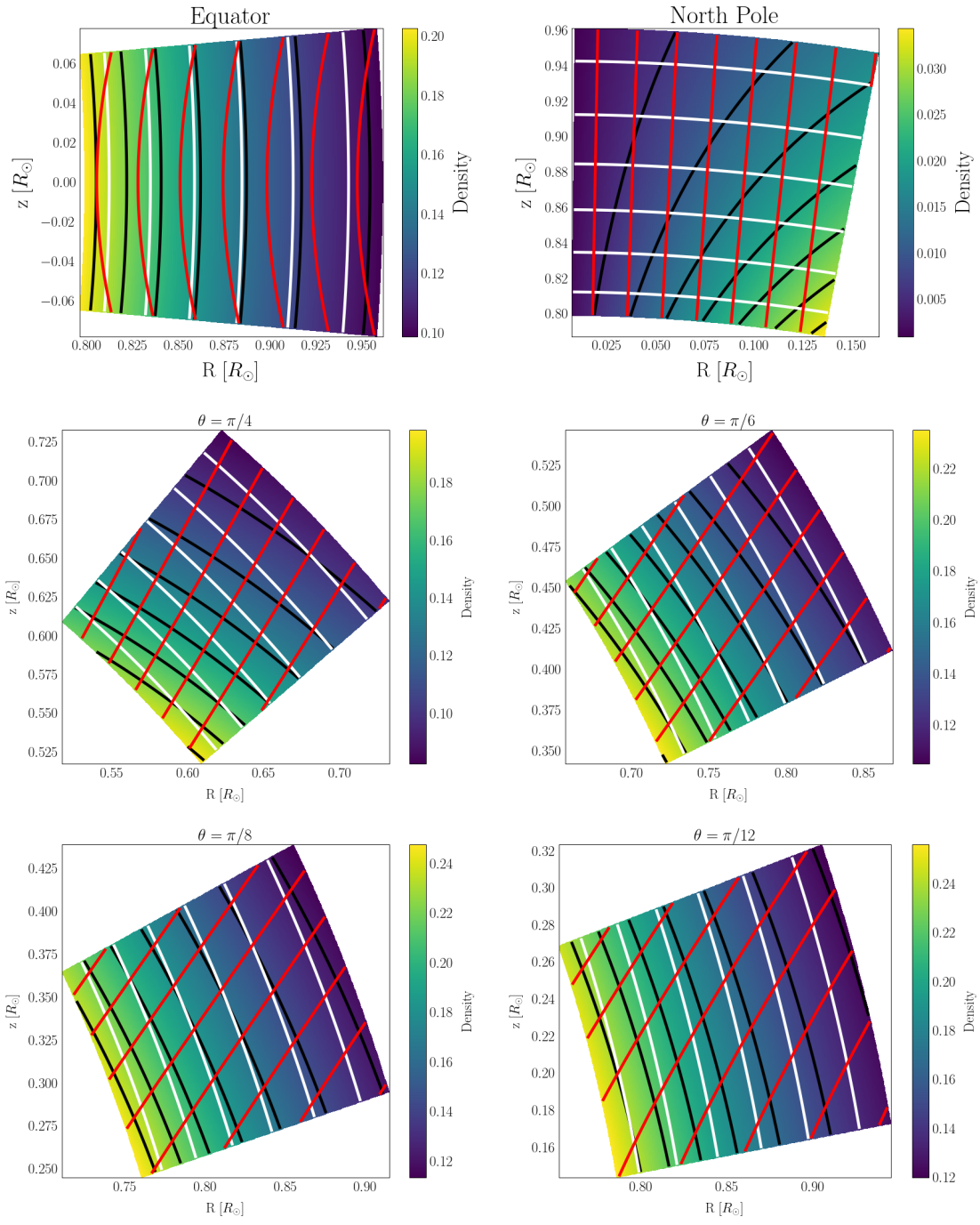


Figure 3.3: Depiction of Initial Conditions as set up for simulations. Color is density. The lines are the respective iso-lines of pressure (black), temperature (white), and specific angular momentum (red). Parameters are chosen as described in Section 3.3.1. The specific angular momentum increases with the cylindrical radius. The pressure-, and temperature iso-lines are inclined in all cases. Inclinations are with respect to the equator.

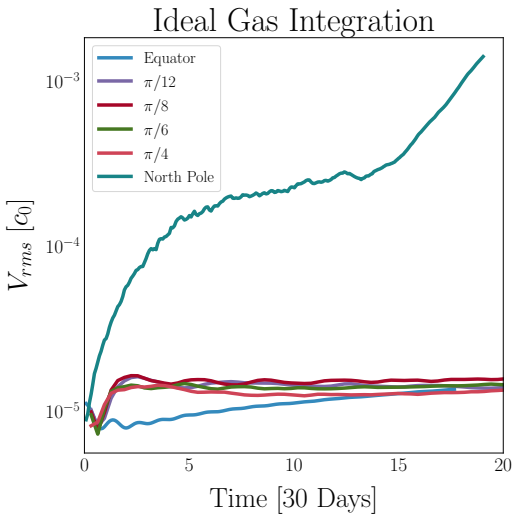


Figure 3.4: Growth rates in terms of the rms-velocity for ideal gas integration of perturbed initial conditions at different inclinations. The simulation box at the north pole appears to be convectively unstable. For all other inclinations, the initial perturbation relaxes at a constant value, indicating stability under adiabatic conditions.

The resolution is 128^2 in all simulation runs and the initial time step is chosen as $\Delta t = 1$, corresponding to $430\Delta t_{explicit}$, which would be enforced by the speed of sound. The time step is limited in all simulation runs to a Courant number of 0.5, so it decreases automatically at later times when instability operates and produces corresponding levels of velocity within the simulation domain.

3.3.3 Stability of Initial Setup

The expressions for density (Equation (3.26)), pressure (Equation (3.27)), and entropy (Equation (3.29)) follow from the choice of $\delta_p(r) = 1$ and the profile of rotational support as provoked through Equations (3.23) and (3.24). Using the constant $C^2 = \frac{c_0^2 \delta_{p,0}}{r_0^q}$ one obtains

$$\rho(r,\theta) = \rho_{0,0} \cos^{C^2 r^{q-1}}(\theta) \left(\frac{r}{r_0}\right)^{p_{h,eq}(r)+1}, \tag{3.34}$$

$$P(r,\theta) = P_0 \cos^{C^2 r^{q-1}}(\theta) \left(\frac{r}{r_0}\right)^{p_{h,eq}(r)+1+q}, \tag{3.35}$$

$$S(r,\theta) = S_0 \cos^{-C^2 r^{q-1}(\gamma-1)}(\theta) \left(\frac{r}{r_0}\right)^{(p_{h,eq}(r)+1+q)(1-\gamma)}. \tag{3.36}$$

There is a clear analogy to the respective profiles in cylindrical geometry (assessed, e.g., by Nelson et al. [2013]). Nevertheless, the choice of parameterizations in Equations (3.2) and (3.3), and the subsequent necessary calculus, especially the radial dependence of the density slope $p = p(r)$, makes the evaluation of the Solberg-Høiland criteria (discussed in Section 1.1.4)

$$\mathcal{N}_z^2 + \mathcal{N}_R^2 + \kappa_{ep}^2 > 0, \tag{3.37}$$

$$\frac{\partial P}{\partial z}\left(\frac{\partial S}{\partial z}\frac{\partial L^2}{\partial R} - \frac{\partial S}{\partial R}\frac{\partial L^2}{\partial z}\right) < 0, \tag{3.38}$$

a rather inconceivable task and it is not exercised here. Without this, the analytical evaluation of the stability of the initial condition setup can not be done.
A more pragmatic verification, if a less rigorous one, is the computational integration of the initial conditions as described in Section 3.3.2, including an initial velocity perturbation ($v_{r,\theta} \approx 10^{-5}$), under ideal gas conditions. To do so, the set of Equations (3.30), (3.31) and (3.32) is solved, neglecting kinematic-, and thermal diffusivity and closing the system with the caloric equation of state $P = (\gamma - 1)E$. Figure 3.4 shows the resulting growth rates.
The initial conditions at the north pole appear to be unstable, even under these conditions. This is not too surprising given the extreme angle between iso-lines of pressure and temperature, combined with a shallow slope of specific angular momentum as shown in the top-right of Figure 3.3. The rms velocities for the other tested inclinations converge to their initial value within the time of interest. This indicates the stability of the setup at all inclinations except at the north pole. Exemplary, the morphology of the flow is shown in Figure B.2 by means of the radial velocity component for the polar-, and equatorial plane, and for inclinations $\theta = \pi/4$ and $\theta = \pi/8$. Departure from equilibrium can be clearly seen in the simulation box close to the north pole. The flow at the other inclinations appears quiescent. Although this numerical evaluation constitutes no proof of the stability of the initial conditions, it strongly indicates this.

3.3.4 Shear in Initial Conditions

The rotational profile with the density distortion mentioned in Section 3.3.2 can be directly computed from Equation (3.25) as

$$\Omega(r,\theta) = \frac{c_0}{r\cos\theta}\left(\frac{r}{r_0}\right)^{q/2}\sqrt{\delta_{p,0}} \tag{3.39}$$

$$\Rightarrow \Omega(R,z) = \underbrace{\frac{c_0\sqrt{\delta_{p,0}}}{r_0^{q/2}}}_{=C=const.}\frac{\sqrt{R^2+z^2}^{q/2}}{R}. \tag{3.40}$$

The consequent expressions for the cylindrically-radial- and vertical shear are

$$\begin{aligned}\partial_R\left[\Omega(R,z)R^2\right] &= C\frac{qR^2+2(R^2+z^2)}{2}(R^2+z^2)^{q/4-1}\\ &= C\frac{q\cos^2(\theta)+2}{2}r^{q/2},\end{aligned} \tag{3.41}$$

$$\begin{aligned}\partial_z\left[\Omega(R,z)R^2\right] &= C\frac{qRz}{2}(R^2+z^2)^{q/4-1}\\ &= C\frac{q}{2}\sin(\theta)\cos(\theta)r^{q/2}.\end{aligned} \tag{3.42}$$

Also useful for later considerations is the epicyclic frequency in the cylindrical radial direction as well as its vertical analog. These can be calculated as

$$\kappa_R^2 = \frac{1}{R^3}\partial_R\left[\Omega(R,z)R^2\right]^2 = 2\frac{\Omega(R,z)}{R}\partial_R\left[\Omega(R,z)R^2\right] \tag{3.43}$$

$$= C^2\frac{(R^2+z^2)^{q/2-1}}{R^2}\left(qR^2+2(R^2+z^2)\right) \tag{3.44}$$

$$= C^2\frac{q\cos^2\theta+2}{\cos^2\theta}r^{q-2} \tag{3.45}$$

$$\kappa_z^2 = \frac{1}{R^3}\partial_z\left[\Omega(R,z)R^2\right]^2 = 2\frac{\Omega(R,z)}{R}\partial_z\left[\Omega(R,z)R^2\right] \tag{3.46}$$

$$= \frac{C^2qz}{R}(R^2+z^2)^{q/2-1} \tag{3.47}$$

$$= C^2q\tan\theta r^{q-2} \tag{3.48}$$

Notably, the vertical shear and the resulting vertical epicyclic frequency are zero at the equator. The radial shear is never zero for $r > 0$ and $q > -2$.

3.4 Simulation Results with MATRICS

In this section, the developed model is put to the test. In Section 3.4.1, the initial conditions described in Section 3.3.2 are integrated using an isothermal equation of state. In Section 3.4.2, the energy equation is solved with a caloric equation of state, and thermal relaxation is realized corresponding to the optically thick regime. Different thermal diffusivities are tested. In Section 3.4.3, two of the thermalized simulations are extended to a larger simulation box.

3.4.1 Isothermal Growth

In the first step, isothermal simulations are conducted at different inclinations, not taking into account the energy equation and instead using the isothermal relation $P(r,\theta) = c_s^2(r)\rho(r,\theta)$. According to Klahr [2024], GSF's onset condition for very short cooling times is $\kappa_z^2 \neq 0$. As shown in Equation (3.48), this is the case everywhere but the equator. Klahr [2024] also gives an analytic prescription of the GSF's linear growth rate for the fastest-growing mode in the limit of very short cooling times[4] as

$$\Gamma_{GSF}(\tau \ll \tau_c) = \frac{1}{2}\frac{|\kappa_z^2|}{\Omega} \tag{3.49}$$

With the rotational profile in Equation (3.40), this evaluates in code units to

$$\Gamma_{GSF}(\tau \ll \tau_c) = \frac{q^2 C}{8} r^{q/2-2} \sin^2(\theta) \tag{3.50}$$

$$\approx 0.14 \times \frac{\sin\theta}{\cos^2\theta} r^{q/2-1} \tag{3.51}$$

The resulting growth rate greatly varies through the simulation domain with radius and inclination. For all configurations, the fastest-growing mode's highest isothermal growth rate is at the box's upper inner end, and the lowest is at the simulation box's lower outer end. Table 3.1 shows the minimal and maximal isothermal growth rates for the used initial conditions. Also depicted are the growth rate at the middle of the box and the angle with respect to the equatorial plane at which the GSF modes are to be expected. This angle is calculated as $\xi_{GSF} = \tan^{-1}\left(\frac{k_R}{k_z}\right)$. The ratio of the cylindrical radial- to the vertical wavenumber of the fastest growing mode can be expressed by means of the epicyclic frequencies in Equations (3.45) and (3.48) as

[4] Much shorter than the critical limit τ_c obtained by Lin and Youdin [2015] for the onset of the VSI.

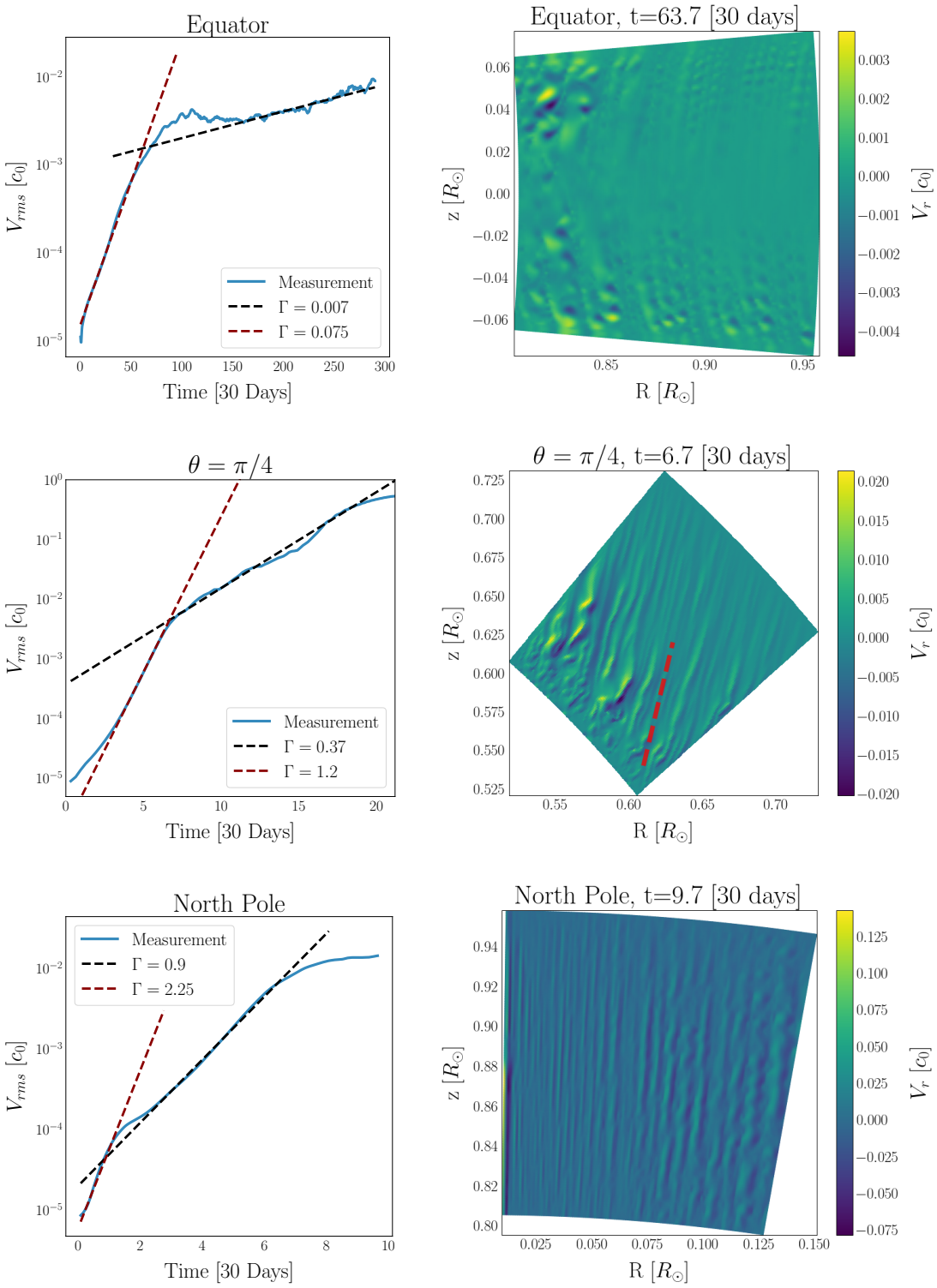


Figure 3.5: Isothermal growth rates (left) and spherically radial velocity component in the simulation domain (right) for runs at different inclinations. The red line in the middle-right plot shows a visual fit of the onset-angle ξ, which in this case, is found at $\xi^{run} \approx 78°$. The angles at the pole and the equator are close to 90°. The angle is taken with respect to the equator.

Table 3.1: Expected growth rates of the fastest growing modes for four simulation boxes at different inclinations relative to the equatorial plane. The growth rate at the box center and the maximal and minimal growth rates in the domain are given in units of 30 days. The two rightmost columns are the expected angles at which instability occurs. They are calculated using the expression found for GSF and VSI (without the factor 2 in Equation (3.52) and otherwise equal parameters), respectively.

Inclination	Γ_{mid}	Γ_{max}	Γ_{min}	ξ_{GSF} [°]	ξ_{VSI} [°]
$\pi/12$	0.17	0.28	0.1	72.7	58
$\pi/8$	0.28	0.42	0.18	70	54
$\pi/6$	0.375	0.65	0.28	70	53
$\pi/4$	0.85	1.35	0.6	73	59

Table 3.2: Measured growth rates and angles of the instability. Γ_1 is the initial growth rate and Γ_2 the secondary. The angle ξ_{run} is measured with respect to the equator.

Inclination	Γ_1	Γ_2	ξ_{run} [°]
Equator	0.075	0.007	~ 90
$\pi/12$	0.5	0.2	70
$\pi/8$	0.7	0.3	71
$\pi/6$	0.8	0.35	74
$\pi/4$	1.2	0.37	78
Pole	2.25	0.9	~ 90

$$\frac{k_R}{k_z} = 2\frac{\kappa_R^2}{\kappa_z^2} = 2\frac{[q\cos^2(\theta)+2]}{q\sin(\theta)\cos(\theta)}. \tag{3.52}$$

The expression is given in reciprocal form by Klahr [2024]. In Equation (3.52) it has been assumed that the value is not evaluated at the equator. The rightmost column in Table 3.1 is calculated without the factor of 2 in Equation (3.52) as is estimated for the VSI (Urpin and Brandenburg [1998]).
Figure 3.5 shows the growth rates in terms of the rms-velocity for the simulation runs at the equator, the north pole, and 45° inclination. Also shown are the typical flow patterns forming in terms of the spherically radial velocity component. For the $\theta = 45°$ run, the angle of the instability is estimated through the fit of a linear function with slope m. The corresponding angle is calculated as $\xi_{GSF}^{run} = \tan^{-1}(m)$. Equivalent figures for the remaining runs can be found in the Appendix Section B.2. The growth rates clearly grow steeper with inclination, and the growth rate close to the north pole is 1.5 orders of magnitude larger than at the equator. As outlined in Section 3.3.3, the simulation box close to the north pole is not stable under adiabatic

conditions. On the other hand, the measured growth rate in Figure 3.5 close to the pole, is steeper than in the adiabatic case (Figure 3.4). This indicates, that GSF and convection may be operating in parallel. The GSF pattern at the right-hand side of the simulation box close to the pole in Figure 3.5 and the barely visible radial velocity peak (likely due to convection) at the left-hand side of the same box is in line with this assumption.
In all runs, the growth phase can be visually separated into two phases, where the first shows steeper growth. This is also observed for VSI growth, where the initial growth corresponds to the linear growth phase of the instability and the second phase to a phase of non-linear growth. Inspection of the respective velocity maps, which are created directly after the linear growth phase (except for the equator and the north pole, which show more evolved patterns), reveals that the initial growth corresponds to the $r_{min} - \theta_{max}$ regions in the simulation boxes. This is consistent with the analytical results summarized in Table 3.1. The growth is, in all cases, close to the analytically described bounds for the respective simulation boxes. The measured angles fit their predicted values very well. The experimental angles are much closer to the GSF predictions than the VSI predictions. Table 3.2 summarizes the experimentally obtained growth rates and angles. Additional plots of the $\theta = \pi/8$ run during the linear growth phase can be seen in the appendix Figure B.6. A comparison of the linear- and non-linear growth phase morphology is given for the $\theta = \pi/12$ run in the appendix Figure B.7.
The takeaway message from this section is that the model as derived in Section 3.2 and initial conditions as in Section 3.3.2 supports the onset of GSF. The obtained growth rates and onset angles are close to analytically estimated values in the isothermal case. This validates the setup and proves the existence of the GSF.

3.4.2 Diffusive Thermalization

Generally, the stability of a stratified fluid under rotation, as described in Section 3.3.2, is determined by the Solberg-Høiland criteria (see Sections 1.1.4 and 3.3.3). The situation changes when thermal relaxation is considered. Klahr [2024] shows that in the case of long cooling times, the GSF's onset condition is equal to the isothermal case given by $\kappa_z^2 \neq 0$. This condition for the onset of instability is fulfilled for the initial conditions studied here, everywhere but the equatorial plane. At this plane, the convective overstability (COS, see Section 1.1.4) may be operating when thermal relaxation damps buoyancy in such a way that the amplitude of epicyclic oscillations is resonantly amplified.
According to Klahr [2024] the COS operates when either the GSF operates ($\kappa_z^4 \neq 0$), or there is buoyant amplification $N_z^2 + N_R^2 > 0$, which at the equator, where there is no vertical shear, reduces to $N_R^2 > 0$. Klahr [2024] gives the growth rates for the fastest-growing modes with long cooling times for the GSF (VSI, Urpin [2003]) as

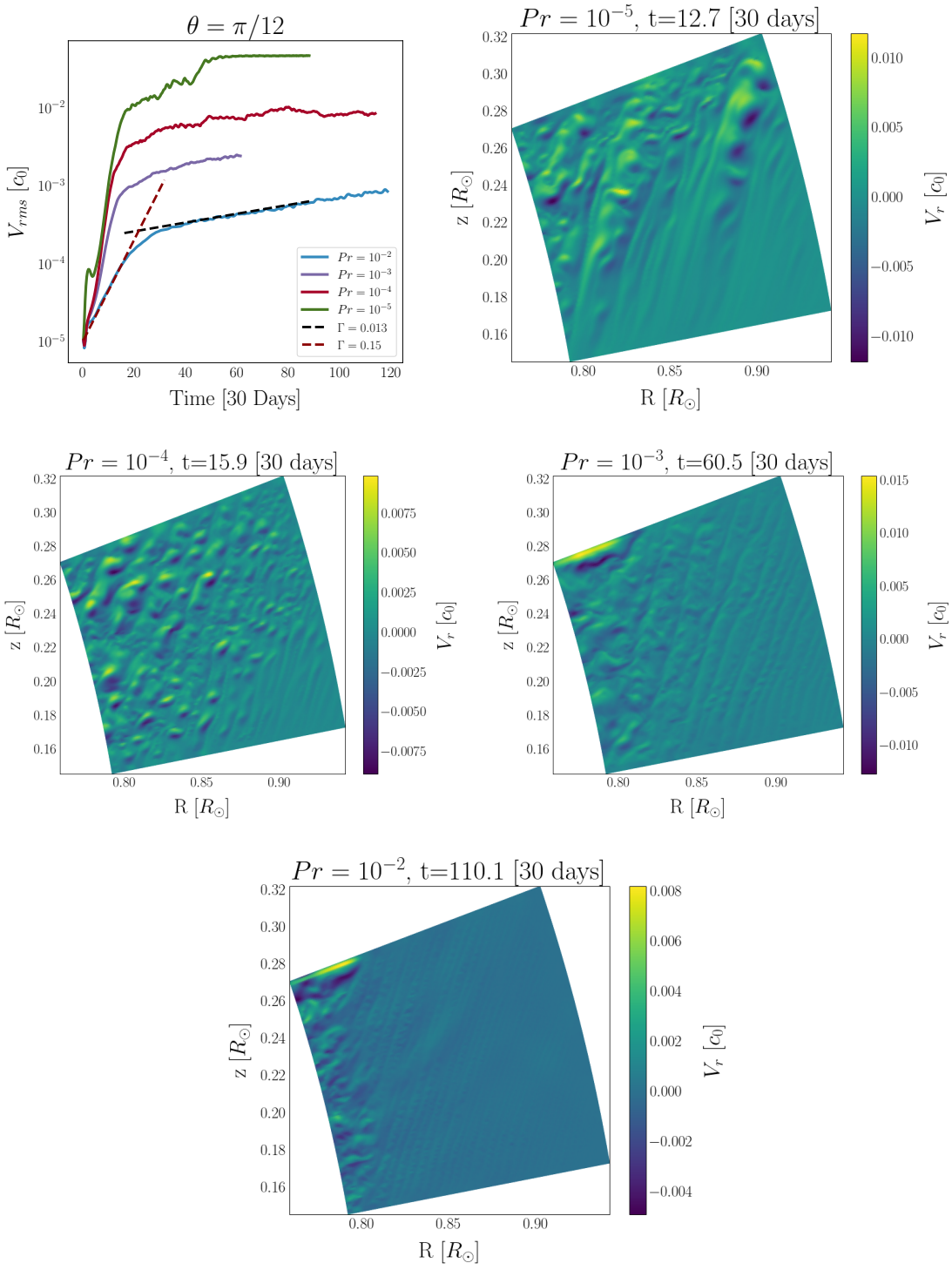


Figure 3.6: Instability growth rates in multiple simulations differing by the employed Prandtl number (top left) and map of the respective spherically-radial velocity components (remaining plots). The inclination in all runs is $\theta = \pi/12$.

$$\Gamma_{GSF}(\tau \gg \tau_c) \approx \frac{\kappa_z^4}{4\tau\gamma\Omega^2 N_z^2} \tag{3.53}$$

$$= \frac{\kappa_z^4}{4\tau^*\Omega N_z^2}, \tag{3.54}$$

and for the COS as

$$\Gamma_{COS}(\tau \gg \tau_c) = -\frac{1}{2}\frac{\tau^*}{1+\tau^{*2}}\frac{N_-^2}{\Omega} \tag{3.55}$$

$$= \frac{1}{8}\frac{\tau^*}{1+\tau^{*2}}\frac{\kappa_z^4}{N_z^2\Omega}. \tag{3.56}$$

$\tau^* = \tau\gamma\sqrt{\kappa_R^2}$ expresses the cooling time and τ_c is the critical cooling time close to the equatorial plane (for the VSI, Lin and Youdin [2015]). The ratio of the growth rates - which is not valid at the equator - reads

$$\frac{\Gamma_{GSF}}{\Gamma_{COS}} = 2 + 2/\tau^{*2}. \tag{3.57}$$

The GSF thus always dominates when active, but the COS becomes more important as τ^* approaches higher values. The COS also may become active close to the equatorial plane, where Equation (3.57) does not hold.
In this section, thermal relaxation, representing the optically thick regime is considered. Equation (3.32) gives the corresponding energy equation and the pressure is obtained through the caloric expression $P(r,\theta) = (\gamma-1)E$, where E is the thermal energy. The neglect of kinetic energy contributions is justified since the expected Mach number is lower than that in the isothermal runs. In this section, the initial conditions at an inclination of $\theta = \pi/12$ are evolved, where the isothermal flow saturates at $Ma \lesssim 0.5$. In all runs in this section, the thermal diffusivity χ is assumed to be a constant, within the respective simulation run, through the approximation $\nabla\cdot(\chi\rho\nabla T) \approx \chi\rho\nabla\cdot\nabla T$. This implies that no density variations exist with respect to the background in the domain. This is true to a first approximation as is evaluated posteriorly in the Appendix Figures B.8 and B.9, where the density fluctuations are determined to be below percent level.
In all simulation runs, the kinematic viscosity is assigned a small value $\nu = 10^{-9}$ and the thermal diffusivity is varied such that the Prandtl numbers $Pr = \nu/\chi \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. The kinematic viscosity, despite its small value, can be assumed not to be overshadowed by numerical diffusion since even slightly larger values for ν are observed to impede the onset of instability. The thermal relaxation times are of the order of the viscous (thermally diffusive) time scale $\tau_\chi = \frac{1}{\chi k^2}$, where k is the wavenumber of the instability in the direction of interest (Lin and Youdin

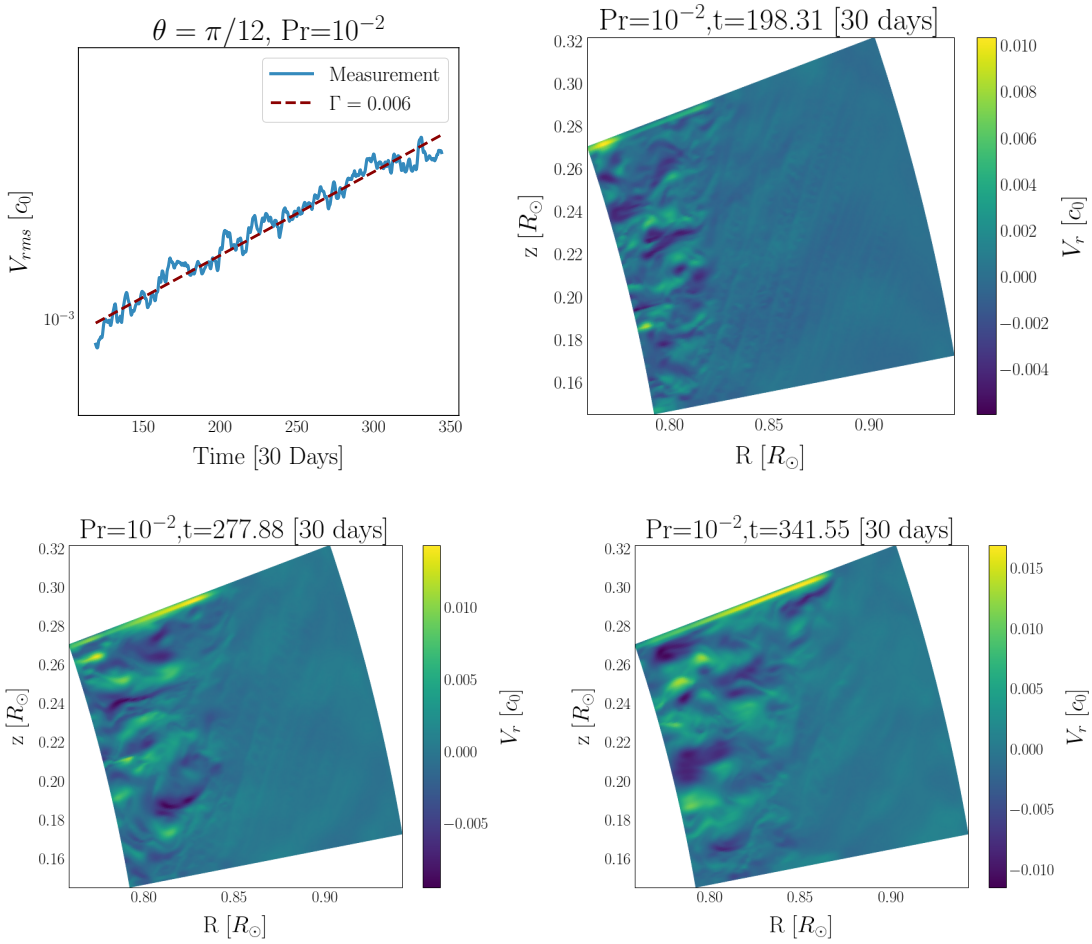
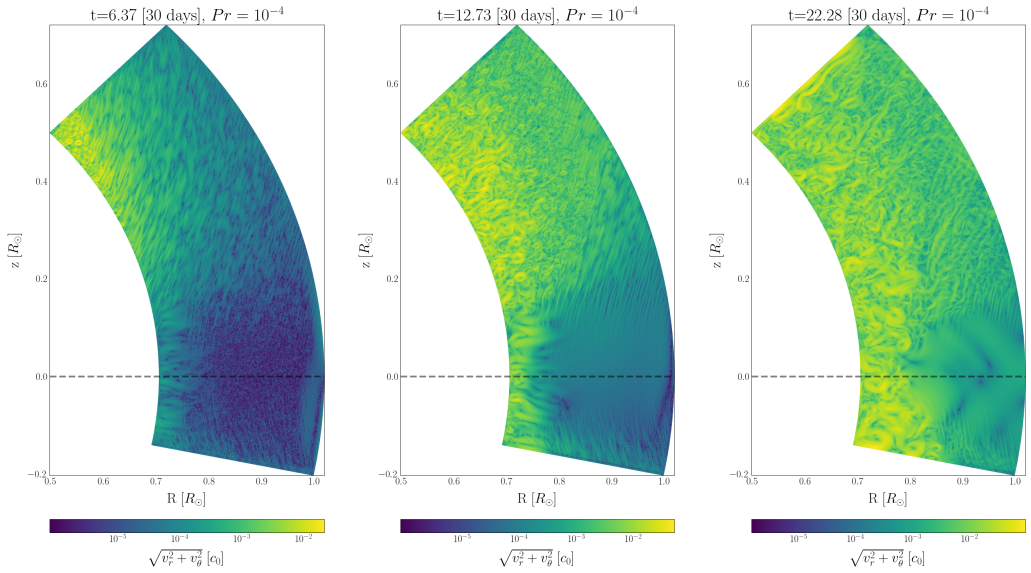


Figure 3.7: Long time evolution of $Pr = 10^{-2}$ at an inclination of $\theta = \pi/12$. The growth rate (top-left) and the radial velocity component are depicted at different times within the same simulation run.

[2015]). The growth rates and patterns in the radial velocity components are shown in Figure 3.6. The angle apparent in the flow patterns is the same as observed in the isothermal case for the respective inclination so one can be sure that it corresponds to the GSF.
Inspection of the growth rates confirms the prediction in Equation (3.54) that longer thermal relaxation times correspond to shallower growth. A secondary growth phase can be seen for the $Pr = 10^{-3}$ and $Pr = 10^{-2}$ runs, where the rate for the latter is approximately one order of magnitude weaker than the initial one. As mentioned, the first growth phase corresponds to GSF, which can be confirmed by the pattern of the flow and the angle at which it forms. Conversely, the secondary phase may be due to another effect. The fact that this secondary growth phase is not present for the lower Prandtl number cases, as the appearance of the pattern itself (compared to Klahr and Hubbard [2014]) hints that the secondary growth phase may be related to the COS.
This is further investigated in the $Pr = 10^{-2}$ case in Figure 3.7, where the corresponding simululation in Figure 3.6 is extended to more than 290 years. The measured growth rate is in the order of $\Gamma \approx 0.006$ and thus roughly a factor of two below the previous estimate for the secondary growth phase. To compare this to predictions, the thermal relaxation time has to be estimated. The pattern building up in the velocity maps in Figure 3.7 appears to be (almost) in the cylindrical radial direction. The relaxation time scale is thus $\tau \approx \frac{1}{\chi k_z^2}$. k_z can be estimated from the flow pattern with $k_z = \frac{2\pi}{\lambda_z}$, where $\lambda_z \approx L_z/m \approx 0.014$. $L_z \approx 0.14$ is the vertical extent of the simulated domain, and $m \approx 10$ is a guess for the number of modes within that extent. The estimate leads, for $Pr = 10^{-2}$ and consequently $\chi = 10^{-7}$, to a time scale $\tau \approx 100$. This is much larger than the critical time scale, which after Lin and Youdin [2015] is, in the case of a cylindrical radial temperature gradient q, obtained as $\tau_c = \frac{h|q|}{\gamma-1} \approx 3.75$. $h = c_s/(\Omega R)$ is the vertical scale height. Because of the cylindrical nature of this expression, it can be expected to match the considerations in the vicinity of the equatorial plane, in which, the inclination $\theta = \pi/12$ here is assumed to be.
Using the estimate for τ and the approximation for Equation (3.56) close to the midplane, or equatorial plane in this case, given by Klahr et al. [2023]

$$\Gamma_{COS} \approx \frac{h^2 q^2}{8} \frac{\gamma}{\gamma - 1} \frac{\tau^*}{1 + \tau^{*2}} \Omega, \tag{3.58}$$

and combining it with Equation (3.39) one obtains $\Gamma_{COS} \approx 0.0095$. Given the assumptions made to obtain this estimate, it is surprisingly close to the measured value in Figure 3.7. The value is only valid when the estimated number of oscillations in the simulated domain is not too far off, which, given the non-linear behavior that can be observed in that figure, is difficult to assess.



(a) $Pr = 10^{-4}$

Figure 3.8: Global simulation in the optically thick cooling regime mediated by diffusion. A Prandtl number of $Pr = 10^{-4}$ is used. Shown are three different time steps. The resolution is $200_r \times 512_\theta$. The initial time step is limited to 430 $\Delta t_{explicit}$. 2.7 radial pressure scale heights are covered, and self-gravity is neglected. The Goldreich-Schubert-Fricke instability is visible in different evolutionary stages and close to the equatorial plane, the typical morphology of the Convective Overstability is visible.

3.4.3 GSF in Large Simulation Box

As a further step in the analysis, simulations using a larger simulation box are conducted. The motivation to do so is that Caleo and Balbus [2016] describe an inverse relation between the vertical wave number k_z and the GSF growth rate (their Figure 2). Since this can be visually confirmed with Figure 3.6 and there is hence also a direct connection to the Prandtl number, a larger box size is employed to search for modes that are too large to fit in the smaller box of previous simulations. The radial extent of the chosen simulation box is $0.7R_\odot < r \leq 1R_\odot$ and the polar extent is $-0.2 \leq \theta \leq \pi/4$ (radians). Self-gravity is still neglected, and the strongest shear, both radial and vertical, is at the inner radial regions. The strongest shear present in this larger simulation box is thus stronger than in the previous simulation runs but follows the same description from Section 3.3.4. The spherical-radial extend is equivalent to 2.7 pressure scale heights and 200 cells are used in this direction. In the polar direction, 512 cells are used. Density and Energy are kept constant at the boundaries and as in reflective boundary conditions, the velocity is set to zero at the boundary. The polar asymmetry of the simulation box is not expected to

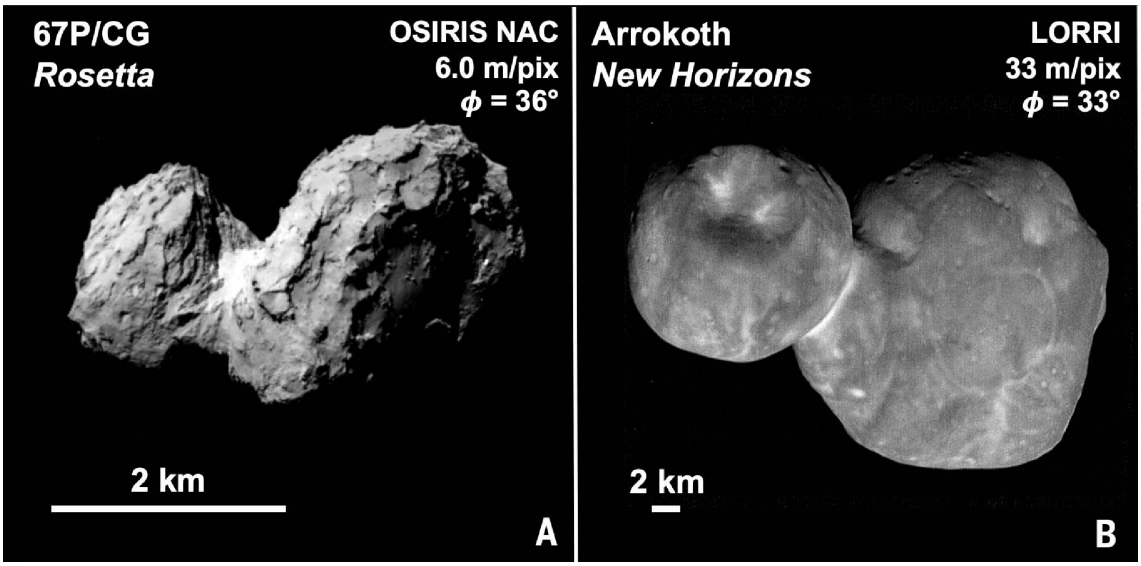
dramatically influence the simulation results; this is validated posteriorly.
The time step is chosen as before ($\Delta t_0 = 430 \Delta t_{explicit}$). The simulation is performed using a Prandtl number of $Pr = 10^{-4}$, where no indications of the COS have been observed in Section 3.4.2. The simulation is carried out up to a time when the initial linear growth phase is expected to be completed (upper-left in Figure 3.6), corresponding to $t_{end} \approx 600$ days.
Figure 3.8 shows the local norm of radial and polar velocity components in units of the speed of sound for both simulation runs at three different times. At later times, the flow $0.7R_\odot < r < 0.8R_\odot$ appears to become rather turbulent, but at larger radii, large-scale filaments form. The COS typical morphology is distinctly visible in the inner regions close to the equatorial plane at the two earlier times. Unfortunately, no claim about the presence of the COS in the depicted simulation can be made with certainty at this point. The reason is, that no validation of the stability of the initial conditions at $0.7R_\odot < r < 0.8R_\odot$ against convection has been conducted and the presence of the "COS" morphology, as well as the turbulence in the inner radial regions may be accessioned with convection. Analysis of this will be the subject of future work.

3.5 Conclusion

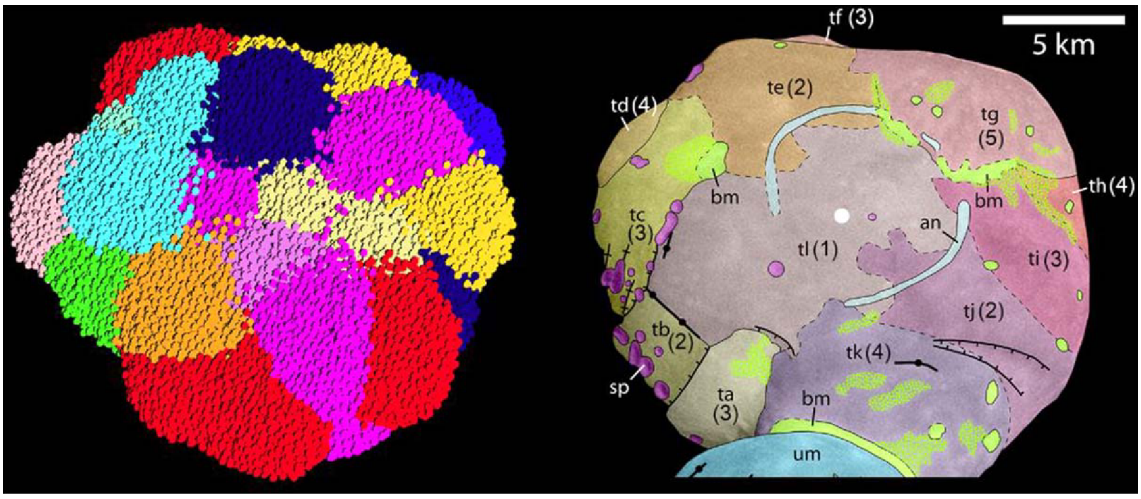
This chapter presents the very first global numerical simulation of the GSF in the spherical geometry of a pressure-supported environment. To do so, a simplified solar model expressing the equilibrium of gravitational-, centrifugal-, and pressure forces is derived under the assumption of negligible self-gravity. The derivation of the model is built on a spherically radial polytropic parametrization of density and temperature, which is closely analogous to studies of the VSI. The rotational contribution to the equilibrium of forces is controlled with a free parameter. Initial conditions with rotation are set up, which is stronger than in stellar interiors but much weaker than in PPDs.
The isothermal evolution of these initial conditions at different inclinations produces modes of the GSF. The results from analytical considerations are confirmed by the hydrodynamic simulations in this chapter. The evolution of the same initial conditions at relatively low inclinations ($\sim 20^\circ$) under optically thick thermalization produces GSF modes. In small simulation boxes, additional modes are observed for high Prandtl numbers $Pr > 10^{-3}$, that are likely linked to the COS. In a larger simulation box at $Pr = 10^{-4}$, COS and GSF modes are observed to exist in different regions of the simulation domain. Modeling of a small box at $\theta = \pi/12$ inclination with a Prandtl number 10^{-2} shows a situation where the initial growth of GSF modes is overtaken, likely by the COS.

4 Planetesimal Formation in Pebble Cloud Collapse Simulations

This chapter aims to contribute to the understanding of the processes that lead to the formation of planetesimals and, finally, planets. It compares the outcome of simulated core collapses for two distances from a solar mass star. The first distance is 2.3 AU, and the second is 25 AU, corresponding to the solar system's asteroid- and Kuiper belt positions. The influence of gas on pebbles at collapse, described by Klahr and Schreiber [2020] and Klahr and Schreiber [2021], is compared to the case of no gas-to-pebble interaction as also studied in Polak and Klahr [2023]. Recent observations of the Kuiper belt objects Arrokoth and 67P/CG (Stern et al. [2019], Stern et al. [2023]), as well as numerical modeling of the asteroid Bennu (Zhang et al. [2022]), have revealed an astonishing level of sub-structure in the respective objects. This is shown in the example of Arrokoth and its larger lobe Wenu in Figure 4.1. Because of the observed substructure, this chapter also analyzes the formation and growth history of the simulated planetesimals to search for an indication of possible substructure in these objects.
The contents of this chapter are taken from Meyer et al. [in prep.] and Meyer and Klahr [in prep.]. The hereby presented contribution to these works is the analysis of the finished simulation runs. The first part, Section 4.1, outlines the numerical methodology relevant to the chapter. Section 4.2 summarizes the results on the obtained size and mass distributions and compares these results to observational data by Delbo' et al. [2017], as well as to previous numerical studies by Polak and Klahr [2023]. It also analyzes the occurrence of parent-satellite- and binary systems in the simulations. This is of interest to find hints for the frequency of occurrence of contact binaries, such as Arrokoth. Section 4.3, shows the analysis of the occurrence and mass scale of mergers as well as their impact on the final planetesimal mass. Noteworthy examples of individual objects that are born through a combination of pebble- and boulder accretion are given.



(a) Rosetta image of 67P/Churyumov–Gerasimenko (left, Sierks et al. [2015]) and Arrokoth (right, Spencer et al. [2020]).



(b) Substructure of Arrokoth's lobe Wenu as modeled and inferred by Stern et al. [2023].

Figure 4.1: Top: Image of the two contact binaries 67P/CG (left) and Arrokoth (right). Image is taken from Spencer et al. [2020]. Bottom: Comparison of the outcome of modeling Arrokoth's lobe Wenu as a compound object (left) to observation (right). The geologic map of Wenu in the observed image is highlighted by color. The image is taken from Stern et al. [2023].

4.1 Methodology

This section introduces the numerical methods relevant to the chapter. This involves the solved equations, initial conditions, the utilized object identification algorithm, and the description of the merger tree algorithm implemented within this work to analyze the possible sub-structure of the identified planetesimals.

4.1.1 Simulation Setup

The simulations in the context of this chapter are conducted using the Meshless Finite Mass (MFM) code GIZMO (Hopkins [2015]). As a derivation of the Smoothed Particle Hydrodynamics (SPH) approach, MFM traces individual simulation particles. In contrast to SPH, mass is conserved instead of density in MFM. This advances its angular momentum conservation properties. The simulations are conducted using a shearing box model in a Cartesian coordinate system, representing a small excerpt of a PPD at a distance $D_\star$ from the central stellar object of mass $M_\star$. Self-gravity, as driving force, is included.
GIZMO solves the full set of compressible Euler equations in a moving frame (velocity $\vec{v}_{\text{frame}}$)

$$\begin{aligned} \frac{\partial \vec{U}}{\partial t} + \nabla \cdot \left(\vec{F} - \vec{v}_{\text{frame}} \otimes \vec{U} \right) &= \mathcal{S}, \\ \vec{U} = \begin{pmatrix} \rho \\ \rho\vec{v} \\ \rho e \end{pmatrix}, \vec{F} &= \begin{pmatrix} \rho\vec{v} \\ \rho\vec{v} \otimes \vec{v} + P\mathcal{I} \\ (\rho e + P)\vec{v} \end{pmatrix}. \end{aligned} \tag{4.1}$$

$\mathcal{S}$ indicates source functions, including gravity terms. $\mathcal{I}$ is the unit tensor and ρ, $\vec{v}$ and P are the density, velocity and pressure. The pressure P has to be obtained through an equation of state (EOS), which solely and completely describes a particle's state as being solid or gaseous. For the simulation runs including the effect of gas on pebble motion (in the following termed diffusive runs), an EOS is constructed, such that for each particle, the pressure is taken as a combination of fluid pressure P_{fluid} and solid pressure $\tilde{P}_{\text{solid}}$

$$P = \sqrt{P_{\text{fluid}}^2 + \tilde{P}_{\text{solid}}^2}. \tag{4.2}$$

With this definition of pressure, all particles can change their phase between solid and fluid through a smooth transition. For the simulation runs not considering gas-to-pebble interaction (in the following non-diffusive runs), the fluid phase is modeled similarly to Polak and Klahr [2023], and the transition to the solid phase takes place through a switch similar to Truelove et al. [1998]. The fluid phase contains - at least in general - gas, dust, and pebbles. In the simulation runs, gas and dust are

not modeled independently, and the fluid phase is modeled as purely consisting of pebbles.
The solid pressure is calculated using the Tillotson equation of state. Tillotson [1962] derived a general expression describing the response of metal to high-velocity impacts in the regimes of compression (regime I), heating (II), vaporization (III), and cooling (IV). Despite its initial application to metals, it has been successfully used for other materials (see, e.g., Brundage [2013]), including giant impact simulations (Deng et al. [2019]). Since the energy of particles is not expected to undergo significant changes, only regimes I and II, where the pressure is expressed as

$$P_{\mathrm{I,II}} = \left[a + \frac{b}{\frac{E}{E_0 \eta^2} + 1} \right] E\rho + A\mu + B\mu^2, \tag{4.3}$$

are considered. Here, E is the specific internal energy, $\eta = \frac{\rho}{\rho_s}$ with anticipated solid-density ρ_s and $\mu = \eta - 1$. a and b are the Grüneisen-coefficients (Grüneisen [1912]), describing temperature variation with volume changes at constant entropy, which here, is assumed zero hence also, a and b are zero. Subsequently, Equation (4.3) reduces to

$$P_{\mathrm{solid}} = A \left[\frac{\rho}{\rho_s} - 1 \right] + B \left[\frac{\rho}{\rho_s} - 1 \right]^2, \tag{4.4}$$

leaving the choice for A - the bulk modulus - and B - the structural integrity factor. Since modeling solids similar to cometary material (i.e., porous icy aggregates) is anticipated, relatively high compressibility and deformability with $A = B = 10^7$ Ba compared to water-ice (for which $A = B = 10^{11}$ Ba) are assumed. At the same time, the choice of A and B allows the material to preserve its structural integrity. To prevent unphysical behavior through negative pressure obtained from the Tillotson EOS, Equation (4.2) is dependent on $\tilde{P}_{\mathrm{solid}} = max(0, P_{\mathrm{solid}})$.
The fluid - or pebble - pressure is modeled in the diffusive simulation runs as $P_{\mathrm{fluid}} = a^2_{\mathrm{pebble}} \rho$, where a_{pebble} is the effective sound speed of the pebbles, which, according to Klahr and Schreiber [2021] is given as a fraction of the isothermal gas sound speed c_s as

$$a_{\mathrm{pebble}} = \sqrt{\frac{\delta}{\mathrm{St} + \delta}} c_s \approx \sqrt{\frac{\delta}{\mathrm{St}}} c_s. \tag{4.5}$$

Here, St is the Stokes number of the pebbles and $\delta = \frac{D}{H c_s}$ the dimensionless diffusivity. H is the (gas) pressure scale height, and D is the diffusivity. Using the dust to gas ratio $\epsilon_{\mathrm{Hill}} = \frac{\rho_{\mathrm{Hill}}}{\rho}$ at Hill density

$$\rho_{\text{Hill}} = \frac{9}{4\pi}\frac{M_\star}{D_\star^3}, \tag{4.6}$$

the dimensionless diffusivity is given by Klahr and Schreiber [2020] for $10 < \epsilon_{\text{Hill}} < 100$ in linear relation to the Stokes number as

$$\delta = \delta_0 \frac{10}{\epsilon_{\text{Hill}}}\frac{\text{St}}{0.1}, \tag{4.7}$$

with $\delta_0 = 2.7 \times 10^{-6}$ at St=0.1 and $\epsilon_{\text{Hill}} = 10$ (Schreiber and Klahr [2018]). The system of Equations (4.1) is thus closed by fixing St, ϵ_{Hill} and ρ_s. In addition, to avoid spurious fragmentation, and to resolve the local Jeans mass, the kernel width h has to be smaller than the Jeans length

$$h < \lambda_{\text{J}} = a_{\text{J}}\sqrt{\frac{\pi}{G\rho}}. \tag{4.8}$$

With the safety factor $2h = \lambda_{\text{J}}$, a lower limit for the pebble sound speed can be found as

$$a_{\text{J}} > 2h\sqrt{\frac{G\rho}{\pi}}. \tag{4.9}$$

From this, a stabilizing pressure is implemented as $P_{\text{fluid}} = max(a^2_{\text{pebble}}, a^2_{\text{J}})\rho$. This is implemented as the fluid phase pressure for every simulation particle

4.1.2 Initial Conditions

For all simulation runs, a solar mass central object $M_\star = M_\odot$ is assumed and the distance to it is fixed at 2.3 AU and 25 AU, respectively. A spherical cloud of uniformly distributed simulation particles with constant density inside is set up. As shown by Klahr and Schreiber [2021], the characteristic radius of a Bonner-Ebert solution, at which such a cloud can not anymore withstand diffusion and allows for contraction, is given by

$$l_{\text{cloud}} = \frac{1}{3}\sqrt{\frac{\delta}{\text{St}}}H. \tag{4.10}$$

This gives not only the radius of the initial cloud but also the cloud mass, when the density is fixed at Hill density

Table 4.1: Summary of conducted and analyzed simulation runs. The simulation runs at a distance of 2.3 AU from the central star are termed `Asteroid` runs and those at 25 AU `Kuiper` runs. ρ_s is the density above which a simulation particle is defined as solid, and N is the number of simulation particles.

Runs	Sub-Runs	$D_\star$ [AU]	ρ_s $[g/cm^3]$	Relaxed	Diffusive	N $[10^5]$
Asteroid10N22	A-L	2.3	2	No	No	0.1
Asteroid10RN22	A-L	2.3	2	Yes	No	0.1
AsteroidN22	A-M	2.3	2	No	No	1
AsteroidN22R	A-M	2.3	2	Yes	No	1
AsteroidN82	A-M	2.3	2	No	Yes	1
AsteroidRN82	A-M	2.3	2	Yes	Yes	1
Kuiper25N22	A-K	25	0.5	No	No	1
Kuiper25N82	A-K	25	0.5	No	Yes	1
Kuiper25RN22	A-K	25	0.5	Yes	No	1

$$m_{\text{cloud}} = \frac{4\pi}{3} l_c^3 \rho_{\text{Hill}}. \tag{4.11}$$

A safety factor to the density is applied, such that the critical length scale is reduced to a factor $\frac{2}{3}$ of its initial value, resulting in

$$\rho^*_{\text{Hill}} = \frac{1}{\pi}\left(\frac{3}{2}\right)^5 \frac{M_\star}{D_\star^3}. \tag{4.12}$$

This ensures the gravitational collapse under the influence of shear forces from the central star as discussed in Polak and Klahr [2023] and Paczynski [1971].
For all simulation runs, the total mass in the simulation box is fixed to the equivalent of a 100 km-sized object of chosen density. The solid density is $\rho_s = \rho_s(D_\star)$. In accordance with Klahr and Schreiber [2020], the density at a distance to the star of 2.3 AU is $\rho_s = 2.0\frac{g}{cm^3}$ and for a distance of 25 AU, the solid density $\rho_s = 0.5\frac{g}{cm^3}$ is chosen. This fixes the cloud masses to $M_{\text{Asteroid}} \approx 1.05 \cdot 10^{21}$ grams and $M_{\text{Kuiper}} \approx 2.62 \cdot 10^{20}$ grams, at the respective distances. By the nature of the GIZMO code being MFM, each simulation particle is assigned a mass at the start, so the initial volume for each particle must be estimated at the initial condition setup. The density is then re-calculated by the code in dependence on the particle mass and the inter-particle distance.
Simulations with and without pebble-to-gas coupling are set up for both distances to the star. In the first case the dimensionless diffusivity δ (Equation(4.5)) is obtained through Equation (4.7) and the consequent sound speed of the pebble fluid through Equation (4.5). In the latter case, δ, and consequently the pebble sound speed,

is set to zero, resulting in vanishing pressure P_{fluid}. For each of these simulation sets, two sets of runs are conducted. The first set has a random simulation particle distribution inside the desired cloud (unrelaxed in the following). The second set has a relaxed initial stage, where a random simulation particle distribution is chosen as the starting point, and temporal integration is done with deactivated self-gravity for a few time steps to smoothen the initial distribution against larger inhomogeneities in the density distribution.
In addition to these setups, lower resolution runs for the 2.3 AU case are conducted for the non-diffuse case with both relaxed and unrelaxed initial conditions using 10^4 simulation particles. All other runs are set up with 10^5 simulation particles. A complete list of all simulation runs in this work with their distinguishing properties is given in Table 4.1. Multiple sub-runs are conducted for each of the hereby described runs, differing only by the inherent random component in the setup. This is done to increase the reliability of the obtained results.
The final simulation time is given in units of the free-fall time t_{ff} defined as

$$\begin{aligned} t_{ff} &= \sqrt{\frac{3\pi}{32G}} \frac{1}{\sqrt{\rho^*_{\text{Hill}}}} \\ &= \frac{\pi}{9} \sqrt{\frac{D_\star^3}{GM_\star}}. \end{aligned} \tag{4.13}$$

For $M_\star = M_\odot$ one obtains $t_{ff}(2.3 \text{ AU}) \approx 6.12 \times 10^6$ seconds and $t_{ff}(25 \text{ AU}) \approx 2.19 \times 10^8$ seconds.

4.1.3 Object and Binary Identification

For every examined time step, objects are identified with the "Density-Based Spatial Clustering of Applications with Noise" algorithm (DBSCAN) (Ester et al. [1996]). Its implementation in the `scikit-learn` (Pedregosa et al. [2011]) package is used. The two free parameters of DBSCAN that must be manually defined are the minimal particle count and the maximal distance between neighboring particles. DBSCAN gives a set of clustered particles and a set of unidentifiable particles. The particles considered for clustering are only the simulation particles with a density higher than the solid density for the respective distance to the central star. Because of this, each identified object corresponds to a solid object or planetesimal. In the following, the terms *planetesimal* and *object* are used interchangeably. The parameters for object detection are similar to Polak and Klahr [2023]. The underlying assumptions for their definition are;

1. A particle is solid once its density exceeds ρ_s. Only solid (simulation-) particles can form planetesimals and are considered for clustering.

2. Each planetesimal contains at least $N_{min} = 10$ particles. The corresponding

minimal equivalence diameter is $\sim$ 4.64 km. This is further motivated in Section 4.2.2.

3. At least N_{min} particles have to be located in a sphere with radius $\epsilon_{min} \sim \left(\frac{0.1‰ M_{\text{Cloud}}}{\rho_s}\right)^{1/3}$ for an object to be detected.

4. Simulation particles fulfilling condition 1 but not, concurrently, 2 and 3 are labeled unidentified and belong to no object. They are thus identified as noise.

The count of unidentified particles can be used to measure the validity of conditions 2 and 3. This is done in Section 4.2.2, which analyses the count of unidentified solid simulation particles and the count of identified objects as a function of different values of N_{min}. After identification, each object is assigned a UUID. This allows the distinguishability of the identified objects.
Binaries and parent-satellite systems are identified in the simulations similar to Polak and Klahr [2023] or Nesvorný et al. [2021], where for each object i a loop over all other objects $j \neq i$ in the same run is performed and the binding energy

$$\begin{aligned} E^{ij}_{\text{binding}} &= E^{ij}_{\text{grav}} + E^{ij}_{\text{kin,rel}}, \\ E^{ij}_{\text{grav}} &= \frac{GM_iM_j}{|\vec{r}_i - \vec{r}_j|}, \\ E^{ij}_{\text{kin,rel}} &= -\frac{1}{2}M_i|\vec{v}_i - \vec{v}_j|^2, \end{aligned} \tag{4.14}$$

is evaluated. For a positive binding energy $E^{ij}_{\text{binding}} > 0$, object i is bound to object j and a parent-satellite system is identified. If, in addition, also $E^{ji}_{\text{binding}} > 0$, objects i and j form a binary system.

4.1.4 Merger Tree Algorithm

This section introduces the merger tree algorithm implemented in the context of this chapter. When comparing the objects in two successive time steps, three different situations can occur regarding the identified object count. The first case is the case of equal object count. Here, the algorithm assumes that no new object has been born and no merging event has occurred. Consequently, a one-to-one mapping is constructed between the object UUIDs obtained from the last time step and those from the current time step. In the second case, the object count for the later time step is larger than in the previous one. In this case, a new object is born and assigned a new UUID. The algorithm here consists of two steps. First, the *new* object must be identified, and second, the remaining objects must be aligned similarly to the equal-object case. Third is the case where the former time-step snapshot contains more objects than the newer one. This is the case where a merging event has taken place, and the tasks are to identify the objects involved in the merger and conduct

the same one-to-one mapping for the remaining objects as in the other cases.
The algorithm is based on the assumption of maximum intersection between the two sets of objects. This holds for the one-to-one mapping and the case of new objects and a merging event. The implications of this assumption are outlined in some detail later in this section. When labeling the set of objects obtained from one of the snapshots as $S = \{s_1, s_2, ..., s_J\}$ and the other as $L = \{l_1, l_2, ..., l_K\}$ one can paraphrase the one-to-one identification algorithm under the condition that both sets contain an equal count of objects ($J = K$) as follows:

1. Calculate for an object l_j all coefficients $\psi_{l_j s_k}$[1], and find the corresponding objects $s_{\bar{k}}$, where ψ is minimal.

2. Likewise, calculate all $\psi_{s_k l_j}$ for every s_k and find $l_{\bar{j}}$ as above.

3. Find all objects that can be unambiguously assigned (equal to the condition $(s_j, l_{\bar{k}}) = (s_{\bar{j}}, l_k)$) and mark them as identical. Overwrite the UUIDs of the found objects in the later time step with the corresponding ones of the former time step.

4. Remove the, in step 3, unambiguously mapped l_k and s_k from L and S, respectively, and repeat at step 1 until all objects are identified.

In the case of a new object or a merging event, the objects found in one set of objects but not the other has to be identified first. In the case of a merging event, step 1 searches for the minimal value of ψ, and in the case of the birth of a new object, step 1 searches for the maximal value of ψ. At this point, it is important to mention that the identification algorithm is constructed to be agnostic of time. It only operates on two sets of objects given by the objects in two consecutive time steps. In the case of birth or merging, these sets have different sizes. So, step 1 can only be used when it is made sure that S is the smaller set and L is the larger. In the case of a birth event, S is obtained from the former time step and L from the latter, and vice-versa for merging events. With the above routine, the case of a birth event and the case of constant object count are fully described.
In the case of a merging event, step 2 is used to find the object that is most unlikely to have a direct predecessor. This object is defined as the merger *parent*, i.e., the object preserved in a merging event. This object must have gained mass. The two merging components are identified by removing the merger parent from S and proceeding with step 1., again maximizing ψ instead of minimizing it. The two objects in L with the highest corresponding ψ values are identified as merger components. Finally, the heavier of these two objects is defined as the surviving one, and its UUID is used to overwrite the merger parent's one. The lighter of the objects is defined as merged away and does not appear in future time steps anymore.

[1] ψ_{xy} is the coefficient obtained from a kernel function ($\Psi(x, y)$) giving the probability that object y is found inside of object x. Because of this, it can likewise be used to determine vanishing objects in the case of merging events and identifying new objects. It is not necessarily symmetric, so generally $\psi_{xy} \neq \psi_{yx}$.

Merger-Tree Kernel Function

As outlined in the previous section, a kernel function is utilized. It defines the probability that one object is found as a component of another. To find such a function, one can use the fact that each object is composed of multiple (at least 10) simulation particles for which unique IDs are known in addition to their positions, masses, velocities, etc. To compare two objects, both are analyzed as super-particle sets and the one containing more particles is named M^+ and the other M^-, respectively. This, again, is completely independent of the time step in which the respective objects are found. Note that the Kernel function operations on two different planetesimals, which are here called sets of particles (the simulation particles), as opposed to the routine in the previous section, which operates on sets of planetesimals, where the individual simulation particles are not considered. The somewhat arbitrarily chosen, but normalized function

$$\begin{aligned} \Psi(M^+, M^-) &= \frac{1}{\omega_{\text{mass}} + \omega_{\text{ID}}} \left[\omega_{\text{mass}} P_M + \omega_{\text{ID}} P_{\text{ID}}\right], \\ P_M &= \frac{\sum M^+_{\text{mass}} - \sum M^-_{\text{mass}}}{M_{\text{Cloud}}}, \\ P_{\text{ID}} &= \frac{|M^-_{\text{ID}} \cap M^+_{\text{ID}}|}{|M^-_{\text{ID}}|}, \end{aligned} \tag{4.15}$$

is used. Here, ω_{mass} and ω_{ID} are weights chosen under the condition $\Sigma_i \omega_i = 1$. P_M is the weighted cluster-mass difference with respect to the cloud mass, and P_{ID} is the count of particle IDs from the smaller set that can be found in the larger set. Both components are normalized. While P_M is symmetric under exchange of M^+ and M^-, P_{ID} is not. Because of this, $\Psi(M^+, M^-)$ can be interpreted as the probability of finding object M^- inside of M^+. For the rest of this work $\omega_{\text{mass}} = \omega_{\text{ID}} = \frac{1}{2}$. Despite its simplicity, Equation (4.15) has proved to be a very useful choice for the merger-tree identification, and more complicated choices of Ψ, taking into account object- and particle positions, have not given better results. This is because the limitations of the merger-tree algorithm are due to the underlying assumptions made in Section 4.1.4 and not by the object association step.

Merger-Tree Limitations

Despite the somewhat obvious fact that all limitations of object identification through `DBSCAN` are directly inherited to the merger-tree algorithm, two main limitations are introduced through its construction. The first comes from the assumption that when the object count is constant between two successive time steps, neither a merging nor a birth event occurs. Especially in the early evolutionary stages, corresponding to $t \approx 1 - 1.1 t_{ff}$, the occurrence of two or more temporally coinciding events that preserve the count of planetesimals can not be ruled out. This is due to the high amount of activity happening on small scales at early times as Section 4.3.1 analy-

ses. However, this chapter's analysis does not depend on the exact tracking of very small objects. The temporal evolution of objects is important only on larger mass scales, where the merger-tree algorithm has proven to be very reliable.
The second limitation is more problematic. It is the fact that collision events - being disruptive or not - can not be identified as such. These events are falsely identified as merging events, followed by the later birth of one or more relatively heavy objects[2]. These situations happen in only a few (~ 10) merger trees, where the defining criterion is that the percentage of boulder accreted mass is $\geq 100\%$ of the final object mass. Compared to the ~ 450 trees produced, this is negligible and the respective objects are excluded from the analysis. The loss of mass after a merging event can, in principle, also be due to very heavy rotation of an object. This case is equally neglected and no distinction to a disruptive event is made. Disruptive events are, in principle, of interest as well. However, their analysis is more meaningful when tensile strengths are included in the modeling of objects, which will be the subject of future work.

[2]Objects are typically first identified - or born - with a mass in the order of the minimum mass allowed in `DBSCAN`. This is not necessarily the case for objects formed after a collision event.

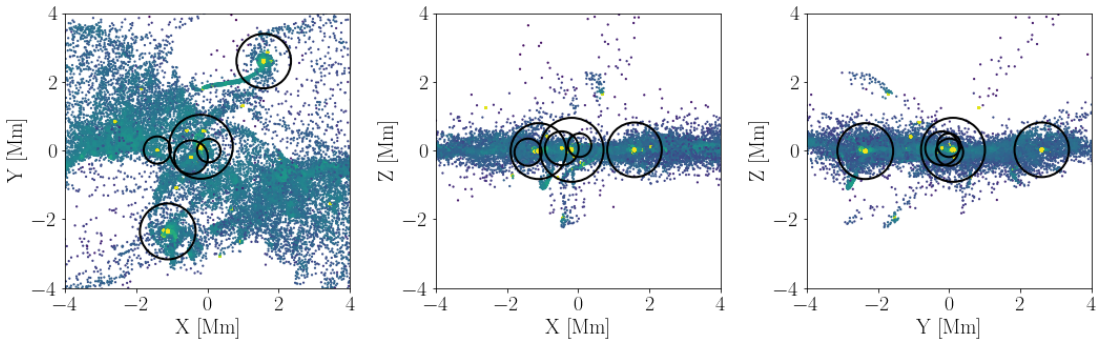


Figure 4.2: Projection scatter plots of simulation particles at the onset of fragmentation with coordinates given in mega-meters. Color represents density. The circles mark the positions of the six most massive objects and the circle radius corresponds to object mass. The data corresponds to the `AsteroidN22A` run at 1.15 t_{ff}. For an explanation of the simulation runs, see Table 4.1.

4.2 Initial Mass Function of Planetesimals

The analysis of simulation runs in this section focuses on the products of cloud collapse. The starting point is the evolution of the solid mass fraction in the simulation runs over time. This is done in Section 4.2.1. Section 4.2.2 validates the choice of parameter values used for the clustering algorithm `DBSCAN` as described in Section 4.1.3. Section 4.2.3 analyses the influence of diffusivity and distance to the central star by means of average planetesimal size per simulation run. Section 4.2.4 outlines the distribution of object masses and, hence, the initial mass function of planetesimals. Section 4.2.5 compares the findings for the `Asteroid` runs to observational data and uses this to calibrate the size scale for all runs. Section 4.2.6 analyzes the frequency of occurrence of parent-satellite- and binary systems through the simulation runs.
Exemplary, a projection plot of the simulation particles and identified objects in Figure 4.2 shows six similar-sized objects alongside multiple smaller objects that are located close to the coordinate origin. It gives an impression of the simulated spatial scales and the count of simulation particles and planetesimals.

4.2.1 Solid Mass Evolution

The solid mass fraction over time for the $N = 10^5$ runs is analyzed. Figure 4.3 shows the averages of total solid mass against simulation time for all conducted runs. This likewise includes particles that are solid but not assigned to objects and those that are assigned to an object as well. It becomes apparent that the solid mass evolution for both distances to the central star is very similar and can be roughly split into three regimes.

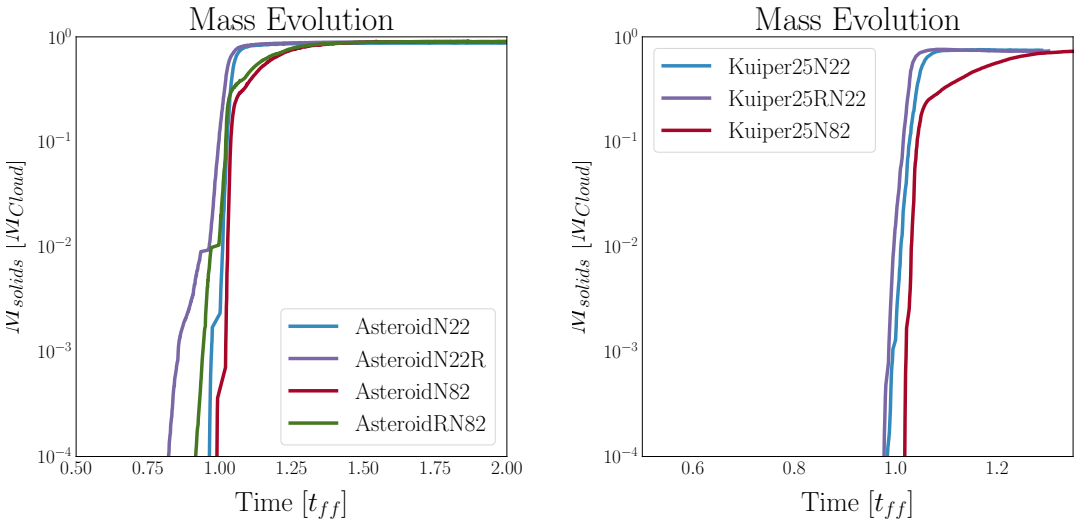


Figure 4.3: Mass contained in solid objects over time in terms of total cloud mass and free-fall time for the $D_\star = 2.3$ AU (left) and $D_\star = 25$ AU (right) runs.

The first is the pre-collapse phase, where no mass is present in the solid state. Although it is due to the stochastical distribution of simulation particles not guaranteed that at all positions, the initial density is below ρ_s, no solid mass at early times is observed. This holds for the `Asteroid` and the `Kuiper` runs. The second phase is the phase of collapse, where solid density is exceeded for most simulation particles, and the phase transition fluid-solid occurs between 0.95 and 1.25 t_{ff}. A clear distinction can be drawn between the diffusive and non-diffusive runs. The non-diffusive runs show rapid growth directly to the final solid mass fraction. The diffusive runs, on the other hand, exhibit a secondary growth phase with decelerated growth between $1 - 1.3\ t_{ff}$.
In the third regime, the solid mass fraction is approximately constant and does not change much. This implies that after this stage is reached, no substantial object growth is to be expected when merging events are neglected. In this section, the analysis is done at simulation times of $\sim 2\ t_{ff}$ for the `Asteroid` and $\sim 1.2\ t_{ff}$ for the `Kuiper` runs. The final fraction of solids for the `Asteroid` runs is $0.87 - 0.90$ M_{Cloud} and $0.7 - 0.75$ M_{Cloud} for the `Kuiper` runs. This difference in solid formation efficiency is already the first result of this chapter. This section concludes with the statements that the distance to the star significantly impacts the final fraction of solid- to fluid-state material, the timescale of solid formation is not influenced by distance to the star when expressed in terms of the free-fall time and the diffusivity influences the formation history of solid material but not the final solid mass fraction.

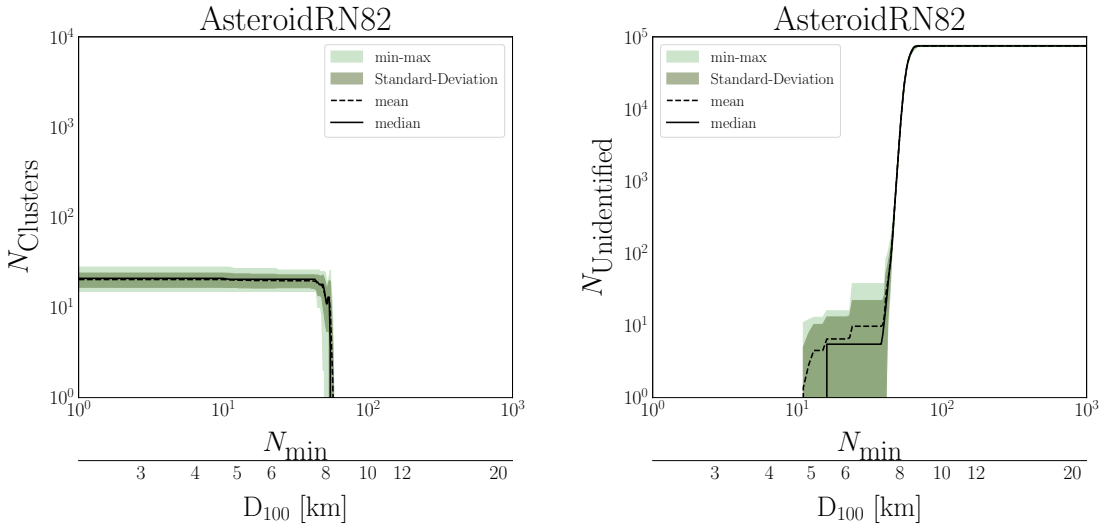


Figure 4.4: The count of identified objects (here clusters) over the respective value of N_{min} used in the identification step (left) and unassigned but solid particles over minimal object size (right). N_{min} is actually $N_{min} + 1$ to exclude the single particle object case. The plots are generated for the set `AsteroidRN82` using all 12 runs within.

4.2.2 Resolution of Planetesimal Identification

The mass of an object in a simulation run is directly accessible as the sum of masses of all simulation particles contained within that object. The diameter is calculated from the mass assuming a sphere with constant solid density ρ_s such that

$$D = 2 \cdot \left(\frac{3}{4\pi} \frac{M_{\text{Object}}}{\rho_s} \right)^{\frac{1}{3}}. \tag{4.16}$$

Since in the simulation runs, the mass of a simulation particle is constant and all simulation particles carry equal mass m, the object mass is $M_{\text{Object}} = N_{\text{Object}} m$, and the object diameter is solely dependent on the count of simulation particles in the respective object. The minimal size that is resolved in the simulations is one simulation particle. The corresponding diameter, in this case, is 2.15 kilometers. However, setting the minimal object size to one simulation particle in `DBSCAN` causes a high amount of single-particle objects that may actually be part of a larger object, thus misidentifying them. On the other hand, a minimal size that is too large possibly excludes meaningful objects. This section's task is to find a value for the minimal count of simulation particles N_{min} that must be contained within one object to minimize the misidentification of objects.
Figure 4.4 shows the amount of unidentified solid simulation particles as a function of differently chosen N_{min}. Also shown is the count of objects that formed for

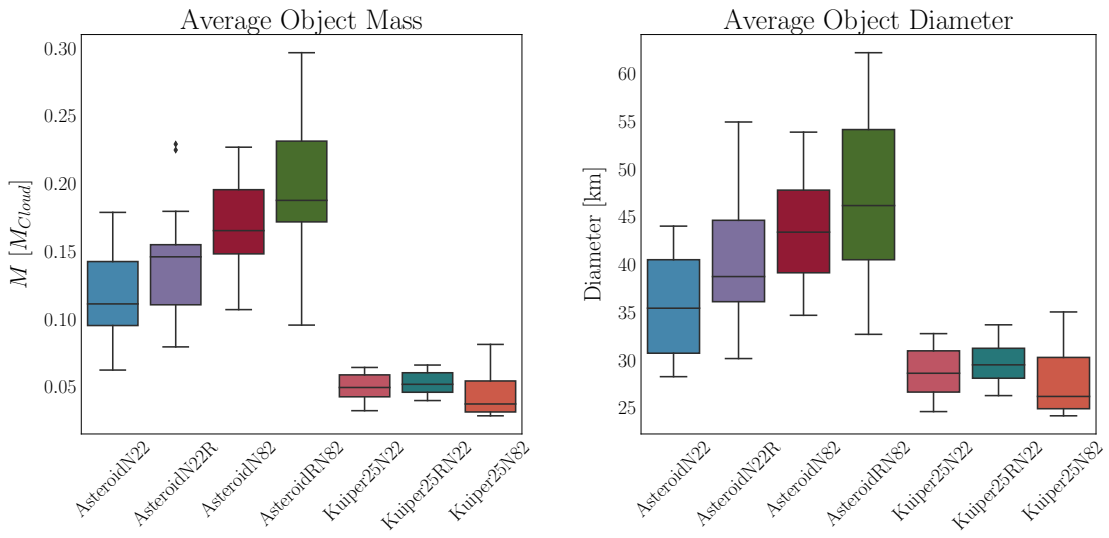


Figure 4.5: Average planetesimal mass normalized to cloud mass (left) and average object diameter in kilometers (right). The diameter is calculated from planetesimal mass before the average is taken. The results for the simulation sub-runs (e.g., `AsteroidN22A-M`) are depicted as a boxplot for every simulation run (e.g., `AsteroidN22`).

different N_{min}. The plots are created using all runs in the set `AsteroidRN82` (12 runs). The amount of objects identified is constant at $D_{eq} < 8$ km, takes a steep drop between $70 < N_{min} < 100$ (8 km $< D_{eq} <$ 10 km), and is zero afterward. Likewise, the amount of unassigned particles grows starting from $N_{min} \sim 10$ until no solid particles can be assigned to an object. The same behavior is observed for the `Kuiper` runs, where an example is given in Figure C.1.
Figures 4.4 and C.1 show, that with the chosen threshold of $N_{min} = 10$ and corresponding ϵ_{min} (see Section 4.1.3) all formed objects are identified and the error in terms of unassigned solid particles is as small as possible.

4.2.3 Object Mass and Diameter

Since multiple simulation sub-runs for every simulation run are conducted, some of the results are given in the form of boxplots (see Figure 4.5) created with the `seaborn` (Waskom [2021]) and `pandas` (pandas development team [2020]) `Python` packages. This type of plot is well suited to show relevant statistical properties of a set of values. An example is shown in Figure 4.5. Each bar in these plots represents a complete simulation set with sub-runs `A` to `M` for the `Asteroid` simulations, and `A` to `K` for the `Kuiper` simulations. The colored area in each bar is the interquartile range (IQR) restricted by the 25th percentile (Q1) at the bottom and the 75th percentile (Q3) at the top. The whiskers reach up to Q3 + 1.5×IQR and down to Q1 - 1.5×IQR. The filled rhombs represent outliers and the horizontal line within

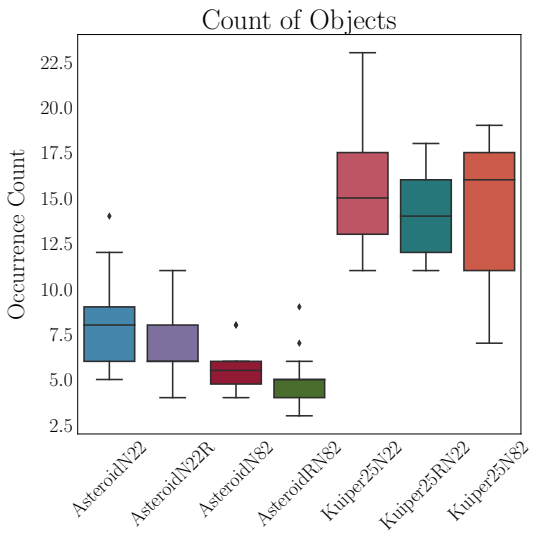


Figure 4.6: Object count for all simulation runs, grouped by simulation sets.

each IQR represents the median.
Figure 4.5, shows the average mass (left) of objects in one simulation run in terms of the total mass in the respective simulation run alongside the average object diameter (right) calculated using Equation (4.16). The average masses normalized to the total cloud mass are substantially higher for the diffusive `Asteroid` runs compared to the non-diffusive runs. For the `AsteroidN82` run, the average object mass varies between $\sim 18\%$ and $\sim 23\%$ of the cloud mass with a median of averages of $\sim 20\%$. For the initially relaxed `AsteroidRN82` run, a slightly smaller mass fraction of $15 - 20\%$ is found. The non-diffusive `Asteroid` runs show averages of $\sim 10\%$ to $\sim 15\%$ and thus form noticeably smaller objects. Since the total mass contained in solids is approximately the same for all `Asteroid` runs (Figure 4.3), one can directly conclude that although diffusivity has no effect on the efficiency with that solids are formed, it influences the size of formed objects. Consequently, the count of objects in the diffusive `Asteroid` runs has to be lower compared to non-diffusive `Asteroid` runs. This can be easily seen in Figure 4.6 which confirms an inverse one-to-one relation between object count and object mass for all runs.
For the `Kuiper` runs this influence of diffusivity is not present. However, the objects in the `Kuiper` runs are significantly smaller on average compared to the `Asteroid` runs. This is due to the fact that many more small objects are created in the `Kuiper` runs. This can likewise be seen in the average object diameters (Figure 4.5). Concluding this section, one can say that; Diffusivity in the `Asteroid` runs influences the mass and count of formed objects. Objects formed further away from the star have a smaller mass and are smaller in diameter due to the excess production of small objects. The effect of distance to the star covers the effect of diffusivity.

4.2.4 Most Likely Size and Mass Distribution

Observations of primordial Asteroids can be analyzed in terms of the cumulative size distribution in diameter space as is done by Delbo' et al. [2017] and Delbo et al. [2019]. Following this paradigm, this section analyses the planetesimal size distribution in terms of the fraction of the total collapsing cloud mass and the corresponding equivalence diameter. This diameter is used in Section 4.2.5 to calibrate the simulations to observations.
Similar to Polak and Klahr [2023] (see also Anders [1965] and Hartmann and Hartmann [1968]) the planetesimal mass is assumed to be distributed around a most likely value (M_{50}), following the Gaussian distribution function

$$M/\mathrm{d}M = \frac{A}{\sqrt{2\pi\sigma_M^2}} \exp\left[-\frac{1}{2}\frac{(M-\mu)^2}{\sigma_M^2}\right]. \tag{4.17}$$

The mean value is $\mu = M_{50}$, the spread $\sigma_M = M_{50} - M_{84}$ and A is a scaling parameter. The meanings of M_{50} and M_{84} are elaborated on further below. One has several options to analyze the dependence of M_{50} and M_{84} on the collapsing cloud parameters. The first is to construct a histogram of object masses and weigh the bins of that histogram to the mass contained within that bin. One can then calculate M_{50} and M_{84} through the mean and standard deviation from the histogram bins. The second possibility is to fit Equation (4.17) to the histogram and obtain M_{50} and σ_M as fitting parameters. The major disadvantage of both these methods is that for a limited number of data points, they heavily rely on the chosen binning and are thus not well suited for analysis in this case.
The third and fourth possible methods act on the planetesimal masses' cumulative distribution function (CDF). For a Gaussian distribution - which, in this case, has a physically imperative minimum mass and radius cap at zero - the CDF is given by the error function. Because of the natural lower cap at $M = 0$ and $D = 0$, it makes sense to utilize the complement error function (erfc)

$$f_{\Sigma m_{>M}} = \frac{A}{2}\mathrm{erfc}\left(\frac{1}{\sqrt{2}}\frac{M - M_{50}}{M_{50} - M_{84}}\right) + \underbrace{C - B \cdot M}_{\text{linear tail}}. \tag{4.18}$$

In Equation (4.18), a linear tail with generic parameters B and C is added to the erfc to enhance the stability of the fitting routine. When no linear tail is added, in some cases no convergence in the fitting routine is obtained. This is due to the low mass end discrepancy to the erfc, which is especially prominent for the `Kuiper` runs. To obtain the complementary CDF (CCDF) from the data, one takes the ordered object mass distribution (minimum to maximum mass), norm it by the total cloud mass, and excludes unidentified solid simulation particles as well as all objects with mass $M_{object} < 0.0025 M_{Cloud}$. Then, the right cumulated mass for each object mass

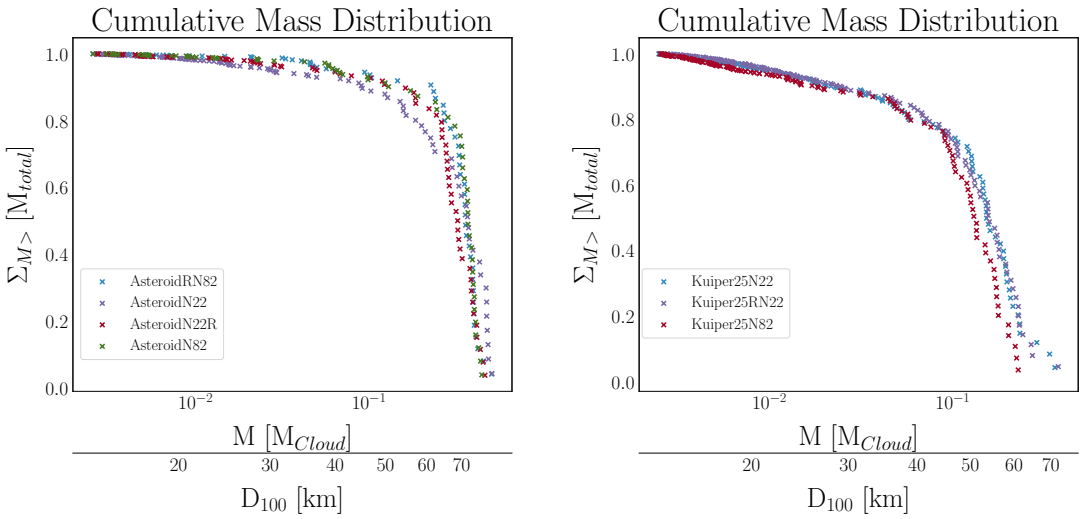


Figure 4.7: Cumulative mass distributions for the `Asteroid` (left) and `Kuiper` sets (right). Each depicted run contains all objects of all sub-runs within the respective simulation run.

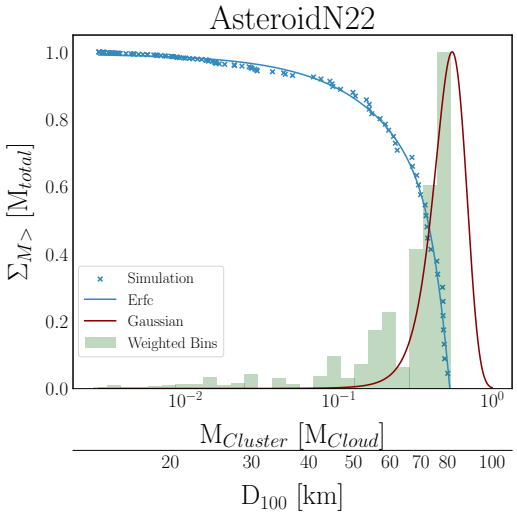


Figure 4.8: Example of cumulative mass distribution and analysis measures. Data points and Fit function, as well as the Gaussian function and histogram, are normalized. The Fit function is given by Equation (4.18), the Gaussian is computed using M_{50} and M_{84} from the fitting routine according to Equation (4.17). Created using data from one of the `AsteroidN22` runs.

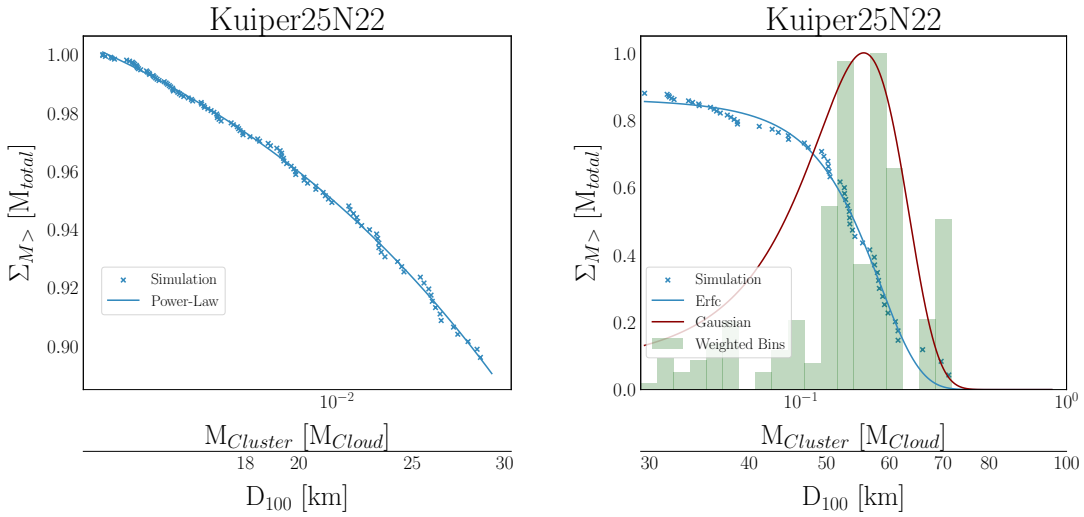


Figure 4.9: Cumulative mass distribution for the `Kuiper25N22` simulation runs fitted with a power law towards small object masses with $0.0025 M_{Cloud} < M < 0.05 M_{Cloud}$ (left) and fitted with the erfc (not using a linear tail extension) at higher masses of $M > 0.05 M_{Cloud}$ (right).

Table 4.2: Summary of conducted and analyzed simulations within this work. The values of M_{50} and M_{84} are given as a fraction of the total cloud mass, and the Diameter D is given in kilometers and valid for a maximum cloud mass equivalent to a 100 km-sized object. The diameter value can be used to calibrate the simulations to observation. Values in the 'Fit' column are obtained as fitting parameters from Equation (4.18). Values in the 'Direct' column are inferred from the raw data points similar to Polak and Klahr [2023], and those in 'CCDF' are the values of the fitted function at the respective mass level.

	Fit		Direct		CCDF		Fit	Direct	CCDF
Set	M_{50}	M_{84}	M_{50}	M_{84}	M_{50}	M_{84}	$D_{M_{50}}$	$D_{M_{50}}$	$D_{M_{50}}$
Asteroid10N22	0.366	0.235	0.331	0.208	0.332	0.201	71.507	69.15	69.275
Asteroid10RN22	0.329	0.145	0.298	0.164	0.309	0.155	69.016	66.828	67.608
AsteroidN22	0.549	0.418	0.374	0.158	0.38	0.147	81.86	72.033	72.424
AsteroidN22R	0.337	0.259	0.326	0.205	0.328	0.212	69.57	68.815	68.955
AsteroidN82	0.39	0.354	0.373	0.244	0.378	0.243	73.055	71.995	72.273
AsteroidRN82	0.374	0.331	0.361	0.27	0.366	0.266	72.08	71.201	71.518
Kuiper25N22	0.172	0.136	0.165	0.048	0.177	0.059	55.666	54.885	56.125
Kuiper25N82	0.142	0.115	0.135	0.049	0.152	0.049	52.198	51.306	53.333
Kuiper25RN22	0.166	0.041	0.175	0.06	0.176	0.062	54.949	55.895	56.05

M_i can be calculated as the sum of all objects heavier than M_i. Figure 4.7 shows the resulting CCDFs. With this, the third method to obtain M_{50} and M_{84} is taking the mass values in the cumulative mass distribution where 50% and 84%, respectively, of the total mass, are reached (direct method in the following). The fourth method is to fit Equation (4.18) to the cumulative mass distribution and obtain M_{50} and M_{84} as parameters (fit method). Both these methods erase the binning problem and are thus better suited for the analysis in this chapter.
Since the fit sometimes depends on the linear extension described above, both methods are compared. When the obtained results are close, the effect of the linear tail is small, and the assumption of a Gaussian distribution holds in the analyzed mass range. A Gaussian distribution can not be assumed over the whole mass range when the values diverge.
To visualize the analysis strategy, Figure 4.8, shows an example of obtained data points, the fitted function (Equation (4.18)) alongside the corresponding Gaussian function (Equation (4.17)). In addition, the mass-weighted histogram is shown to visualize the meaning of the Gaussian distribution. Equivalent plots for all other simulation runs are depicted in the Appendix (Figures C.2 and C.3). If the initial assumption of uniform distribution around a favored mass fraction holds, one would expect the histogram to match the fitted Gaussian distribution function. The agreement is good for the `Asteroid` runs, but not for the `Kuiper` runs. This disagreement is due to the large amount of small objects in the `Kuiper` runs, which appear to influence the overall distribution more than for the `Asteroid` runs.
Because of this, the cumulative mass distribution for the `Kuiper` runs is split in a lower mass ($0.0025 M_{Cloud} < M < 0.05 M_{Cloud}$) and a higher mass ($M > 0.05 M_{Cloud}$) component as Figure 4.9 shows. The higher mass component is fitted again using the erfc. In this case, where the CCDF is split into two regimes, there is no need for a linear tail extension. The lower mass component is fitted using a power-law of the form $f = C + B \cdot M^D$, which is in line with previous studies (Morbidelli and Vokrouhlický [2003], Johansen et al. [2015], Simon et al. [2017], Schäfer et al. [2017]). Equivalent plots for the remaining `Kuiper` runs can be found in the Appendix Figure C.4. The results for M_{50} and M_{84} obtained from the fit parameters, the direct read from the raw data, as well as from a readout of the fitted function, are summarized in Table 4.2. Additionally, the value of the fitted CCDF at the respective mass level is depicted in column `CCDF read`. The diameter values on the right of the table are calculated from the corresponding M_{50} values using Equation (4.16).
One sees very good agreement in almost all cases when examining the M_{50} values from the fit- and direct method for the higher resolution cases. Nevertheless, there is a clear outlier at `AsteroidN22`. More details on this particular case are displayed in Figure 4.8, where the cumulative distribution of masses and the corresponding linear-tailed error function is shown alongside the mass-weighted size histogram of objects and the inferred Gaussian Function. Looking at this figure, a failure of the fitting routine can be excluded as a reason for the discrepancy since the data points and the fit appear to be in very good agreement. However, the obtained erfc substantially overestimates the values of M_{50} and M_{84}. The effect is due to the lack of

objects on the high-mass end and is likely related to poor statistics.
Independent of this, the most likely diameters from the `Asteroid` runs are $\sim 50\%$ higher compared to those obtained from the `Kuiper` runs with masses being about a factor of 2.5 higher. The spread in masses is higher by a factor of 3 to 5. The most likely diameter and mass for the `Asteroid` runs are $D_{M_{50}} \approx 70.4_{\pm 2.7}$ km and $M_{50} \approx 0.35_{\pm 0.04} M_{\text{Cloud}}$, and $D_{M_{50}} \approx 53.1_{\pm 1.9}$ km and $M_{50} \approx 0.15_{\pm 0.015} M_{\text{Cloud}}$ for the `Kuiper` runs. In combination with the results in Section 4.2.3, one sees that in the `Kuiper` runs not only many more small objects ($D_{100} < 30$ km) are produced but also that the larger objects are born smaller. The results from Table 4.2 are also visualized in the Appendix Figure C.5, where the Coefficient of Variation

$$\text{CV} = \frac{\sigma}{\mu} = \frac{M_{50} - M_{84}}{M_{50}}, \tag{4.19}$$

is introduced as a measure for the relative distribution width which eliminates every dependency on normalization in M_{50} and M_{84}.
Summarizing the findings from this section, the `Kuiper` runs produce an excess of small objects, which does not allow for an interpretation of the size distribution as being purely Gaussian. Despite this, M_{50} and M_{84} values can be extracted. Not only is there a strong dependence of these values on the distance to the central star, but a small dependence on diffusivity is also observed. Runs incorporating diffusivity are more likely to produce slightly larger objects. The dependence of the distribution width M_{84} on diffusivity that is present in the `Asteroid` runs but overshadowed in the `Kuiper` runs, is an important result in this section. It is a strong indication, that the initial angular momentum has the largest effect on the initial mass function. This effect is not majorly influenced by the incorporation of diffusion, especially not at 25 AU.

4.2.5 Comparison to Observations

As outlined in Section 4.1.2, a cloud mass that is equivalent to a 100 km-sized object is chosen for the simulations. The actual mass value depends on the distance to the central star via the solid density at that distance. To calibrate the findings, the cumulative mass distributions obtained from the `Asteroid` simulation runs can be compared to observational data from Delbo et al. [2019] (Table B.5 therein). The diameter values compiled therein can be converted to a *virtual* mass using an arbitrary but non-zero test density. The sum of object masses can then be used for normalization, and the test densities are canceled out again. The value of the test density and *virtual* mass is thus irrelevant. From equivalent analysis as described in the previous section, a most likely diameter of $D_{M_{50},\text{observed}} = 125$ km is obtained for the observational data. This value can be used to calibrate the simulation data by applying the factor

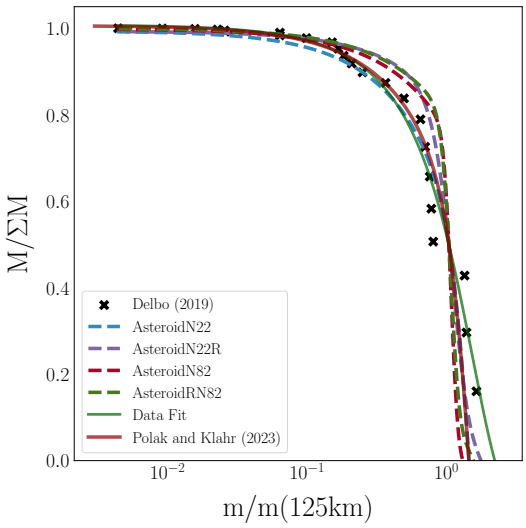


Figure 4.10: Observational data taken from Delbo et al. [2019] (violet). Data is converted to mass following Equation (4.16) assuming an arbitrary test density and normalized to the sum of all masses. The x-axis is normalized to the mass equivalent of a 125 km-sized object. Fit according to Equation (4.18) is shown in blue. The solid red line shows the inferred erfc from data from Polak and Klahr [2023].

$$A = \frac{125 \text{ km}}{D_{M_{50}}}, \tag{4.20}$$

to the diameter results for the `Asteroid` runs. The value of A can be evaluated to $A \approx 1.8$ for all `Asteroid` runs. Figure 4.10 shows the converted and normalized observational data together with the fitted CCDFs. In addition, the calibrated CCDFs for the `Asteroid` runs are shown. All fit functions obtained from simulated data approximate the observational data reasonably well with the seemingly best match obtained for the `AsteroidN22` run. This is confirmed when comparing the CV value obtained from the observational data of 60% to the values in Figure C.5, where the CV value for the `AsteroidN22` is also at 60%. This also agrees with the data from Polak and Klahr [2023] at 1 Au distance from the central star for which a CV value of 55% can be inferred.
Since the `AsteroidN22` run yields the best match to observation, it is used to find the scaling factor as $A = 1.75$. When this factor is applied to calibrate the `Kuiper` runs, one obtains a most likely planetesimal diameter of $90 - 100$ km.

Table 4.3: Total count of binary- and parent-satellite systems (PSS) summed over all simulation sub-runs within each simulation run. The count of PSS includes the count of binaries. The same planetesimal can be part of multiple PSS simultaneously. Column "All" is the total count of objects in the simulation run.

Run	Binaries	PSS	All
AsteroidN22	3	41	107
AsteroidN22R	4	30	87
AsteroidN82	6	39	67
AsteroidRN82	3	31	65
Kuiper25N22	2	55	171
Kuiper25RN22	5	42	156
Kuiper25N82	9	69	155

4.2.6 Parent-Satellite/- and Binary- Systems

The simulation runs encompass not only the formation of solid objects but also the formation of binary- as well as parent-satellite systems (PSS). These are characterized by the binding energy in Equation (4.14). The binding energy for each object to every other object in each sub-run is calculated to analyze the distribution of parent-satellite systems and binaries per simulation run. Only the $N = 10^5$ particle runs are considered. Table 4.3 shows the count of binaries, parent-satellite systems, and individual objects. The table gives the sum of all objects and systems in all sub-runs (`A-M` for `Asteroid` and `A-K` for `Kuiper`). The formation of parent-satellite systems is common throughout all simulation runs. Although at least one parent-satellite system is found in every simulation sub-run, the appearance of true binaries is much rarer with most sub-runs not producing any true binaries. Nevertheless, true binaries are found as well.
The orbits of the formed binaries are not necessarily determined by two body astrodynamics as can be seen, e.g., in Figure 4.11 which shows the time-integrated paths of the two objects in a binary system, detected in one of the unrelaxed and non-diffusive `Kuiper` sub-runs. The mutual influence of the binary objects is clearly visible, yet there appears to be a central mass at the coordinate origin to which the binary system behaves as a satellite. A similar effect can be seen for the `Kuiper25N82C` binary (bottom panel, left) in Figure 4.12. This figure shows a collection of some binary systems, each exhibiting interesting behavior. The runs `AsteroidRN82K` (upper panel, left) and `AsteroidN82I` (upper panel, middle) are dominated by two body dynamics and form the dominant entity in their respective simulation run by mass. The lighter mass appears in a secondary parent-satellite relation with a much closer object for both runs. The runs `AsteroidN22RB` (top panel, right) and `Kuiper25RN22C` (bottom panel, right) show a converging trajectory resulting in a potential merging event. The `Kuiper25N82I` binary (bottom panel, middle) forms relatively late ($\sim 1.16 t_{ff}$) and - although in binary relation - is completely domi-

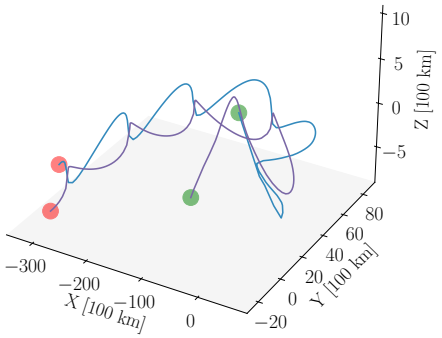


Figure 4.11: Time integrated trajectories of two bodies forming a binary system. The green points mark the birth position in space of the respective object, and the red dots are the point in time where the binary identification routine according to Equation (4.14) is carried out. The time difference between the birth of the two planetesimals is in the order of $10^{-3} t_{ff}$.

nated by a parent mass, causing the binary to fall almost in a straight line to what is the left-hand side in the shown projection.
Defining orbits directly associated with binary interaction is difficult for all observed binary systems. In fact, this would even only be possible for the `AsteroidN82I` run and also for the `Kuiper25N82C` (see Figure 4.12). For all other runs, higher-order body interactions can not be neglected. Similar to Polak and Klahr [2023], the ratio of masses, the sum of masses in terms of cloud mass, and the inclination of the binary system with respect to the gravitational vector due to the central star are analyzed. The inclination is calculated as

$$i \approx \tan^{-1}\left(\frac{|Z_1 - Z_2|}{\sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}}\right). \tag{4.21}$$

Here, all inclinations are computed at $t = 1.2 t_{ff}$ to be in line with the analysis conducted in the previous sections.
Figure 4.13a shows the relations of the obtained binary inclinations to the total binary masses. The radii and distances are scaled using the scaling factor in Equation (4.20) ($A = 1.75$) obtained in Section 4.2.5. A maximum value for the inclination of $\sim 50°$ is found. Most binaries are inclined by $0° - 5°$. On the other hand, one can clearly see that the formed binaries are relatively heavy, with almost all containing

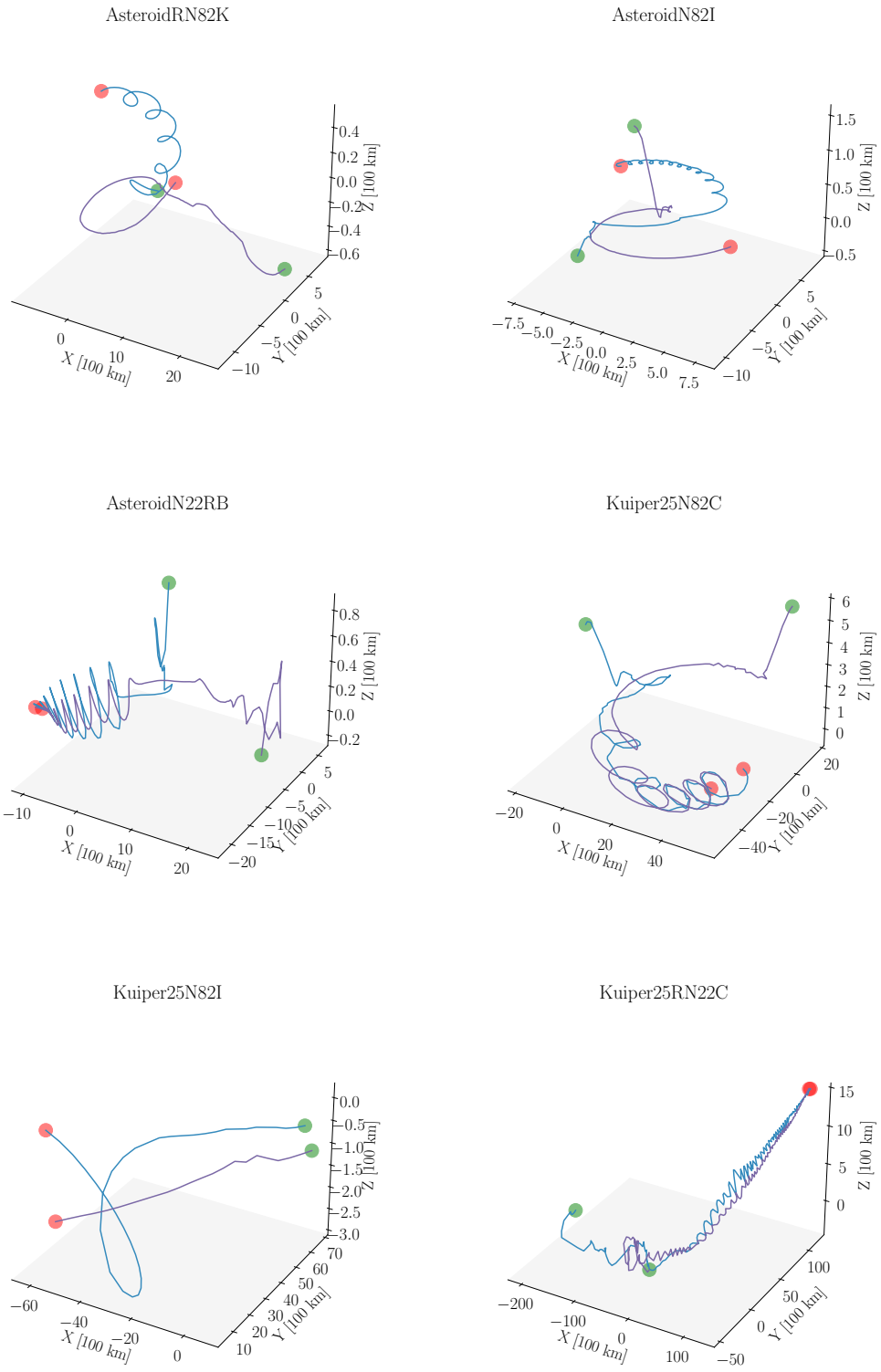
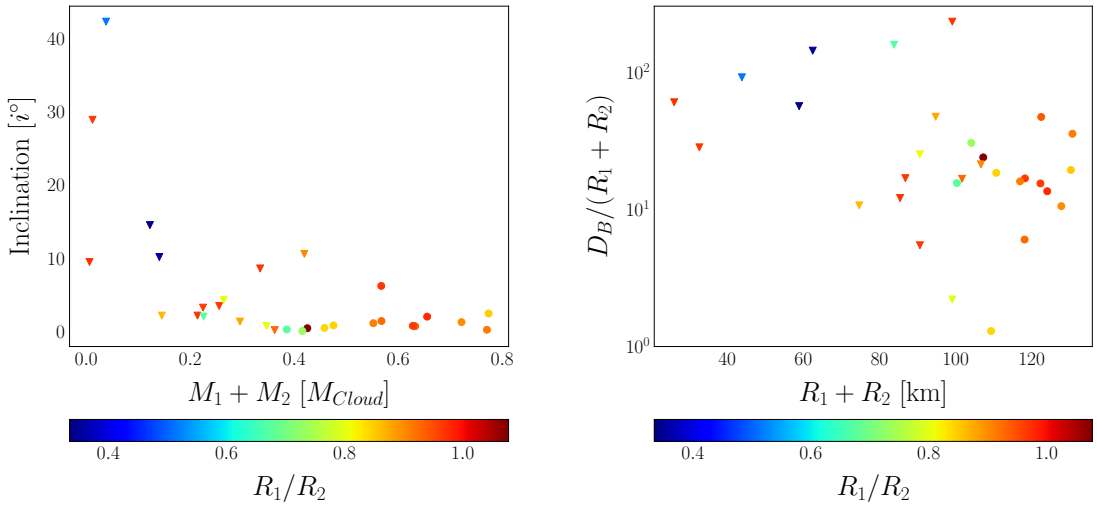


Figure 4.12: Time integrated trajectories of two bodies forming a binary system. The green points mark the birth position in the space of the respective object (not necessarily at the same time), and the red dots the point in time where the binary identification routine according to Equation (4.14) is carried out (time synced).



(a) Absolute value of binary inclination against total binary mass.

(b) Normalized binary against the sum of binary radii.

Figure 4.13: Triangles indicate data at 2.3 AU and circles at 25 AU. The color indicates the relative size of the constituents. Radii are calibrated using findings from Section 4.2.5.

more than 10 % of the initial cloud mass and many exceeding 50 % of that value. Noteworthy here is that there appears to be a clear separation between the binaries formed at 2.3 AU and those formed at 25 AU. The former are exclusively found with masses below 40% cloud mass. The latter are almost exclusively found above that value.

At first sight, this seems to contradict the finding that the preferred mass of formed planetesimals is larger at smaller distances to the central star. However, it appears that $M_{50}(2.3\text{AU})$ is so large that it is unlikely to form another object of comparable mass within the same simulation run. Such an object would be needed to form a binary system by the applied definition. If no object of comparable mass is formed, a parent-satellite system is detected, not a binary system. For $D_\star = 25$ AU, it is more likely to find at least two similarly weighted, heavy planetesimals with combined mass above $0.4M_{\text{Cloud}}$, which are then bound in a binary system. To be clear, two body systems with $M > 0.4M_{\text{Cloud}}$ exist in the `Asteroid` runs as well, but they tend to form parent-satellite rather than binary systems.

Figure 4.13b shows the relative distances between binary constituents, normalized to the sum of their sizes. This is called the separation factor, and a separation factor of one corresponds to a contact binary. The separation factor is 10-100 for most binaries, but there are at least two with separation factors of 2-5. The binary with the smallest separation factor consists of one 50 km object and one 60 km object at $\sim$ 120 km separation. The values are computed at a single time step without considering any variation in binary orbits. The distance between binary components

can thus only be seen as an order-of-magnitude approximation. Additionally, no statement can be made about the presence of contact binaries in the simulation runs since those systems are identified as single objects in DBSCAN. Because of this, contact binaries have to be analyzed the same way as mergers are analyzed. This is done in the next section. This section, however, shows a strong indication that contact binaries do exist in the simulations and proves that close binaries are formed and that the size and mass distribution of the formed asteroids match observational data very well.

4.3 Merger Characteristics and Boulder Accretion

With the merger-tree algorithm introduced in Section 4.1.4, questions about the evolution history of individual planetesimals can be answered. The analysis in this section compares the `Asteroid` and `Kuiper` runs in the diffusive and non-diffusive cases. Section 4.3.1 shows the frequency of occurrence of mergers as a function of time, and integrated over time. Section 4.3.2 investigates the mass scale at which mergers transfer the most mass and how many mergers a heavy object typically undergoes. Section 4.3.3 analyzes the mass gain of objects associated with merging events and Section 4.3.4 shows examples of the merger trees and mass evolutions of individual planetesimals.

4.3.1 Merger Occurrence

As already shown in Section 4.2, the time evolution of the solid mass fraction in the simulations depends on diffusivity and distance to the central star. Figure 4.14 shows, that this is also the case for the amount of merging events. Depicted are the time-cumulated merging events for all simulation runs, normalized to the total cumulative count. For all simulation runs a phase of high activity sets in at around one free-fall time - slightly earlier for the `Asteroid` runs. The diffusive runs exhibit a secondary phase where merging activity proceeds with a shallower slope. This is similar to the evolution of the solid mass fraction over time in Figure 4.3. Comparing the total count of merging events up to $t = 1.5t_{ff}$ (the norm in Figure 4.14), there appears to be no clear difference between diffusive and non-diffusive runs. Although diffusivity influences the rates at which merging events are observed, the total count of merging events does not strongly depend on diffusivity.
One may also ask about the merging history of the objects analyzed in Section 4.2. Those are the objects that are born before $t = 2t_{ff}$ for the `Asteroid` runs and $t = 1.2t_{ff}$ for the `Kuiper` runs and are not merged away until that point in time. Figure 4.15 shows the ratio of objects undergoing at least one merging event for all simulation runs. This ratio is between 40% and 70% for the majority of `Asteroid` and `Kuiper` runs. The `AsteroidN82` runs are clear outliers with a noticeably higher fraction of objects having participated in at least one merging event. Since this effect can not be observed for the diffusive `Kuiper` runs, it is likely of a statistical

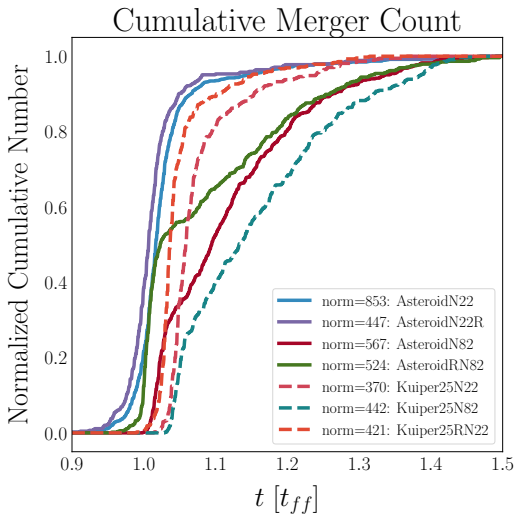


Figure 4.14: Cumulative distribution of merging events for all simulation runs normalized to unity. A clear difference is visible between the diffusive and non-diffusive runs. Norm gives the average of the total count of mergers over all sub-runs. `Kuiper` end slightly before 1.5 t_{ff} and are continuatively extrapolated to 1.5 t_{ff}.

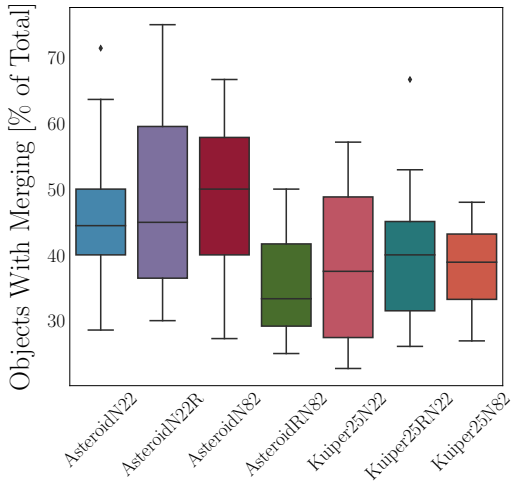
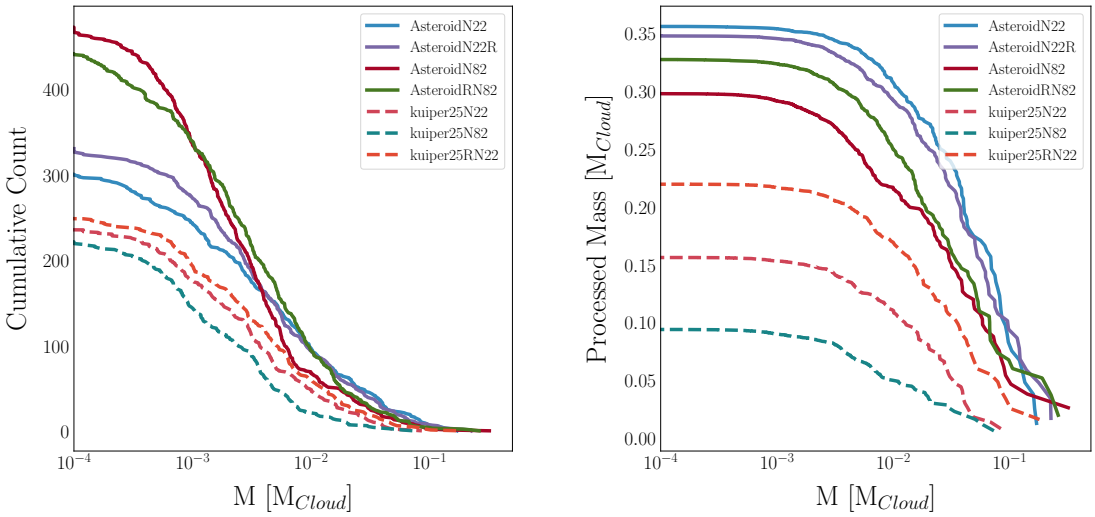


Figure 4.15: Ratio of Objects at $t = 2t_{ff}$ for the `Asteroid` runs and $t = 1.2t_{ff}$ for the `Kuiper` runs undergoing at least one merging event to that point in time. No objects are considered born after the respective free fall time or merged away before it.



(a) Distribution of merged objects by count. (b) Distribution of merged objects by mass.

Figure 4.16: Cumulative distributions with respect to the mass of the smaller object involved in a merger event. Considered are all merging events to $t = 2t_{ff}$ for the `Asteroid` runs and $t = 1.2t_{ff}$ for the `Kuiper` runs. Mass is processed equally above planetesimal sizes $M \sim 10^{-2}\ M_{Cloud}$.

nature. One can conclude from Figure 4.15, that merging events are common for the formed planetesimals.

4.3.2 Mass Scale of Mergers

By definition of the merger tree algorithm, every merging event involves two objects. The heavier objects are called *parents*; their ID is carried to the new object after merging. The lighter of the objects is called a 'merged away object' in the following. This section aims to clarify if there are typical mass scales of parents and merged away objects.

The cumulative number and mass distributions are analyzed for the merged away objects. The corresponding distribution functions are shown in Figures 4.16a and 4.16b. Taken into account are all merging events below a simulation time of 2 t_{ff} for the `Asteroid` runs and 1.2 t_{ff} for the `Kuiper` runs. Most merging events occur at small mass scales, down to the object identification limit of $10^{-4} M_{Cloud}$. Despite this, the smallest objects are not responsible for the most mass transfer. The seemingly linear drop, in Figure 4.16b, indicates, that objects with masses $10^{-2} M_{Cloud} < M_{Object} < 10^{-1} M_{Cloud}$ transfer most mass in merging events. Thus, although most of the mergers are minor, the highest amount of mass is processed through relatively major mergers.

For the merger parents, a different measure is employed. The reason for this is that mergers are very common, and the cumulative distribution function in mass would

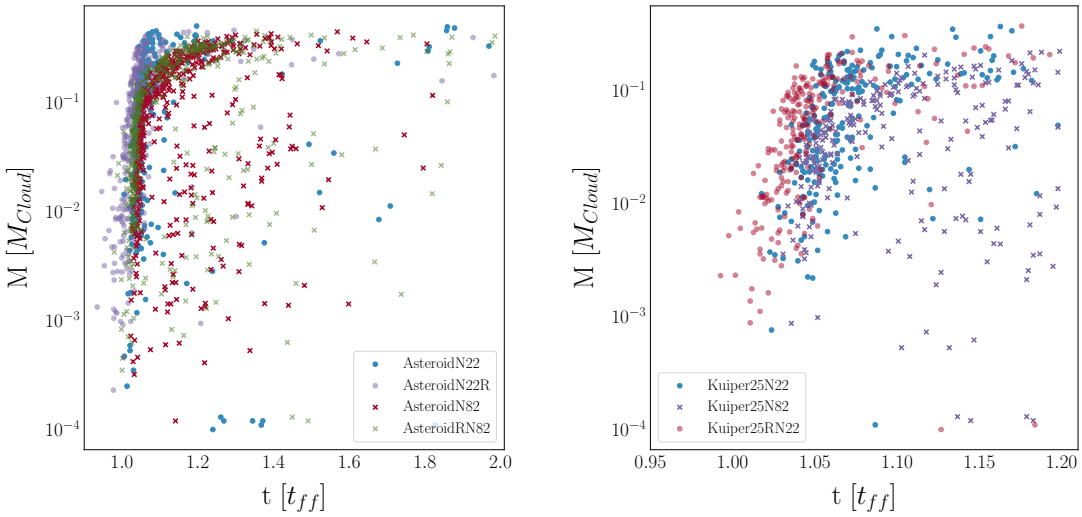


Figure 4.17: Size of larger object participating in merging event over time for the `Asteroid` runs (left) and the `Kuiper` runs (right). There appears to be no obvious correlation between merger mass and time. Crosses mark runs incorporating diffusivity.

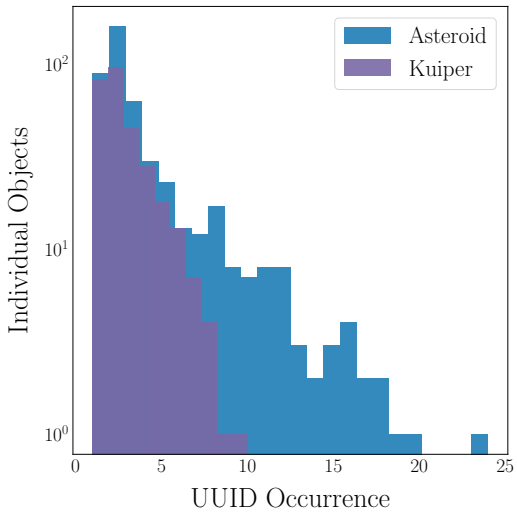


Figure 4.18: Histogram of the appearance count of individual objects that participate in merging events at times $t = 0.95 - 1.1t_{ff}$ (`Asteroid` runs) or $t = 1 - 1.08t_{ff}$ (`Kuiper` runs). Objects that participate only in a single merging event are not shown. Many objects participate in multiple merging events within the respective time frame.

just reproduce the results obtained in Section 4.2.4. Instead, it is more meaningful to look at the size of merger parents over time as shown in Figure 4.17. For these objects, which are growing in the merging event, a strong dependency on time can be observed and roughly split into three branches.
The first is a very steep branch ranging from $t = 0.95 - 1.1t_{ff}$ and $M = 10^{-4} - 0.5M_{Cloud}$ for the `Asteroid` runs and $t = 1 - 1.08t_{ff}$ and $M = 10^{-3} - 0.3M_{Cloud}$ for the `Kuiper` runs. For the `Asteroid` runs, the branch appears almost vertical and has a narrow spread. For the `Kuiper` runs, this region is tilted at roughly 45° and has a wider spread. This branch is associated with the initial steep ascent that can also be seen in Figure 4.14. It shows that planetesimals participating in merging events are distributed over a broad mass range shortly after the first detection of merging events. The second region is the branch sitting horizontally around the maximum merger mass in all runs and extends from the end of the first branch to the end of the measurement. For the Asteroid runs it is located at slightly higher masses of $0.2 - 0.4M_{Cloud}$, compared to $0.09 - 0.35M_{Cloud}$ for the `Kuiper` runs. The third region constitutes the rest of the depicted domain. It is almost exclusively dominated by objects from the diffusive runs, and, for the `Asteroid` runs, lighter objects become increasingly underrepresented approaching $t = 2t_{ff}$.
An important question to answer for the first branch is: does it describe an evolutionary track and show the same objects over and over again at different times, or does it show unrelated objects? In the former case, one would expect only lightweight objects early in the collapse phase, followed by their rapid agglomeration into larger objects with substructure. In the latter case, planetesimals are born in all sizes through pebble accretion, and merging events do not majorly contribute to growth. To answer this question, Figure 4.18 shows how often individual branch one objects participate in a merging event. In the figure, an object can appear as a parent in a merging event and as a merged away object. Both cases are counted equally. For the `Asteroid` (`Kuiper`) runs, a total of 457 (297) objects are categorized in region one. 368 (161) of them participate in at least two merging events. For both distances to the central star, reprocessing of objects is not rare but substantially more prevalent in the `Asteroid` runs, where three objects even participate in ~ 20 merging events. For the `Kuiper` runs, only a few objects undergo more than five merging events and none participates in more than ten.
As a concluding remark to this section, one can say that mergers at early times tend to happen between lightweight objects. Those at later times involve at least one heavier object ($M > 0.1M_{Cloud}$). Additionally, it is common for objects to participate in multiple merging events.

Table 4.4: Correlation r between object mass and count of undergone mergers (left), average mass an object accretes through merging events, and correlation between undergone mergers and gained mass, for all simulation runs.

Set	r	Avrg. Merger Accr. Mass per Object [%]	Mass Gain Corr-Coev.
AsteroidN22	0.92	15.5	0.15
AsteroidN22R	0.93	18.1	0.77
AsteroidN82	0.92	20.0	0.66
AsteroidRN82	0.95	15.9	0.78
Kuiper25N22	0.90	12.2	0.14
Kuiper25RN22	0.86	10.3	0.28
Kuiper25N82	0.84	12.5	0.015

4.3.3 Mass Gain Due to Mergers

As shown in Figure 4.18, object reprocessing is not rare in the simulation runs. The natural next step is to analyze the mass that individual objects gain through merging events. In the analysis, only objects, which are also analyzed in Section 4.2, that is, objects existing at $t = 2\ t_{ff}$ for the `Asteroid` runs and $t = 1.2\ t_{ff}$ for the `Kuiper` runs are considered. In this section, the relative mass gain through mergers of planetesimals is analyzed. In addition, the correlation (r-value, Pearson [1895]) between planetesimal mass and the count of mergers the respective object has undergone is considered.
Table 4.4 shows the respective values. For the `Asteroid` runs, all r-values are above 90% and $\sim$ 85% for the `Kuiper` runs. This makes the clear statement that bigger objects tend to have undergone more merging events. The average relative mass gain through merging events is $10 - 20\%$ for all simulation runs. It is higher for the `Asteroid` runs compared to the `Kuiper` runs. As a reminder, this corresponds to boulder accretion. The rightmost column in Table 4.4 shows the correlation between object mass and the mass gained through merging events. A clear difference between the `Asteroid` runs and the `Kuiper` runs can be seen. For the former, it can be stated that heavier objects have gained a higher fraction of their mass through merging events. An exception is made by `AsteroidN22`, where the correlation is only 15%. For the `Kuiper` runs, the correlation between the mass of objects and the mass fraction gained through merging events is too weak to make a conclusive statement. A more detailed depiction of the relative mass gain per object than the average value in the mid-column of Table 4.4 is given in Figure 4.19. This figure depicts the number of objects experiencing a certain mass gain due to merging events. Objects that do not gain any mass from merging events are excluded for clarity. As mentioned in Section 4.3.2, those objects comprise about half of all planetesimals. For the `Asteroid` runs, there is a maximum at a merger-accretion mass fraction of $\sim$ 30%. For the `Kuiper` runs, the maximum is at the lower end of accretion

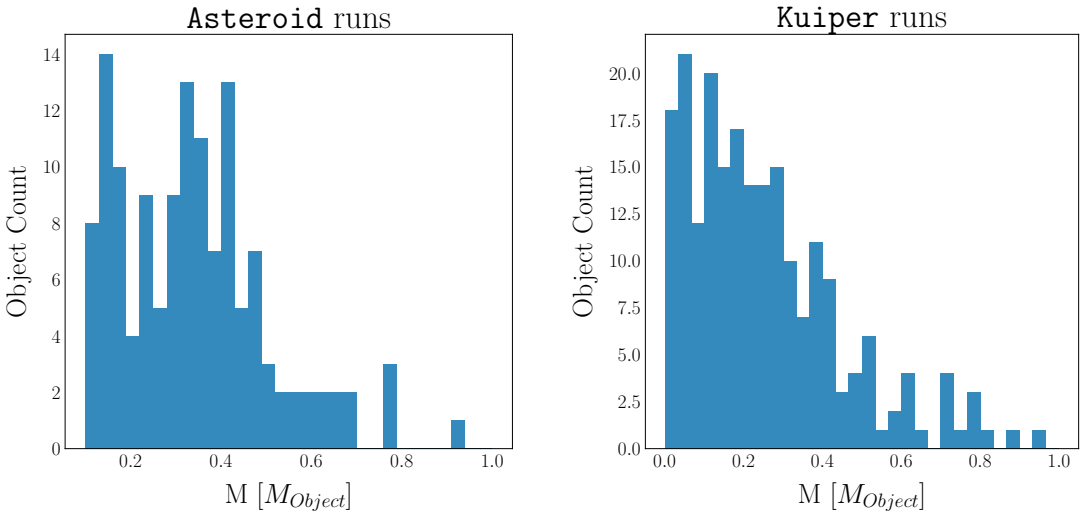


Figure 4.19: Mass Gain due to mergers for `Asteroid` runs (left) and `Kuiper` runs (right). Objects where no merger is encountered are removed completely because they dominate with 30-50 planetesimals in both cases. Also taken out are objects where the depicted ratio exceeds one, some objects in a total, of 5 or so. These are objects where a disruptive event is misidentified as a merger. Naturally, the rest of the mass gain is due to pebble accretion. Also, in Table 4.4, there should be the percentage of total mass accreted through pebble accretion and that through merger accretion inclusive and exclusive of the zero merger case. Criteria for objects taken into account are the same as in Figure 4.15.

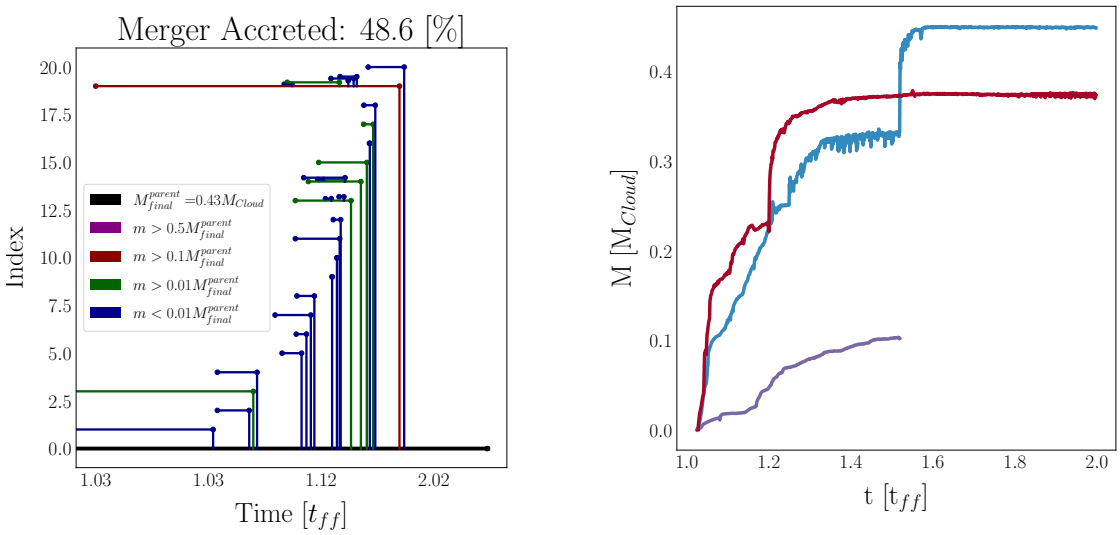
fractions. In both cases, there are objects that have acquired almost all their mass through merging events. These objects tend to be small objects that have undergone multiple merging events, swallowing a number of even smaller objects.
The results, presented in this section have to be taken with a grain of salt since the `Kuiper` runs are integrated to a smaller multiple of the free fall time compared to the `Asteroid` runs. Although Figure 4.14 indicates that the majority of merging events have already happened at this time, merging events that process bigger constituents are expected at later times. Because of this, the simulation time of the `Kuiper` runs is extended in future work. A first inspection of single `Kuiper` runs that are integrated to a higher multiple of the free fall time (not presented in this thesis) indicates, that none of the results presented in the former sections change dramatically. However, this is yet to be confirmed through thorough analysis.

4.3.4 Formation of Individual Objects

Sections 4.3.2 and 4.3.3 show, that in the simulations, planetesimals are formed very diversely with different fractions of mass gained from pebble- and merger-, or boulder-accretion. They also show that this formation diversity exists for large and small objects. This section shows examples of the merging history of a selection of planetesimals. To select merger trees, the full merger trees for all objects in each simulation run are generated. The focus then is on merger trees, where at least one major merging event (those with $\frac{M_{\mathrm{Small}}}{M_{\mathrm{Big}}} > 0.1$) is present. Finally, some trees where both larger participants have undergone merging events are manually selected.

Explanation of the Merger Tree Plot

Since the depiction of the merging history of individual objects encompasses a lot of information, the corresponding plots are complicated and are explained here. Figure 4.20a shows the full merging history of an example planetesimal in one of the `Asteroid` runs. This object is named here the *parent* object. The title shows the total mass fraction that the parent object accreted through merger accretion. The index at the ordinate is the index of the objects, that are merged into the parent object. Integer index objects are merged directly into the parent object and non-integer objects are merged through a secondary process. This can be best seen at indices 10 to 10.5 in Figure 4.20a. The parent object of the tree has an index of zero in all merger trees. The horizontal lines correspond to the lifetime of an object and the vertical lines to the point in time where an object is merged. The color of the parent object is always black and in the legend, its final mass is given with respect to the cloud mass. The color of the merged-away objects is taken with respect to the final mass of the parent object.
Figure 4.20b shows the mass evolution of the parent object and two other heavy objects in that simulation run. The blue line in this figure corresponds to the mass of the parent object in Figure 4.20a. The red line in Figure 4.20b is another heavy object that is unrelated to the depicted merger tree. The purple line corresponds to



(a) Meger Tree (b) Mass Evolution

Figure 4.20: Merger tree for an example planetesimal of one of the diffusive `Asteroid` runs (left) and mass evolution of individual objects in the corresponding simulation run (right). The object has a final mass of $0.45M_{Cloud}$ and has gained almost 50% of it through merger accretion. A total of 20 smaller boulders are merged directly into this object. One of them brings in at least 10% of the final mass and consists of five smaller objects itself. Only the most massive merged object (violet line on the right) produces a clearly visible rise in the mass level.

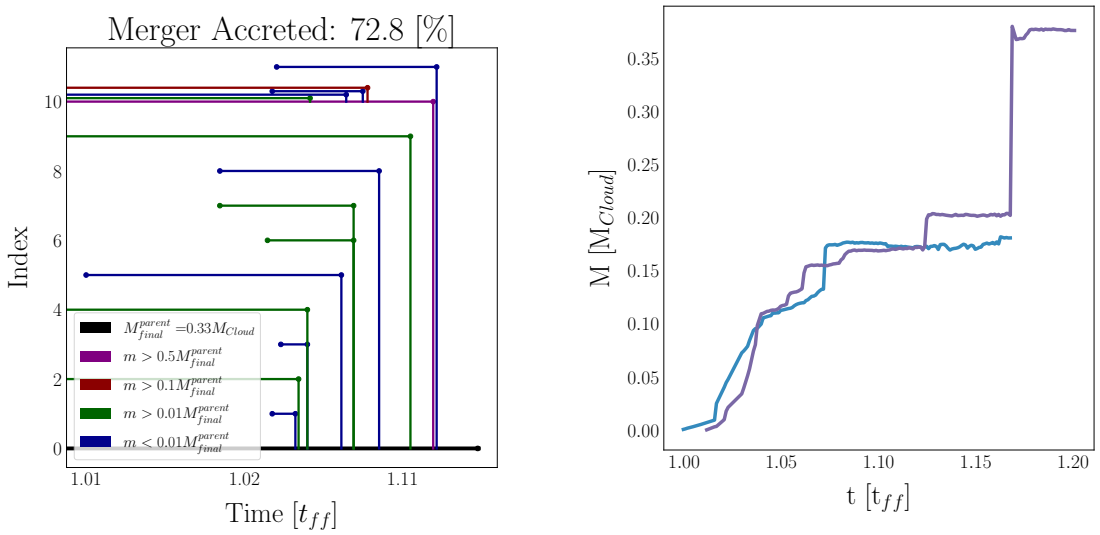


Figure 4.21: Example merger tree from one of the non-diffusive `Kuiper` runs (left). The final mass of the object is $0.38M_{Cloud}$ and the biggest object swallowed contains slightly more than 50% of the final object mass. For this object, the mass loss after the merger is due to disruption, carrying away a small fraction of the object's mass, as can be seen in the mass evolution plot (right). The points where pebble accretion and merger accretion occur are well visible.

the heaviest merged away object in Figure 4.20a. It causes a clearly visible spike in the mass of the parent object. Generally, pebble accretion and merger accretion (or boulder accretion) are not mutually exclusive and happen in parallel.

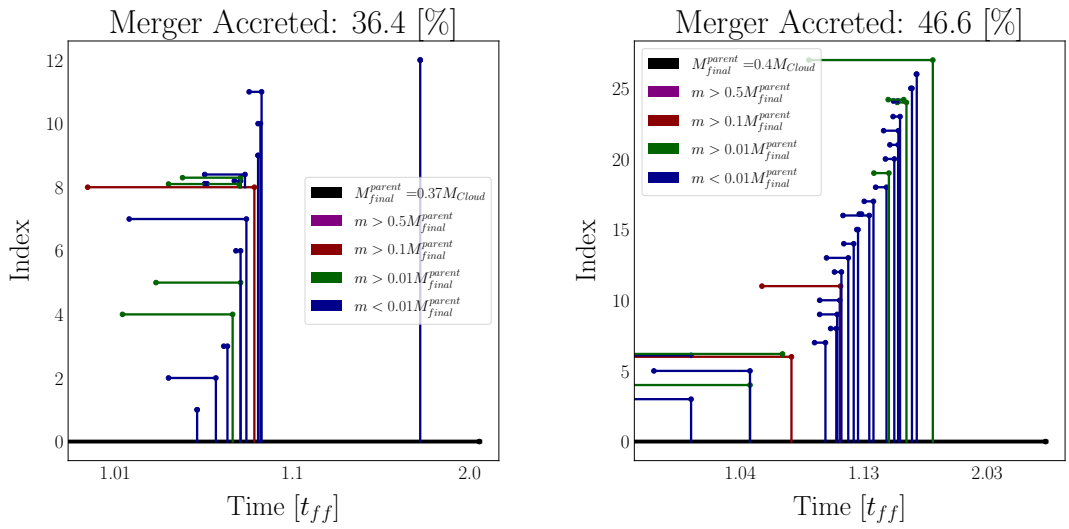


Figure 4.22: Examples of merger-trees for objects from the non-diffusive (left) and diffusive (right) `Asteroid` runs. The object on the right undergoes many merging events with very light objects. These light objects appear to form near the reference object and are merged shortly after birth.

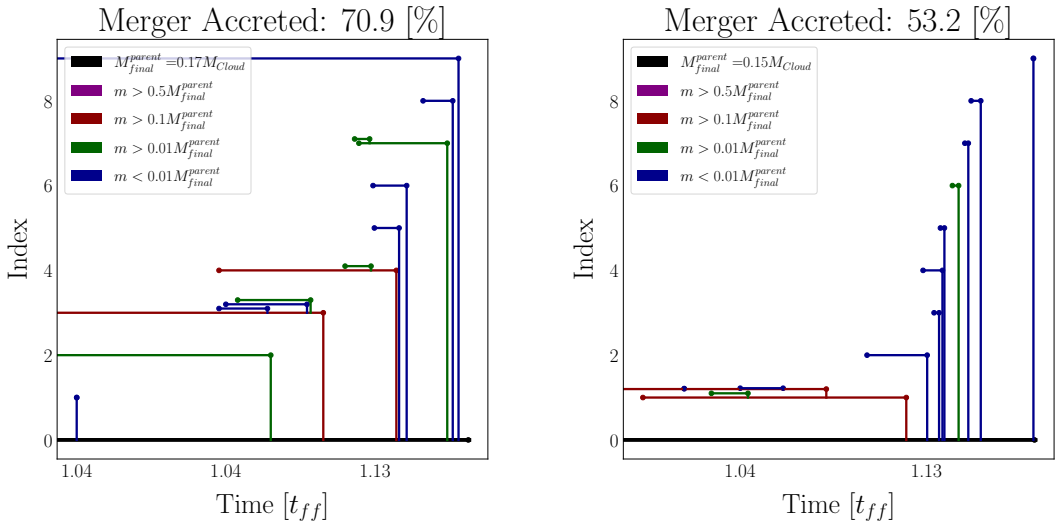


Figure 4.23: Merger tree examples taken from the diffusive `Kuiper` runs. Both reference objects gain a substantial fraction of their final mass through merging events. The object on the left undergoes two major merging events, and the object on the right.

In Figure 4.20a. Most swallowed objects contain less than 1% of the baseline object's mass when merging. There are, however, six merged objects with more than 1% mass fraction and one with more than 10%. Some of the constituent objects have undergone merging events themselves. An equivalent example for one of the non-diffusive `Kuiper` runs is given in Figure 4.21. The effect of individual merging events can be seen even more clearly in the mass evolution of the reference object (violet line on the right), which takes a step-like shape. The objects defining the displayed major-merging event are the two most massive objects in the simulation run.
Further examples of comparable merger trees for two of the `Asteroid` runs are shown in Figure 4.22 and for two of the `Kuiper` runs in Figure 4.23.

4.4 Conclusion

This chapter analyzes the products from the collapse of a cloud of pebbles within the gravitational field of a solar mass star at distances matching those of the Asteroid main belt and the Kuiper belt and compares collapses, where pebble-to-gas interaction is modeled to collapses, where this is not the case. The formation of multiple solid objects in all analyzed simulation runs is observed. The solid mass fraction's evolution history depends on the simulations' diffusivity. The distribution of solid mass after the initial collapse phase can be approximated by a normal distribution around a preferred value μ_M which strongly depends on the distance to the star but also weakly on the presence of diffusivity. Despite this preferred mass value, the additional appearance of an excess of small $(10 - 20$ km) sized objects gives a bimodal shape to the distribution function of the observed Kuiper belt equivalent objects.
Because of this, the cumulative mass distribution is modeled with a power-law for smaller objects and a Gaussian profile for larger objects. The cumulative size distribution for the `Asteroid` runs perfectly agrees with observations. The most likely diameters are calibrated to 125 km in the `Asteroid` simulation runs and 100 km in the `Kuiper` simulation runs. The simulated initial mass function appears to be majorly influenced by the distance to the central star but only at a distance of 2.3 AU by other effects, such as incorporated diffusivity. At a distance of 25 AU, the dependence on diffusivity is completely overshadowed. This indicates that the initial angular momentum content is the single most important defining factor influencing the planetesimal initial mass function.
In all simulation runs, parent-satellite- and binary systems are found. The binary systems are all similar in mass and have a ratio of radii of order unity. All binary systems are found at small inclination angles, which indicates that they are dominated by the ancestral angular momentum of the parent cloud beyond the formation stage. Visual inspection of time-integrated binary trajectories gives reason to believe that mergers can occur after the initial collapse. Multiple close binaries can be detected in the simulated Asteroid- and Kuiper Belt equivalents.
The count of merging events depends on the distance to the central star. Diffusiv-

ity can prolong the phase of *heavy bombardment* and significantly reduce the total count of merging events. The size of objects most likely to participate in a merging event is $10^{-3} - 10^{-2} M_{Cloud}$, corresponding to a diameter of $15 - 25$ km. A significant amount of the cloud mass is processed through merging events, and the mass of an object is strongly correlated with the number of merging events it has undergone in the course of its formation. To a lesser extent, this is also true for the mass an object has gained through merging events. The formation history of planetesimals is diverse. About half of the objects in the simulation runs are formed directly from pebble accretion. The other half experiences one or more merging events. Some objects gain the majority of their mass through merger accretion. This is not limited to heavy objects but applies to the range of formed planetesimals.

5 Summary and Outlook

In today's theoretical astronomy, efficient numerical modeling and analysis tools are essential. This thesis introduces novel methods to simulate the generating mechanisms of turbulence in disks and stars and develops analysis tools used in the context of planetesimal formation modeling. This thesis introduces the globally implicit matrix-free hydrodynamic Eulerian solver MATRICS as the main project. The modeling capabilities of MATRICS are used to present the first-ever global numerical simulations of the Goldreich-Schubert-Fricke instability in the optically thick environment of stellar radiative zones. Furthermore, this thesis develops and applies analysis tools, such as a merger tree algorithm, to examine the results of core-collapse simulations that reproduce the initial mass function and formation history of asteroids and Kuiper belt objects. This chapter summarizes the most important findings of this thesis and gives an outlook on future work.

5.1 Summary

Chapter 2; The MATRICS Code: The MATRICS code solves the NSE globally implicit and fully time-simultaneous. It is built on the finite volume method using a staggered grid. Interpolation of scalar variables is done by either the donor cell or the κ-scheme, which, in combination with a slope limiter of choice, composes a second-order in space-accurate MUSCL-type scheme. MATRICS integrates the NSE in time with the first-order accurate backward-Euler or the second-order accurate BDF-2 scheme. The necessary solution procedure, related to the implicit nature of the solver, is carried out using a matrix-free Newton-Krylov approach. The GMRES linear solver is implemented and tested. The elements of the Jacobian matrix, which would, in a matrix-based method, be necessary to carry out Newton's method, are calculated approximately to allow for the use of the ILUT preconditioner. Alternatively, a simplified physical Jacobi-preconditioner and the identity preconditioner are implemented to allow for a fully matrix-free operation mode. In the fully matrix-free operation mode, the extensibility of the physical equations and the addition of new equations are equal amounts of work compared to explicit methods. At the same time, it offers the full spectrum of advantages of the implicit approach.
Chapter 3; GSF and COS in Stellar Radiative Zone: This chapter introduces a simplified solar model that considers the equilibrium of gravity-, rotation-, and pressure forces. Still, it neglects self-gravity as well as magnetic effects and chemical gradients. In analogy to VSI models, density and temperature are parametrized in the radial direction using a polytropic prescription. Contrary to VSI models, this direction is the spherical radial direction instead of the cylindrical one. The

model introduces a control parameter for the relative strength of rotational support. Choosing rotational support in the order of $\sim 20\%$, analytical linear growth rates are estimated assuming an isothermal equation of state. The first numerical simulations of the GSF, using the full set of NSE, are then presented. They confirm the estimated linear GSF growth rates. With thermally relaxed simulations in the optically thick regime that is moderated by diffusion processes, a dependence of the linear growth rate on the Prandtl number is confirmed. For higher Prandtl numbers ($Pr > 10^{-3}$), a secondary growth phase that is likely linked to the COS is found within a relatively small simulation box of one spherical radial pressure scale height. An estimate of the growth of the COS fits well with the measured value, and the morphology of the growing modes resembles those identified as COS modes in the literature. Additionally, large simulation box simulations, that cover more than two and a half scale heights in the spherical radial direction and extend in the polar direction from $-\pi/12 < \theta < \pi/4$ at $Pr = 10^{-4}$, show the typical morphologies of GSF and COS.

Chapter 4; Formation of Planetesimals: This chapter analyzes the outcome of pebble-cloud-collapse simulations conducted to model the formation of planetesimals. A solar mass star is assumed, and distances corresponding to the asteroid-main- and the Kuiper belt are considered. In simulations corresponding to the main belt, the observed initial mass function of asteroids could be reproduced, and the size of the numerically formed planetesimals was normalized to the observational value of 125 km. Applying the same normalization factor to the planetesimals formed at a distance equivalent to the Kuiper belt gives a most likely size for them of ~ 100 km. An excess of small objects with masses, relative to the simulated mass, of $0.0025 M_{Cloud} < M < 0.05 M_{Cloud}$ is found in these simulations. The distribution of small objects in the Kuiper equivalent simulations follows a power-law shape. The effect is not observed in the simulations corresponding to the main belt. The simulated IMF corresponding to the asteroid's main belt shows dependence on incorporated particle-to-gas coupling. This effect is completely overshadowed by the additional angular momentum amount contained within the Kuiper belt equivalent simulations, indicating that angular momentum is the most important factor influencing the planetesimal IMF. Multiple binary and parent-satellite systems are found within all simulation runs. A merger tree algorithm, implemented within this thesis, identifies multiple objects that grow in various forms through the agglomeration of smaller boulders. Planetesimals in all simulation runs are found to have a diverse formation history. Some are formed solely through pebble accretion, while others partially or even solely through merger accretion. Multiple candidates with the necessary formation history to form a substructure are presented. Some of these objects may be contact binaries since they are found to form through the merging of comparably sized objects. A dedicated assessment of this is not currently possible as the conducted simulations consider the nature of the formed objects as fluid. Future models will consider tensile strengths and open exciting new possibilities to model and analyze the formation of asteroids and Kuiper belt objects.

5.2 Outlook

Chapter 2; The MATRICS Code: As a novel code, MATRICS will experience different stages of refinement. This section outlines the immediate and mid-term plans. Future work on the MATRICS code will require the implementation of more dedicated matrix-free preconditioners to enhance the convergence behavior of the underlying linear solver GMRES. Indeed, the current operational limitation of MATRICS is the count of GMRES iterations needed when a high resolution of the modeled grid is chosen and the fully matrix-free operation mode is selected. Implementing, e.g., the geometric multigrid method as a preconditioner will remedy this limitation. MATRICS currently provides OpenMP parallelized operation, but further refinement to enhance the scalability of the code is planned. In the mid-term, extensions of the solver to cover magnetic effects are within the range of possibilities as is the implementation of more complex and even adaptive grid structures.
Chapter 3; GSF and COS in Stellar Radiative Zone: In this chapter, strong rotational support and a low speed of sound are considered, making the preconditions in the considered spherical environment more favorable for the onset of GSF and COS compared to, e.g., the solar radiative zone. Future work in this area will need to consider more realistic fractions of rotational support and a shellular rotation profile as is expected in stellar radiative zones. This will be possible with minor adjustments to the simplified model presented in this chapter. Also, self-gravity, magnetic effects, and chemical gradients are neglected, allowing further model improvement.
Chapter 4; Formation of Planetesimals: In this chapter, the collapse of dense clouds of pebbles is analyzed. On the large-scale end of the simulations is the formation of these dense pebble clouds. As elaborated on in the introduction, these clouds likely form in high-pressure regions in the center of vortices that arise, e.g., due to the Vertical Shear instability. A very general connection between the effects that are considered in Chapter 3 and Chapter 4 becomes clear. Consistent modeling in disks, starting before the conditions for the onset of gravitational instability of the pebble clouds are met, will be of interest. On the small scale, improvement of the modeling of solid material in formed planetesimals will be considered. The inclusion of tensile strengths in the method will not only allow for realistic collision and fragmentation modeling but also allow for a more detailed analysis of the geological properties of the formed planetesimals. Modeling tensile strengths requires a combination of the utilized Lagrangian method within the GIZMO code and an implicit method to recover the stiff behavior of solid objects. The combination of methods used in MATRICS has proven very useful and efficient, and an extension of GIZMO with these methods may become relevant in future work on the formation of asteroids and Kuiper belt objects.

A Appendix to MATRICS Code

A.1 About the Implementation of MATRICS

- MATRICS contains $\sim 15,000$ lines of code, written fully object-oriented, in `C++-17`.
- Simulation parameters are passed through a `parameter.cfg` file and more general properties are defined at compile time in a `config.h` file.
- Data I/O is in `HDF5` format, which directly translates to the data structure of the code, where information is stored in `Cells`.
- Variables in MATRICS are stored within the `Grid`-class. The `Grid`-class consists of instances of the `Cell`-class. The `Cell`-class stores variables within a cell as described in Section 2.2.3.
- The `Grid`-class has the additional task of managing the complete functionality related to geometry. This means that mathematical operators (e.g., the gradient and divergence operators) related to geometry are implemented within the `Grid`-class.
- At setup, (a) `SystemOfEquations`-instance(s) (`soe`) is/are constructed from the grid, and the knowledge about the equations that shall be solved (defined in the `Config.h`-file). In Matrics-free operation mode, the `soe` acts merely as a proxy to manage the instances of the `Equation`-class. The relation between `SystemOfEquations` and `Equation` is comparable to that of `Grid` and `Cell`. In Matrix-based operation, the `SystemOfEquations` instance is also responsible for matrix assembly.
- The `SystemOfEquations`-class can operate on the complete `Grid`-instance or on a subdomain. This allows for out-of-the-box usage, e.g., of Schwarz-type methods.
- The `Equation` class is a virtual class with children: `ContinuityEquation`, `RadialMomentumEquation`, `VerticalMomentumEquation`, `AngularMomentumEquation` and `EnergyEquation`. The order in which these equations are solved is definable in the `SystemOfEquations`-instance
- Depending on the definition, MATRICS creates one or multiple instances of `SystemOfEquations`. This results in a full-simultaneous or operator-splitting

approach or a combination. Instances of `Equation` may appear in one-, or multiple `SystemOfEquations`-instances, or not at all.

- The MATRICS implementation heavily relies on the Factory-Pattern. This includes the solver and preconditioner, and the type of boundary conditions (reflective, periodic, etc.) chosen.
- `BoundaryConditions` are managed by the `SystemOfEquations` instance(s).
- One instance of the `Solver`-class is created for every `SystemOfEquations`-instance. Each instance of `Solver` contains one preconditioner. The preconditioner itself is implemented as child of the `Solver`-class.
- Some preconditioners, can also be used as a solver. In this case, Newton's method is converted to a defect-correction approach (This can be done, e.g., with ILUT).
- The Initial Conditions are passed through an `HDF5`-IC file to the code. A specifically implemented Python module is used to create the ICs. ICs must contain the relevant geometric information as well as the conserved variables.
- MATRICS can be restarted using its own output files as input, without adjustment or explicit specification.
- Loops in MATRICS are parallelized using `OpenMP`.

A.2 Operators in Different Coordinates

The axisymmetric expression for the gradient is

$$\nabla q = \frac{\partial q}{\partial x}\vec{e}_x + \frac{\partial q}{\partial y}\vec{e}_y \quad \text{(Cartesian)}, \tag{A.1}$$

$$\nabla q = \frac{\partial q}{\partial R}\vec{e}_R + \frac{\partial q}{\partial z}\vec{e}_z \quad \text{(Cylindrical)}, \tag{A.2}$$

$$\nabla q = \frac{\partial q}{\partial r}\vec{e}_r + \frac{1}{r}\frac{\partial q}{\partial \theta}\vec{e}_\theta \quad \text{(Spherical)}, \tag{A.3}$$

for the divergence

$$\nabla \cdot (q\vec{v}) = \frac{\partial q v_x}{\partial x} + \frac{\partial q v_y}{\partial y} \quad \text{(Cartesian)}, \tag{A.4}$$

$$\nabla \cdot (q\vec{v}) = \frac{1}{R}\frac{\partial \left(q v_R\right)}{\partial R} + \frac{\partial q v_z}{\partial z} \quad \text{(Cylindrical)}, \tag{A.5}$$

$$\nabla \cdot (q\vec{v}) = \frac{1}{r^2}\frac{\partial \left(r^2 q v_r\right)}{\partial r} + \frac{1}{r\cos\theta}\frac{\partial \left(q v_\theta \cos\theta\right)}{\partial \theta} \quad \text{(Spherical)} \quad , \tag{A.6}$$

and for the tensorial divergence

$$\begin{aligned} \nabla \cdot T = &\left(\frac{\partial T_{xx}}{\partial x} + \frac{\partial T_{xz}}{\partial z}\right)\vec{e}_x \\ &+ \left(\frac{\partial T_{zx}}{\partial x} + \frac{\partial T_{zz}}{\partial z}\right)\vec{e}_z \\ &+ \left(\frac{\partial T_{yx}}{\partial x} + \frac{\partial T_{yz}}{\partial z}\right)\vec{e}_y, \end{aligned} \tag{A.7}$$

in Cartesian coordinates,

$$\begin{aligned} \nabla \cdot T = &\left(\frac{1}{R}\frac{\partial (R \cdot T_{RR})}{\partial R} + \frac{\partial T_{Rz}}{\partial z} - \frac{T_{\varphi\varphi}}{R}\right)\vec{e}_R \\ &+ \left(\frac{1}{R}\frac{\partial (R \cdot T_{zR})}{\partial R} + \frac{\partial T_{zz}}{\partial z}\right)\vec{e}_z \\ &+ \left(\frac{1}{R}\frac{\partial (R \cdot T_{\varphi R})}{\partial R} + \frac{\partial T_{\varphi z}}{\partial z} + \frac{T_{R\varphi}}{R}\right)\vec{e}_\varphi, \end{aligned} \tag{A.8}$$

in Cylindrical coordinates and

$$
\begin{aligned}
\nabla \cdot T = & \left(\frac{1}{r^2} \frac{\partial (r^2 \cdot T_{rr})}{\partial r} + \frac{1}{r \cos\theta} \frac{\partial (\cos\theta \cdot T_{r\theta})}{\partial \theta} - \frac{T_{\theta\theta} + T_{\varphi\varphi}}{r} \right) \vec{e}_r \\
& + \left(\frac{1}{r^2} \frac{\partial (r^2 \cdot T_{\theta r})}{\partial r} + \frac{1}{r \cos\theta} \frac{\partial (\cos\theta \cdot T_{\theta\theta})}{\partial \theta} + \frac{T_{r\theta}}{r} \right) \vec{e}_\theta \\
& + \left(\frac{1}{r^2} \frac{\partial (r^2 \cdot T_{\varphi r})}{\partial r} + \frac{1}{r \cos\theta} \frac{\partial (\cos\theta \cdot T_{\varphi\theta})}{\partial \theta} + \frac{T_{r\varphi}}{r} \right) \vec{e}_\varphi \\
& - \frac{T_{\varphi\varphi} \cdot \tan\theta}{r} \vec{e}_\theta + \frac{T_{\theta\varphi} \cdot \tan\theta}{r} \vec{e}_\varphi,
\end{aligned}
\tag{A.9}
$$

in Spherical Coordinates.

A.3 Components of the Viscous Stress Tensor in Different Coordinates

The non zero components of the viscous stress tensor arising from the expression $\underbrace{\mu \left(\vec{\nabla}\vec{v} + (\vec{\nabla}\vec{v})^T \right)}_{\tau^{(1)}}$ (see Equation 2.6) are

$$\tau_{xx}^{(1)} = 2\mu \frac{\partial v_x}{\partial x}, \tag{A.10}$$

$$\tau_{yy}^{(1)} = 0, \tag{A.11}$$

$$\tau_{zz}^{(1)} = 2\mu \frac{\partial v_z}{\partial z}, \tag{A.12}$$

$$\tau_{xy}^{(1)} = \mu \frac{\partial v_y}{\partial x}, \tag{A.13}$$

$$\tau_{yz}^{(1)} = \mu \frac{\partial v_y}{\partial z}, \tag{A.14}$$

$$\tau_{zx}^{(1)} = \mu \left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right), \tag{A.15}$$

in Cartesian Coordinates

$$\tau_{rr}^{(1)} = 2\mu \frac{\partial v_R}{\partial R}, \tag{A.16}$$

$$\tau_{\varphi\varphi}^{(1)} = 2\mu \frac{v_R}{R}, \tag{A.17}$$

$$\tau_{zz}^{(1)} = 2\mu \frac{\partial v_z}{\partial z}, \tag{A.18}$$

$$\tau_{R\varphi}^{(1)} = \mu R \frac{\partial}{\partial R}\left(\frac{v_\varphi}{R}\right), \tag{A.19}$$

$$\tau_{\varphi z}^{(1)} = \mu \frac{\partial v_\varphi}{\partial z}, \tag{A.20}$$

$$\tau_{zr}^{(1)} = \mu \left(\frac{\partial v_R}{\partial z} + \frac{\partial v_z}{\partial R}\right), \tag{A.21}$$

in cylindrical coordinates and

$$\tau_{rr}^{(1)} = 2\mu \frac{\partial v_r}{\partial r}, \tag{A.22}$$

$$\tau_{\theta\theta}^{(1)} = 2\mu \left(\frac{1}{r}\frac{\partial v_\theta}{\partial \theta} + \frac{v_r}{r}\right), \tag{A.23}$$

$$\tau_{\varphi\varphi}^{(1)} = 2\mu \frac{v_r - v_\theta \tan\theta}{r}, \tag{A.24}$$

$$\tau_{r\theta}^{(1)} = \mu \left(r \frac{\partial}{\partial r}\left(\frac{v_\theta}{r}\right) + \frac{1}{r}\frac{\partial v_r}{\partial \theta}\right), \tag{A.25}$$

$$\tau_{\theta\varphi}^{(1)} = \mu \frac{\cos\theta}{r}\frac{\partial}{\partial \theta}\left(\frac{v_\varphi}{\cos\theta}\right), \tag{A.26}$$

$$\tau_{\varphi r}^{(1)} = \mu r \frac{\partial}{\partial r}\left(\frac{v_\varphi}{r}\right), \tag{A.27}$$

in spherical coordinates.

A.4 Backbone Algorithms of the MATRICS Code

This section shows the functions of the MATRICS code that constitute its backbone. They are taken without modification from the code base and may be used as you wish.

A.4.1 Implemention of Newton's Method

```
SolverParameters NewtonIteration::solve_system(
    int max_iter,
    double max_res,
    Eigen::VectorXd x_0
){
    double r;
    size_t iterations = 0;

    x_i = x_0;

    delta_x = system->get_system_defect(
        x_i
    );

    solver->prepare_newton_method();

    do {
        solver->prepare_newton_iteration_step();

        // damping:
        correction = solver->solve_for_rhs(
            delta_x,
            x_i
        );

        x_i += lambda * correction;

        system->update_variables( x_i );
        system->get_grid()->update_old_variables();

        delta_x = system->get_system_defect(
            x_i
        );

        r = correction.norm();
        iterations++;

        #ifdef PRINT_NEWTON_ITERATION_INFO
            #ifdef CONSOLE_OUTPUT
                std::cout << "      iteration: " << iterations
                          << " complete, residual: " << r
                          << ", inner iterations: " << solver->iterations
                          << ", inner residual: " << solver->residual << std::endl;
            #endif
        #endif

    } while(
        ( iterations <= max_iter ) &&
        (
            r > max_res ||
            ( iterations < system->get_grid()->get_params()->min_iterations_newton )
        )
    );

    return SolverParameters{
        iterations,
        r
    };
}
```

Listing A.1: Newton's method as implemented in MATRICS. The function `prepare_newton_method()` and `prepare_newton_iteration_step()` communicate to the employed instance of the `Solver`-class to prepare operation. In matrix-based operation, this includes computation of the matrix. The function `get_system_defect()` evaluates Equation (2.42) under consideration of the time integration scheme and boundary conditions.

A.4.2 Implementation of GMRES

```
Eigen::VectorXd GMRESMatFree::solve_for_rhs(
    Eigen::VectorXd _rhs,
    Eigen::VectorXd testing
){
    bool do_restart = false;
    bool stop = false;
    double error = 10000.0;

    prepare_internal_iteration_step();

    x_0.setZero();
    x = x_0;

    p_0 = soe->get_system_residual( _rhs, x );
    r_0 = preconditioner->solve_for_rhs( p_0, x );

    double r_norm = r_0.norm();

    if( r_norm < max_res )
    {
        iterations = 0;
        residual = r_norm;

        return x_0;
    }

    v_ji.col(0) = r_0  / r_norm;

    b(0) = r_norm;

    int iteration = 0;
    int j, i;

    for( j = 0; j < max_iter - 1; ++j )
    {
        ++iteration;

        p = soe->get_jacobi_vector_product( v_ji.col(j) );
        omega = preconditioner->solve_for_rhs( p, p );

        for( i = 0; i <= j; i++ )
        {
            h(i,j) = v_ji.col(i).transpose().dot( omega ) ;
            omega -= h(i,j) * v_ji.col(i);
        }

        h(j+1,j) = omega.norm();

        for( i = 0; i < j; i++ )
        {
            double _temp = c(i+1) * h(i,j) + s(i+1) * h(i+1,j);

            h(i+1,j) = -s(i+1) * h(i,j) + c(i+1) * h(i+1,j);
            h(i,j) = _temp;
        }

        beta = sqrt(  h(j,j)* h(j,j) + h(j+1,j)*h(j+1,j) );

        s(j+1) = h(j+1,j) / beta;
        c(j+1) = h(j,j) / beta;

        h(j,j) = beta;

        b(j+1) = -s(j+1) * b(j);
        b(j) = c(j+1) * b(j);

        error = abs( b(j+1) );

        stop = error <= max_res || iteration == max_iter-1;

        do_restart = bool( ( (j+1) % restart_at ) == 0) ;

        if( stop || do_restart  )
        {
            for( i = j; i >= 0; i-- )
            {
                double _temp = 0.0;

                for(int k=i+1; k<=j; k++){
                    _temp += h(i,k) * y(k);
                }

                y(i) = 1.0 / h(i,i) * ( b(i) - _temp );
            }

            temp.setZero();
```

```
            for( i = 0; i <= j; i++ )
            {
                temp += y(i) * v_ji.col(i);
            }

            x += temp;

            if( do_restart && !stop )
            {
                p_0 = soe->get_system_residual( _rhs, x );
                r_0 = preconditioner->solve_for_rhs( p_0, x );

                j = -1;

                h.setZero();
                c.setZero();
                s.setZero();
                v_ji.setZero();
                b.setZero();

                r_norm = r_0.norm();
                b(0) = r_norm;

                v_ji.col(0) = r_0  / r_norm;

                continue;
            }

            break;

        } else {
            v_ji.col(j+1) = omega / h(j+1,j);
        }
    }

    iterations = iteration;
    residual = error;

    return x;
}
```

Listing A.2: GMRES solver function as implemented in the MATRICS code. It is the main function of the `GMRESMatFree` class. The `GMRESMatFree`-class itself is derived from a virtual base class `Solver` such that the solver can be easily exchanged. The function `prepare_internal_iteration_step()` resets all variables, vectors, and matrices used to zero. `soe->get_system_resdidual( _rhs, x)` computes the outcome of Equation (2.40) for the system of equations that shall be solved, a given right-hand side (`_rhs`) and guess `x`. The function `preconditioner->solve_for_rhs(p, p)` evaluates the outcome of $P^{-1}p$, where P is the precondition. `soe->get_jacobi_vector_product( x )` evaluates the outcome of Equation (2.41).

A.4.3 Setup of Matrix in Matrix-Based Operation Mode

This section lists the most important functions for a matrix-based operation mode. They are given for the purpose of illustrating the usefulness of a matrix-free operation mode since, in this case, the functions below can be omitted completely.

```
void SOE::set_full_matrix(
){
    size_t j, k, i;

    P.setZero();
    triplets.clear();

    #pragma omp parallel for collapse( 2 ) private( j, k )
    for( j = 0; j < nrc_j; j++ )
    {
        for( k = 0; k < nrc_k; k++ )
        {
            calculate_diags_from_equations( j, k );
        }
    }

    for( i = 0; i < subsystem_count; i++ )
    {
        int j_start = 0;
        int k_start = 0;

        int k_end = nrc_k;
        int j_end = nrc_j;

        #ifdef SANITIZE_RADIAL_MOMENTUM_RADIAL
            // Do this only if current sub-domain is at jmin-boundary at grid
            if( j_grid_start == 0 ){

                if( i == equation[MOMENTUM] ){
                    j_start = 1;

                    #pragma omp parallel for private(k) reduction(merge: triplets)
                    for( k=0; k<nrc_k; k++ ){
                        triplets.push_back(
                            Eigen::Triplet<double>(
                                i+nrc*k,
                                i+nrc*k,
                                1.0
                            )
                        );
                    }
                }
            }
        #endif

        #ifdef SANITIZE_POLAR_MOMENTUM_POLAR

            if( k_grid_start == 0 ){

                if( i == equation[POLAR_MOMENTUM] ){
                    k_start = 1;

                    #pragma omp parallel for private(j) reduction(merge: triplets)
                    for( j = 0; j < nrc_j; j++ )
                    {
                        triplets.push_back(
                            Eigen::Triplet<double>(
                                i + subsystem_count*j,
                                i + subsystem_count*j,
                                1.0
                            )
                        );
                    }
                }

            }
        #endif

        #pragma omp parallel for collapse(2) private(j, k) reduction(merge: triplets)
        for( j = j_start; j < j_end; j++ )
        {
            for( k = k_start; k < k_end; k++ )
            {

                std::vector< Eigen::Triplet<double> > vec = set_diagonals( i, j, k, 0, POINT );

                triplets.insert(
                    triplets.end(),
                    vec.begin(),
                    vec.end()
                );
```

```
                #ifdef SOLVE_RADIAL_DIRECTION

                    if( j < nrc_j - 1 ){
                        vec = set_diagonals(
                            i, j, k, subsystem_count, EAST
                        );

                        triplets.insert(
                            triplets.end(),
                            vec.begin(),
                            vec.end()
                        );
                    }

                    if( j > 0 ){
                        vec = set_diagonals(
                            i, j, k, -subsystem_count, WEST
                        );

                        triplets.insert(
                            triplets.end(),
                            vec.begin(),
                            vec.end()
                        );
                    }

                #endif

                #ifdef SOLVE_POLAR_DIRECTION

                    if( k < nrc_k - 1 ){
                        vec = set_diagonals(
                            i, j, k, subsystem_count*nrc_j, NORTH
                        );

                        triplets.insert(
                            triplets.end(),
                            vec.begin(),
                            vec.end()
                        );
                    }

                    if( k > 0 ){
                        vec = set_diagonals(
                            i, j, k, -subsystem_count*nrc_j, SOUTH
                        );

                        triplets.insert(
                            triplets.end(),
                            vec.begin(),
                            vec.end()
                        );
                    }
                #endif
            }
        }

    }

    P.setFromTriplets( triplets.begin(), triplets.end() );
    P.makeCompressed();
}
```

Listing A.3: Calculates Jacobian-matrix explicitly and approximately with geometric 3 point stencil. This is only needed when a matrix-based method is used as a preconditioner or solver. It computes matrix elements approximately and stores them in the `eigen` sparse matrix format. 'SANATIZE_...' intercepts account for boundaries in the matrix as described in Section 2.3.4. The function `set_diagonals(...)` manages the matrix structure and is given in a listing below. The matrix elements are computed within the instances of `Equations`. An example for the continuity equation is given below as well.

```
inline std::vector<Eigen::Triplet<double>> SOE::set_diagonals(
    size_t main_eqn_index,
    size_t j,
    size_t k,
    int offset,
    short direction
){
    std::vector<Eigen::Triplet<double>> diagonals;

    size_t cell_position = j + nrc_j * k;

    size_t soe_row = subsystem_count * cell_position + main_eqn_index;
    size_t soe_column;

    for( auto const& [sub_eqn_name, sub_eqn_index] : equation )
    {
        if( sub_eqn_index != UNDEFINED ){

            soe_column = get_column( soe_row, main_eqn_index, sub_eqn_index, offset );

            diagonals.push_back(
                Eigen::Triplet<double>(
                    soe_row,
                    soe_column,
                    soes.at(main_eqn_index)->get_equations()->at(cell_position)->get_diags(
                        sub_eqn_name, direction
                    )
                )
            );
        }
    }

    return diagonals;
}
```

Listing A.4: Function `set_diagonals(...)` defined as a method of the `SOE` (System of Equations class) manages the matrix structure taking into account the equations solved, as well as the portion of the grid domain that is solved.

```
void ContinuityEquation::set_continuity_diags(
){
    jacobi.dL_drho = NULL_DIAG;

    OppositeInterfaces conditions;
    OppositeInterfaces values;
    UpwindRegion diags;

    #ifdef SOLVE_RADIAL_DIRECTION

        conditions.at_min = cell.p->get_vrrmin();
        conditions.at_max = cell.e->get_vrrmin();

        values.at_min = grid->div_r( cell.e, cell.p, 0.0, conditions.at_min );
        values.at_max = grid->div_r( cell.e, cell.p, conditions.at_max, 0.0 );

        diags = coupling_scheme->get_main_diag_vectors( values, conditions );

        jacobi.dL_drho.w += diags.w;
        jacobi.dL_drho.p += diags.p;
        jacobi.dL_drho.e += diags.e;

    #endif

    #ifdef SOLVE_POLAR_DIRECTION

        conditions.at_min = cell.p->get_vttmin();
        conditions.at_max = cell.n->get_vttmin();

        values.at_min = grid->div_theta( cell.n, cell.p, 0.0, conditions.at_min );
        values.at_max = grid->div_theta( cell.n, cell.p, conditions.at_max, 0.0 );

        diags = coupling_scheme->get_main_diag_vectors( values, conditions );

        jacobi.dL_drho.s += diags.w;
        jacobi.dL_drho.p += diags.p;
        jacobi.dL_drho.n += diags.e;

    #endif

    jacobi.dL_drho *= dt;

    #ifdef SOLVE_SECOND_ORDER_TIME
        #ifdef CRANK_NICHELSON
            jacobi.dL_drho *= 0.5;
```

```
            jacobi.dL_drho.p += 1.0;
        #else
            jacobi.dL_drho.p += 1.5;
        #endif
    #else
        jacobi.dL_drho.p += 1.0;
    #endif
}
```

Listing A.5: Computes elements of the Jacobian matrix that are derivatives of the discritized continuity equation by density entries. For every equation, functions like this must be found for each conserved variable (5 Equations result in 25 functions comparable with the hereby shown). Matrix elements are stored in the `jacobi`-struct. In this example, elements `jacobi.dL_drho.p` are associated with the matrix's main diagonal.

A.4.4 Matrix-Free Implementation in MATRICS

```
void SOE::set_generic_jacobian(
){
    int i, j, k, idx;

    _mu_stored = _mu;
    F2 = get_lhs( _mu );
}

Eigen::VectorXd SOE::get_jacobi_vector_product(
    Eigen::VectorXd x
){
    int idx, i, j, k;

    double euclid_norm_x = x.norm();

    if( euclid_norm_x == 0.0 ){
        J_x_real.setZero();
        return J_x_real;
    }

    double epsilon = sqrt( std::numeric_limits<double>::epsilon() ) / euclid_norm_x;

    q_1 = _mu_stored + epsilon * x;

    F1 = get_lhs( q_1 );

    J_x_real =  ( F1 - F2 ) / ( epsilon );

    #ifdef SANITIZE_RADIAL_MOMENTUM_RADIAL
        if( equation[MOMENTUM] != UNDEFINED )
        {
            if( j_grid_start == 0 ){

                #pragma omp parallel for private( k )
                for( k=0; k<nrc_k; k++ ){
                    idx = equation[MOMENTUM] + nrc * k;
                    J_x_real( idx ) = x( idx );
                }
            }
        }
    #endif

    #ifdef SANITIZE_POLAR_MOMENTUM_POLAR
        if( equation[POLAR_MOMENTUM] != UNDEFINED )
        {
            if( k_grid_start == 0 ){

                #pragma omp parallel for private( j )
                for( j=0; j<nrc_j; j++ ){

                    idx = equation[POLAR_MOMENTUM] + subsystem_count * j;
                    J_x_real( idx ) = x( idx );
                }
            }
        }
    #endif

    return J_x_real;
}

Eigen::VectorXd SOE::get_lhs(
```

```
    Eigen::VectorXd x
){
    int j, k, i;
    int idx;

    update_variables( x );

    for( i=0; i<subsystem_count; i++ )
    {
        #pragma omp parallel for collapse( 2 ) private( j, k, idx )
        for( j=0; j<nrc_j; j++ )
        {
            for( k=0; k<nrc_k; k++ )
            {
                idx = j*subsystem_count+i+nrc*k;
                lhs_flux( idx ) = soes.at(i)->get_equations()->at( j + nrc_j * k )->get_lhs();
            }
        }
    }

    #ifdef SANITIZE_RADIAL_MOMENTUM_RADIAL
        if( equation[MOMENTUM] != UNDEFINED )
        {
            #pragma omp parallel for private( k )
            for( k=0; k<nrc_k; k++ ){
                idx = equation[MOMENTUM] + nrc * k;
                lhs_flux( idx ) = 0.0;
            }
        }
    #endif

    #ifdef SANITIZE_POLAR_MOMENTUM_POLAR
        if( equation[POLAR_MOMENTUM] != UNDEFINED )
        {
            #pragma omp parallel for private( j )
            for( j=0; j<nrc_j; j++ )
            {
                idx = equation[POLAR_MOMENTUM] + subsystem_count * j;
                lhs_flux( idx ) = 0.0;
            }
        }
    #endif

    #ifdef SOLVE_SECOND_ORDER_TIME
        #ifdef CRANK_NICHELSON
            return x + ( 1. - grid->get_params()->alpha_cn ) * lhs_flux * grid->get_params()->dt;
        #else
            if( !grid->get_params()->first_timestep )
            {
                return 1.5 * x + lhs_flux * grid->get_params()->dt;
            }
        #endif
    #endif

    return x + lhs_flux * grid->get_params()->dt;
}
```

Listing A.6: The three functions constituting the matrix-free operation mode. The function `soes.at(i)>get_equations()>at(j+nrc_jk)>get_lhs()` evaluates the respective equation (continuity, momentum, etc.) for the respective position in the grid (`j+nrc_j·k`). This represents every piece of physics within that equation. The functions in this listing replace all functions shown in Section A.4.3, and an additional $\sim$ 1000 lines of code evaluating matrix elements. Furthermore, the functions in Section A.4.3 only show the matrix evaluation, not the matrix-vector product computation. That part is handled by the `eigen` library in the matrix-based approach and fully described below in the matrix-free approach. 'SANATIZE_...' intercepts account for boundaries in the matrix as described in Section 2.3.4.

B Appendix to GSF Modeling

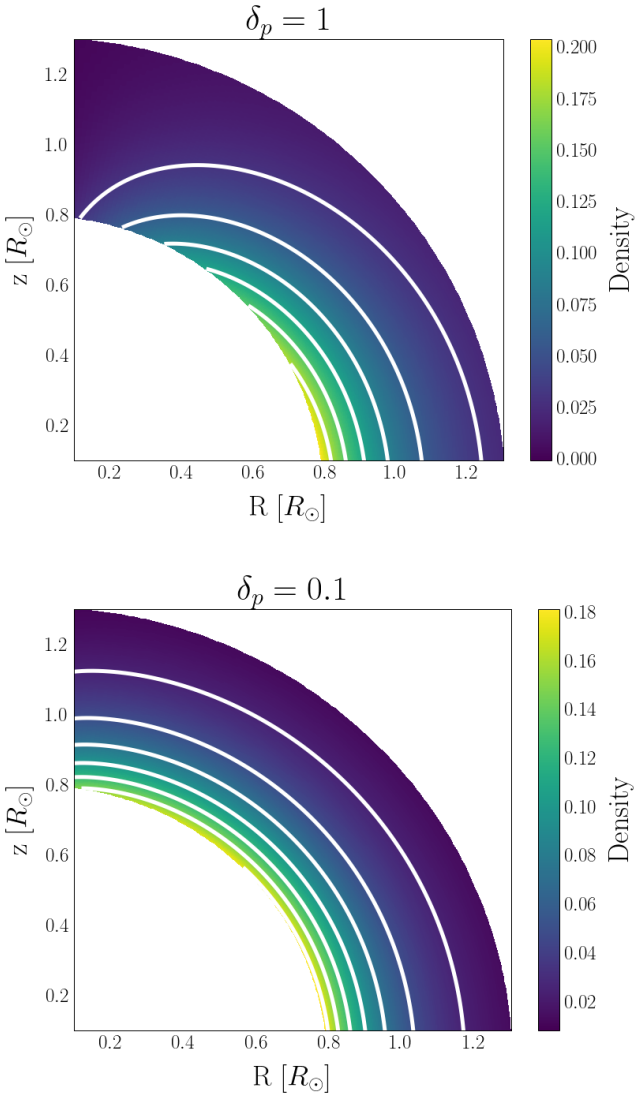


Figure B.1: Density map and corresponding iso-lines for different strengths of rotation support in the stellar toy model described in Chapter 3. The density contours are a measure of the modeled object's oblateness. The rotational support in the top picture is a factor of ten than in the bottom. This directly translates to the density iso-lines.

B.1 Morphologies at Different Inclinations for Ideal Gas

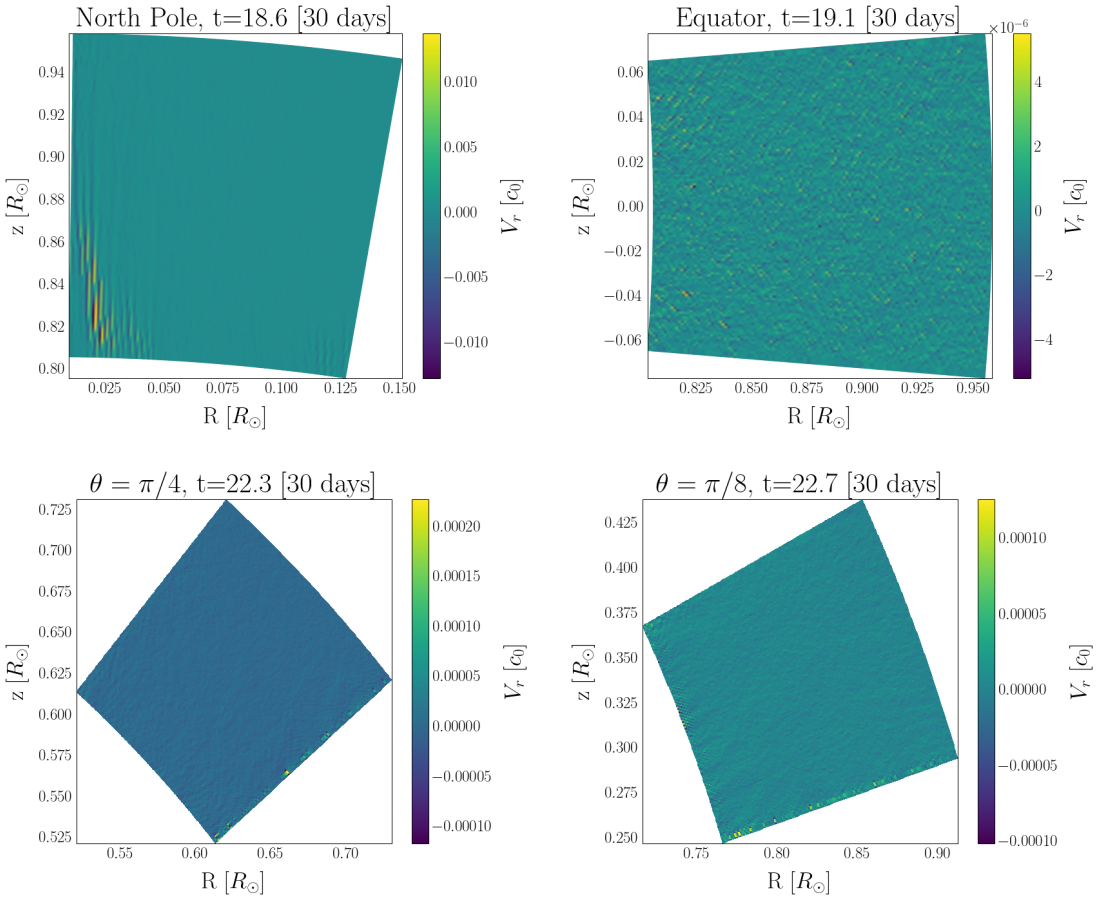


Figure B.2: Radial velocity component obtained through time integration under ideal gas conditions of slightly perturbed initial conditions. The depicted boxes are located at the north pole (top, left) and the equatorial plane (top, right), and inclinations $\theta = \pi/4$ (bottom left) and $\theta = \pi/8$ (bottom, right). The initial perturbation at the equatorial plane (top-right) was set (accidentally) a factor of ten lower than in the other images. This is not expected to make any difference beyond this very factor in the observed value for the velocity component.

B.2 Isothermal GSF Growth at Different Inclinations

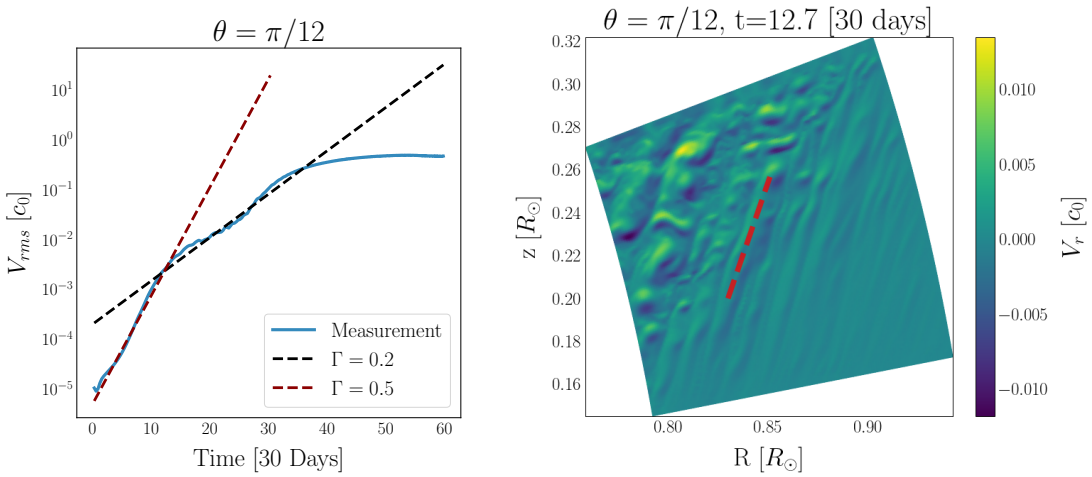


Figure B.3: Isothermal growth rate (left) and spherically radial velocity component in the domain (right) at inclination $\theta = \pi/12$. The right plot is created at the end of the initial rapid growth time. The red line in the right plot shows a visual fit of the angle ξ, which is found at $\xi \approx 70°$.

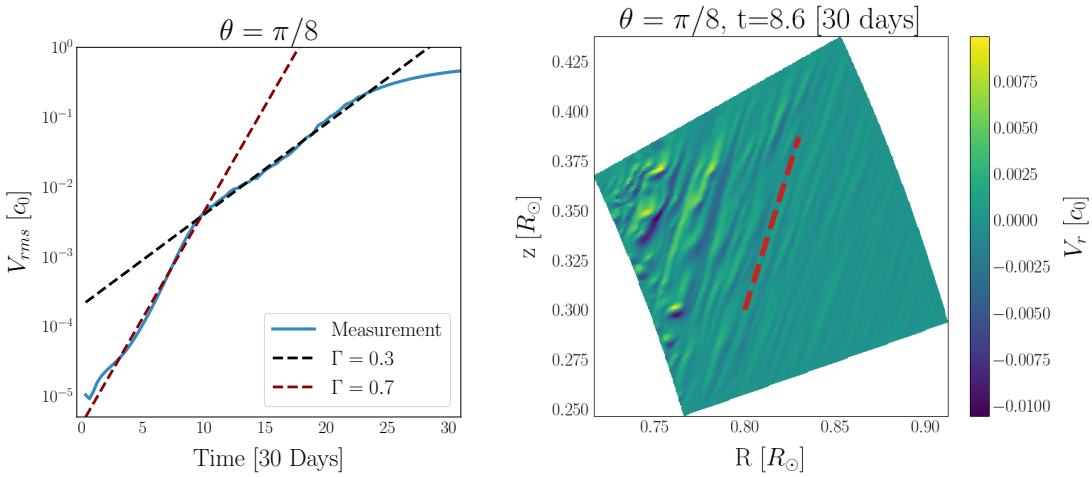


Figure B.4: Isothermal growth rate at inclination $\theta = \pi/8$ (left) and spherically radial velocity component in the domain (right). The right plot is created at the end of the initial rapid growth time. The red line in the right plot shows a visual fit of the angle ξ, which is found at $\xi \approx 71°$.

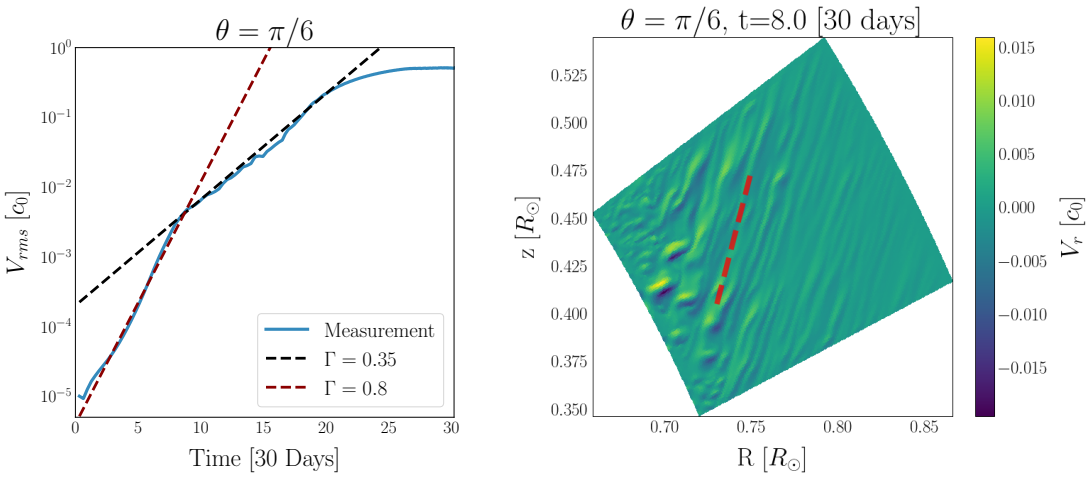


Figure B.5: Isothermal growth rate at inclination $\theta = \pi/6$ (left) and spherically radial velocity component in the domain (right). The right plot is created at the end of the initial rapid growth time. The red line in the right plot shows a visual fit of the angle ξ, which is found at $\xi \approx 74°$.

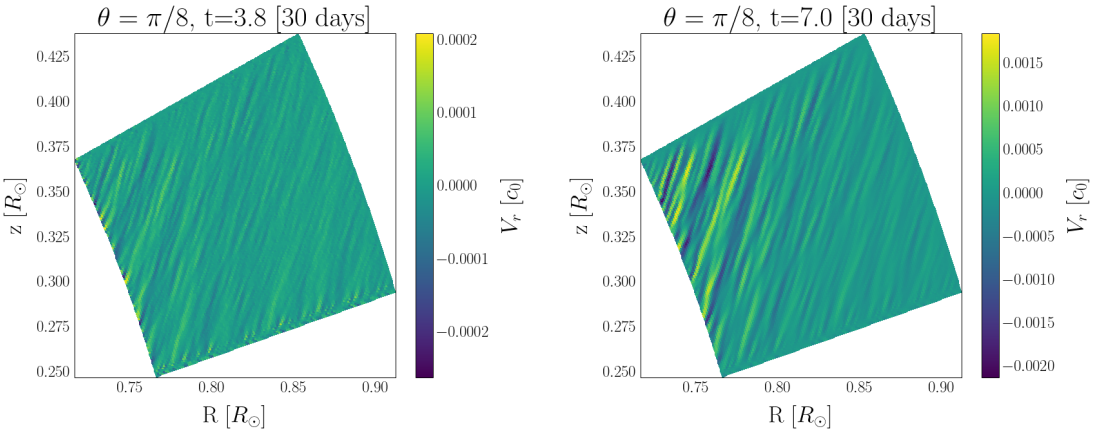


Figure B.6: GSF instability simulated at $\theta = \pi/8$ using an isothermal equation of state at two different time steps during the initial linear growth phase.

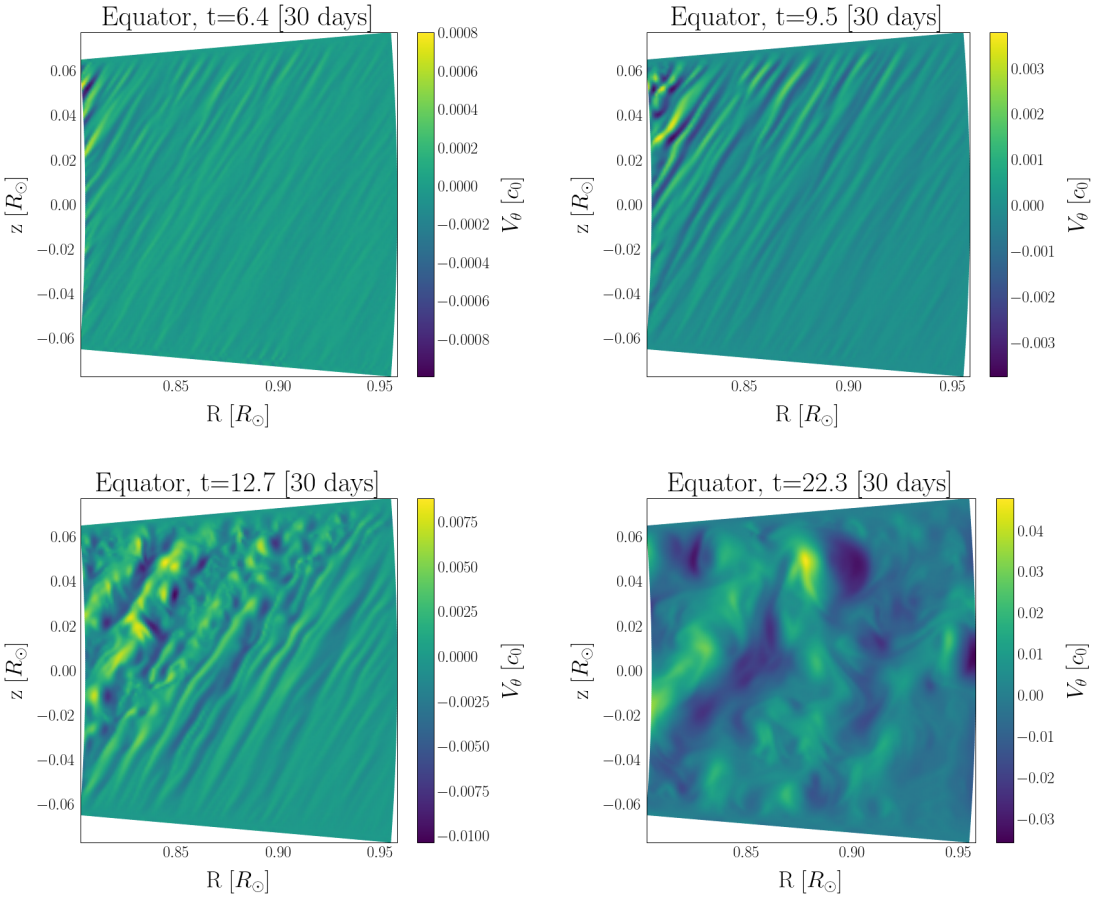


Figure B.7: GSF instability simulated at $\theta = \pi/12$ using an isothermal equation of state at four different time steps during the initial linear growth phase (top) and non-linear growth phase (bottom).

B.3 Density Variations in Thermalized Runs

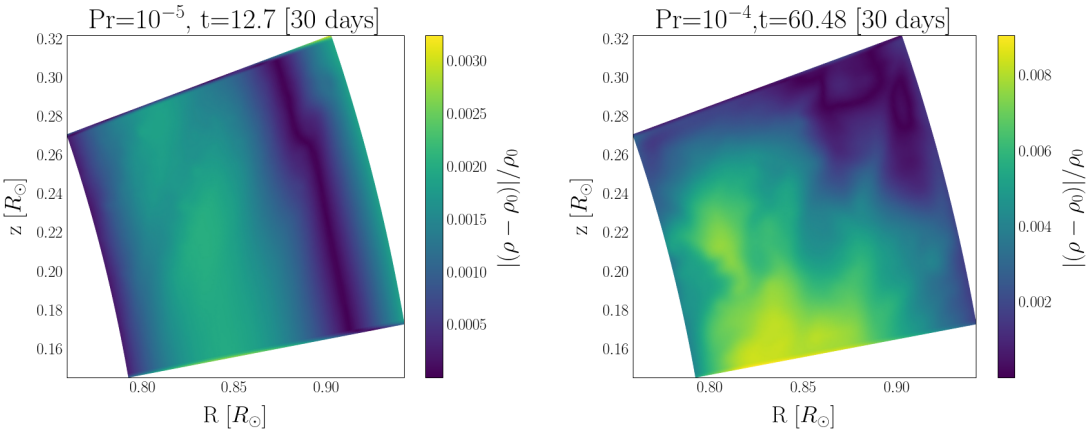


Figure B.8: Density variations relative to background density for the diffusive-thermally relaxed runs. Depicted is the simulation with Prandtl number $Pr = 10^{-5}$ on the left and the one with $Pr = 10^{-4}$ on the right. The relative variation in density is $\lesssim 8‰$for both simulation runs.

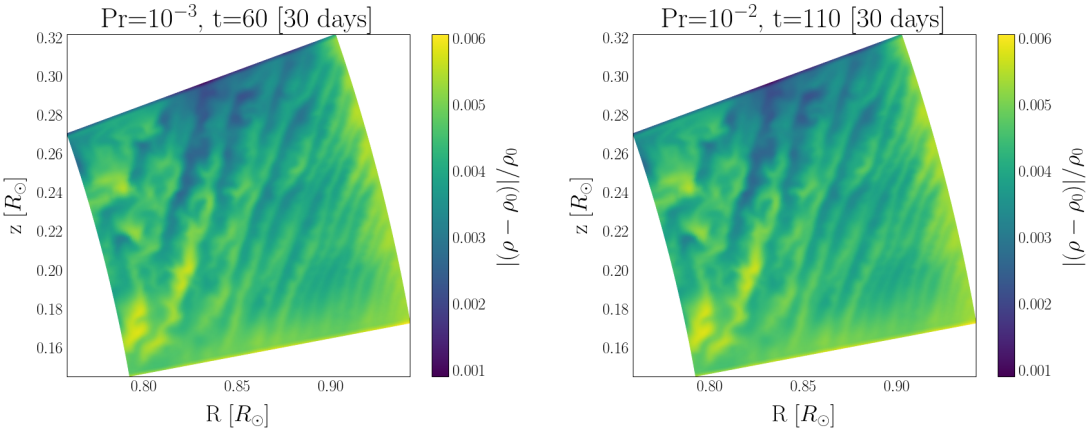


Figure B.9: Density variations relative to background density for the diffusive-thermally relaxed runs. Depicted is the simulation with Prandtl number $Pr = 10^{-3}$ on the left and the one with $Pr = 10^{-2}$ on the right. The relative variation in density is $\lesssim 6‰$for both simulation runs.

C Appendix to Pebble Cloud Collapse Analysis

C.1 Object Identification

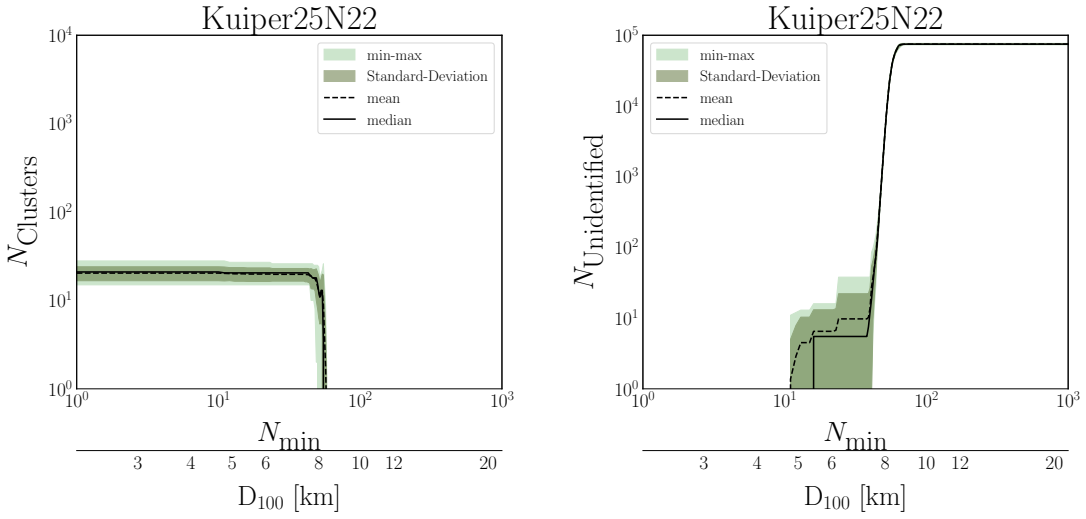


Figure C.1: The count of identified objects (here clusters) over the respective value of N_{min} used in the identification step (left) and unassigned but solid particles over minimal object size (right). N_{min} is actually $N_{min} + 1$ to exclude the single particle object case. The plots are generated for the set `Kuiper25N22` using all eleven runs within.

C.2 Size Distribution of Planetesimals

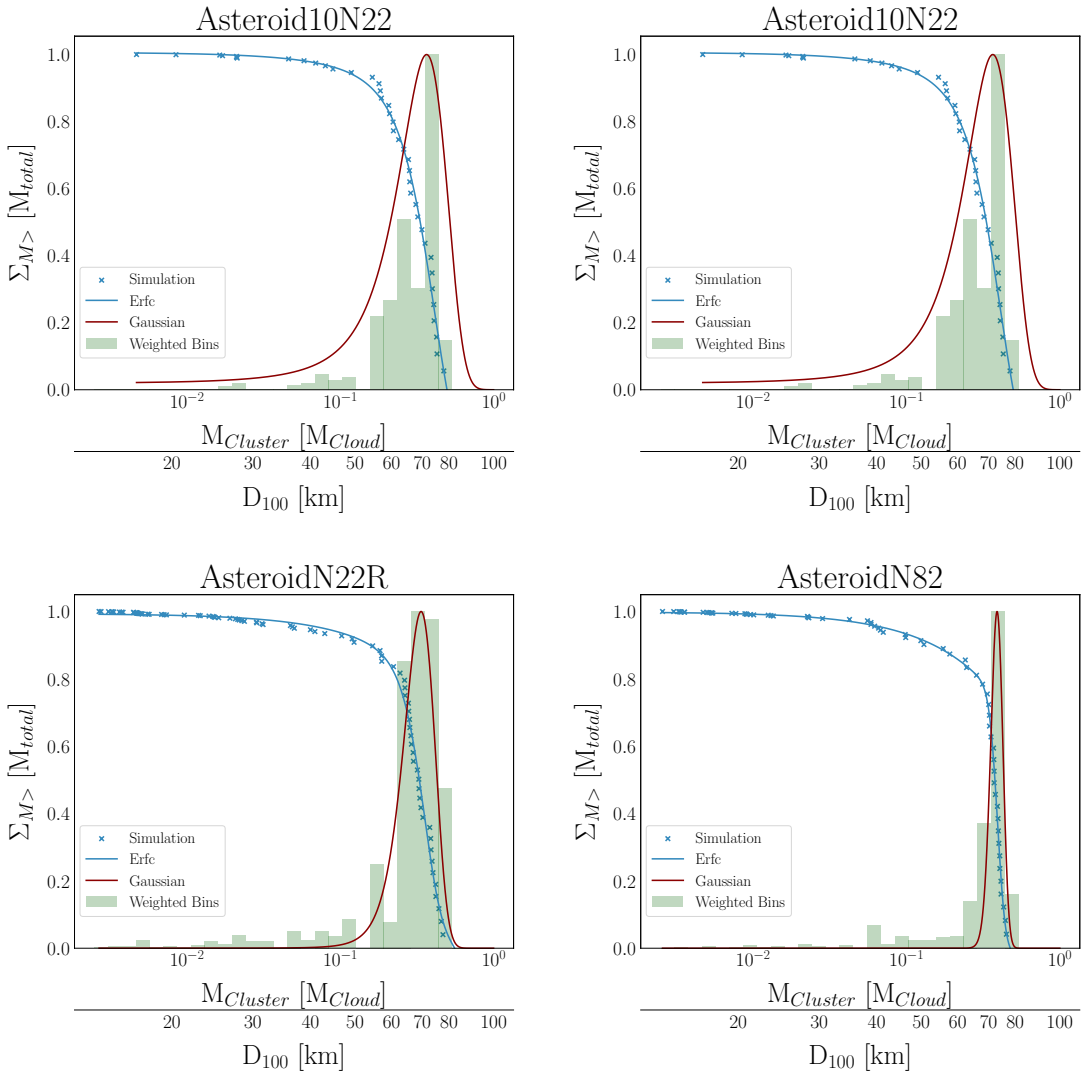


Figure C.2: Cumulative mass distributions for all runs. Data points and Fit function, as well as the Gaussian function and histogram, are normalized. The Fit function is given by Equation 4.18, the Gaussian is computed using M_{50} and M_{84} from the fitting routine according to Equation 4.17.

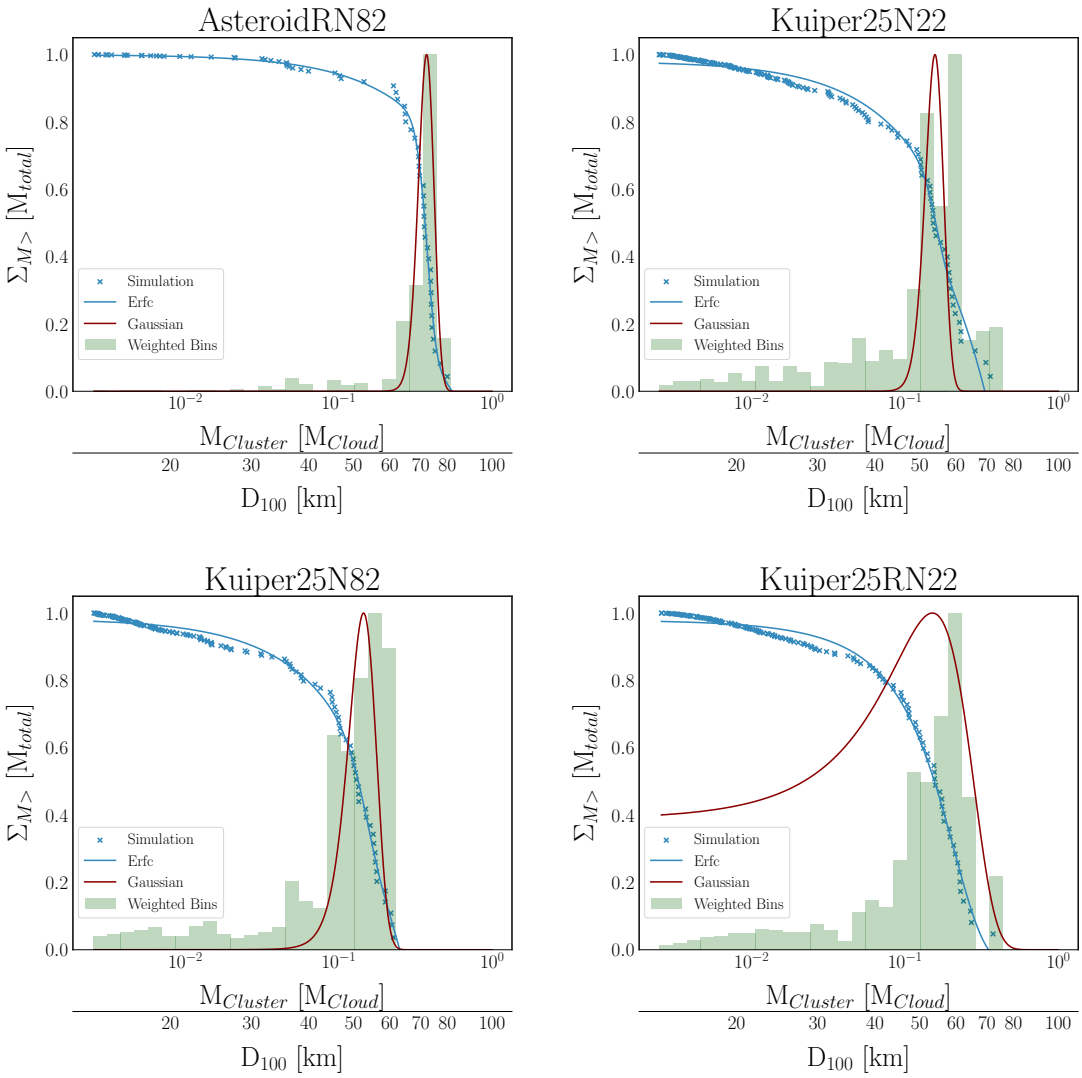


Figure C.3: Cumulative mass distributions for all runs. Data points and Fit function, as well as the Gaussian function and histogram, are normalized. The Fit function is given by Equation 4.18, the Gaussian is computed using M_{50} and M_{84} from the fitting routine according to Equation 4.17. The fits for the `Kuiper` runs diverge towards the low mass end.

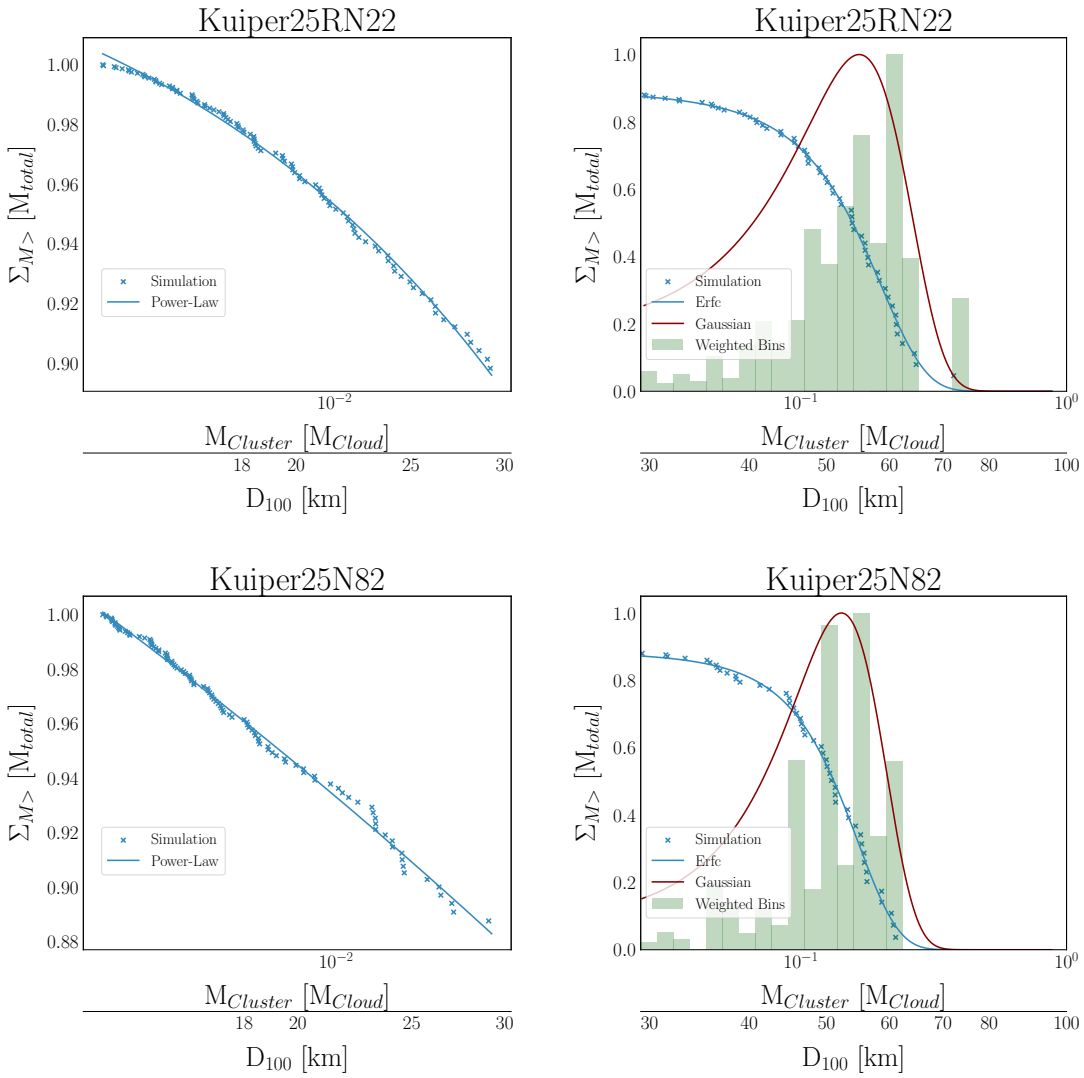


Figure C.4: Cumulative mass distribution for the `Kuiper25RN22` (top) and the `Kuiper25N82` (bottom) simulation runs. The fit at masses $0.0025 M_{Cloud} < M < 0.05 M_{Cloud}$ is done with a power law towards (left). For masses $M > 0.05 M_{Cloud}$ the erfc (not using a linear tail extension) is fitted (right).

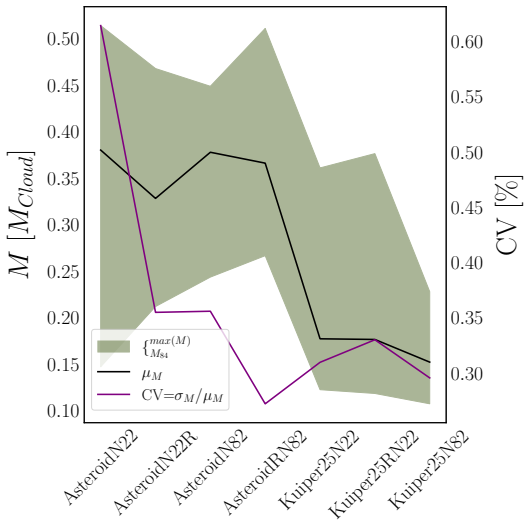


Figure C.5: Object masses for simulation runs. Values of M_{50} are read from CCDF. The filled area ranges from M_{84} to the maximum object mass in the respective simulation set (corresponding to the left axis). The purple line is the Coefficient of Variation (CV) for the respective data points (right axis).

C.3 Merging Activity

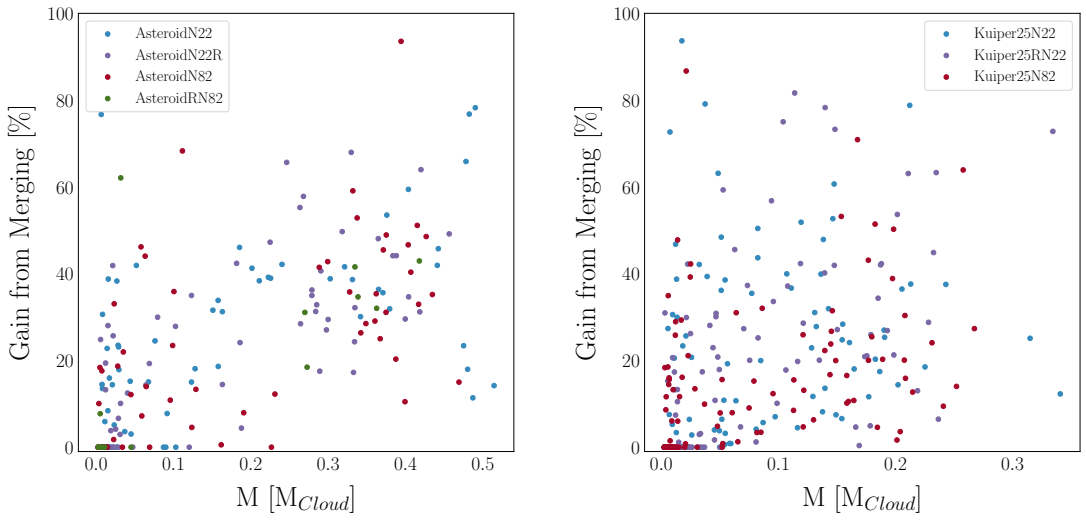


Figure C.6: Fraction of an object's mass gain in relation to the mass of the respective object for the `Asteroid` runs (left) and the `Kuiper` runs (right). For most of the `Asteroid` runs, a strong positive linear correlation can be inferred; this value is lower for the `Kuiper` runs (Table 4.4).

D Lists of Figures and Tables

D.1 List of Figures

D.2 List of Tables

E List of Publications

- J. Meyer, J. D. Melon-Fuksman, and H. Klahr. Matrics: The implicit matrix-free eulerian hydrodynamics solver. Astronomy & Astrophysics doi: 10.1051/0004-6361/202348746, in production
- J. Meyer, J. Tjoa, and H. Klahr. Formation of asteroids and Kuiper-belt objects via pebble cloud collapse I: Mass distribution and binary systems, in preparation
- J. Meyer and H. Klahr. Simulations of asteroids and Kuiper-belt objects II: Merging history, in preparation

F Additional Tools Used in This Thesis

- The MATRICS code is developed in `C++-17`, using VS Code. The packages used and not contained in the standard library are
  - `eigen` (Guennebaud et al. [2010])
  - `OpenMP` (Chandra et al. [2001])
  - `HDF5` (The HDF Group)
- Every analysis in this thesis is done using `Python3.10` (Van Rossum and Drake Jr [1995]) the utilized non-native packages are
  - `numpy` (Harris et al. [2020])
  - `scipy` (Virtanen et al. [2020])
  - `Matplotlib` (Hunter [2007])
  - `pandas` (McKinney et al. [2010])
  - `seaborn` (Waskom [2021])
- MATRICS development and all simulations in Chapter 2 are conducted using a `Lenovo Legion 5 15ARH05H` Laptop
- Simulation in Chapter 3 and 4 are conducted on the MPIA-`VERA`-cluster
- This thesis is written in Overleaf
- Spell-checking in this thesis is done using Grammarly

G Bibliography

F. C. Adams and R. Watkins. Vortices in Circumstellar Disks. , 451:314, September 1995. doi: 10.1086/176221.

F. C. Adams, C. J. Lada, and F. H. Shu. Spectral Evolution of Young Stellar Objects. , 312:788, January 1987. doi: 10.1086/164924.

C. Aerts, S. Mathis, and T. M. Rogers. Angular Momentum Transport in Stellar Interiors. , 57:35–78, August 2019. doi: 10.1146/annurev-astro-091918-104359.

C. D. Andereck, S. S. Liu, and H. L. Swinney. Flow regimes in a circular couette system with independently rotating cylinders. *Journal of Fluid Mechanics*, 164: 155–183, March 1986. doi: 10.1017/s0022112086002513. URL https://doi.org/10.1017/s0022112086002513.

E. Anders. Fragmentation history of asteroids. *Icarus*, 4(4):399–408, September 1965. ISSN 0019-1035. doi: 10.1016/0019-1035(65)90044-8. URL http://dx.doi.org/10.1016/0019-1035(65)90044-8.

P. J. Armitage. *Astrophysics of Planet Formation*. Cambridge University Press, January 2020. ISBN 9781108420501. doi: 10.1017/9781108344227. URL http://dx.doi.org/10.1017/9781108344227.

P. J. Armitage and W. Kley. *From Protoplanetary Disks to Planet Formation: Saas-Fee Advanced Course 45. Swiss Society for Astrophysics and Astronomy*. Springer Berlin Heidelberg, 2019. ISBN 9783662586877. doi: 10.1007/978-3-662-58687-7. URL http://dx.doi.org/10.1007/978-3-662-58687-7.

W. Auzinger. Defect correction for nonlinear elliptic difference equations. *Numerische Mathematik*, 51(2):199–208, March 1987. doi: 10.1007/bf01396749. URL https://doi.org/10.1007/bf01396749.

S. A. Balbus and J. F. Hawley. A Powerful Local Shear Instability in Weakly Magnetized Disks. I. Linear Analysis. , 376:214, July 1991. doi: 10.1086/170270.

D. S. Balsara, S. Garain, V. Florinski, and W. Boscheri. An efficient class of weno schemes with adaptive order for unstructured meshes. *Journal of Computational Physics*, 404:109062, 2020. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2019.109062. URL https://www.sciencedirect.com/science/article/pii/S0021999119307673.

A. J. Barker, C. A. Jones, and S. M. Tobias. Angular momentum transport by the GSF instability: non-linear simulations at the equator. *Monthly Notices of the Royal Astronomical Society*, 487(2):1777–1794, 05 2019. ISSN 0035-8711. doi: 10.1093/mnras/stz1386. URL https://doi.org/10.1093/mnras/stz1386.

A. J. Barker, C. A. Jones, and S. M. Tobias. Angular momentum transport, layering, and zonal jet formation by the GSF instability: non-linear simulations at a general latitude. *Monthly Notices of the Royal Astronomical Society*, 495(1):1468–1490, 05 2020. ISSN 0035-8711. doi: 10.1093/mnras/staa1327. URL https://doi.org/10.1093/mnras/staa1327.

R. Barrett, M. Berry, T. F. Chan, J. Demmel, et al. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Society for Industrial and Applied Mathematics, January 1994. doi: 10.1137/1.9781611971538. URL https://doi.org/10.1137/1.9781611971538.

P. Bastian, E. Hermann Müller, S. Müthing, and M. Piatkowski. Matrix-free multigrid block-preconditioners for higher order discontinuous galerkin discretisations. *Journal of Computational Physics*, 394:417–439, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2019.06.001. URL https://www.sciencedirect.com/science/article/pii/S0021999119303973.

K. R. Bell, P. M. Cassen, H. H. Klahr, and T. Henning. The Structure and Appearance of Protostellar Accretion Disks: Limits on Disk Flaring. , 486(1):372–387, September 1997. doi: 10.1086/304514.

M. Benisty, C. Dominik, K. Follette, A. Garufi, C. Ginski, J. Hashimoto, M. Keppler, W. Kley, and J. Monnier. Optical and Near-infrared View of Planet-forming Disks and Protoplanets. In S. Inutsuka, Y. Aikawa, T. Muto, K. Tomida, and M. Tamura, editors, *Protostars and Planets VII*, volume 534 of *Astronomical Society of the Pacific Conference Series*, page 605, July 2023. doi: 10.48550/arXiv.2203.09991.

R. B. Bird, W. E. Stewart, and E. N. Lightfoot. *Transport Phenomena*. John Wiley & Sons, Chichester, England, November 2006.

T. Birnstiel, C. W. Ormel, and C. P. Dullemond. Dust size distributions in coagulation/fragmentation equilibrium: numerical solutions and analytical fits. , 525: A11, January 2011. doi: 10.1051/0004-6361/201015228.

T. Birnstiel, M. Fang, and A. Johansen. Dust evolution and the formation of planetesimals. *Space Science Reviews*, 205(1–4):41–75, May 2016. ISSN 1572-9672. doi: 10.1007/s11214-016-0256-1. URL http://dx.doi.org/10.1007/s11214-016-0256-1.

J. Blum. Dust evolution in protoplanetary discs and the formation of planetesimals: What have we learned from laboratory experiments? *Space Science Reviews*,

214(2), February 2018. ISSN 1572-9672. doi: 10.1007/s11214-018-0486-5. URL http://dx.doi.org/10.1007/s11214-018-0486-5.

P. Bodenheimer, G. P. Laughlin, M. Rozyczka, et al. *Numerical methods in astrophysics*. Series in Astronomy and Astrophysics. Institute of Physics Publishing, London, England, December 2006.

F. Brauer, C. P. Dullemond, and T. Henning. Coagulation, fragmentation and radial motion of solid particles in protoplanetary disks. *Astronomy amp; Astrophysics*, 480(3):859–877, November 2007. ISSN 1432-0746. doi: 10.1051/0004-6361:20077759. URL http://dx.doi.org/10.1051/0004-6361:20077759.

A. L. Brundage. Implementation of tillotson equation of state for hypervelocity impact of metals, geologic materials, and liquids. *Procedia Engineering*, 58:461–470, 2013. ISSN 1877-7058. doi: https://doi.org/10.1016/j.proeng.2013.05.053. URL https://www.sciencedirect.com/science/article/pii/S1877705813009594. Proceedings of the 12th Hypervelocity Impact Symposium.

A. Caleo and S. A. Balbus. The radiative zone of the sun and the tachocline: stability of baroclinic patterns of differential rotation. *Monthly Notices of the Royal Astronomical Society*, 457(2):1711–1721, February 2016. ISSN 1365-2966. doi: 10.1093/mnras/stw098. URL http://dx.doi.org/10.1093/mnras/stw098.

A. Caleo, S. A. Balbus, and E. Tognelli. The goldreich–schubert–fricke instability in stellar radiative zones. *Monthly Notices of the Royal Astronomical Society*, 460 (1):338–344, April 2016. ISSN 1365-2966. doi: 10.1093/mnras/stw1002. URL http://dx.doi.org/10.1093/mnras/stw1002.

M. Cantiello, C. Mankovich, L. Bildsten, J. Christensen-Dalsgaard, and B. Paxton. Angular momentum transport within evolved low-mass stars. *The Astrophysical Journal*, 788(1):93, May 2014. ISSN 1538-4357. doi: 10.1088/0004-637x/788/1/93. URL http://dx.doi.org/10.1088/0004-637X/788/1/93.

R. Chandra, L. Dagum, D. Kohr, et al. *Parallel programming in OpenMP*. Morgan kaufmann, 2001.

S Chandrasekhar. Hydrodynamic and hydromagnetic stability. 1 1961. URL https://www.osti.gov/biblio/5063967.

R. Choquet. A Matrix-Free Preconditioner Applied to CFD. Research Report RR-2605, INRIA, 1995. URL https://inria.hal.science/inria-00074080.

T. J. Chung. *Computational Fluid Dynamics*. Cambridge University Press, February 2002. ISBN 9780511606205. doi: 10.1017/cbo9780511606205. URL http://dx.doi.org/10.1017/CBO9780511606205.

M. Delbo, C. Avdellidou, and A. Morbidelli. Ancient and primordial collisional families as the main sources of x-type asteroids of the inner main belt. *Astronomy &amp Astrophysics*, 624:A69, April 2019. doi: 10.1051/0004-6361/201834745. URL https://doi.org/10.1051/0004-6361/201834745.

M. Delbo', K. Walsh, B. Bolin, C. Avdellidou, and A. Morbidelli. Identification of a primordial asteroid family constrains the original planetesimal population. *Science*, 357(6355):1026–1029, September 2017. ISSN 1095-9203. doi: 10.1126/science.aam6036. URL http://dx.doi.org/10.1126/science.aam6036.

H. Deng, C. Reinhardt, F. Benitez, L. Mayer, J. Stadel, and A. C. Barr. Enhanced mixing in giant impact simulations with a new lagrangian method. *The Astrophysical Journal*, 870(2):127, January 2019. doi: 10.3847/1538-4357/aaf399. URL https://doi.org/10.3847/1538-4357/aaf399.

E. A. Dorfi. Radiation hydrodynamics: Numerical aspects and applications. In *Computational Methods for Astrophysical Fluid Flow*, pages 263–341. Springer-Verlag, 1997. doi: 10.1007/3-540-31632-9_3. URL https://doi.org/10.1007/3-540-31632-9_3.

D. R. Durran. *Numerical methods for fluid dynamics*. Springer, New York, NY, September 2010.

R. W. Dymott, A. J. Barker, C. A. Jones, and S. M. Tobias. Linear and non-linear properties of the Goldreich–Schubert–Fricke instability in stellar interiors with arbitrary local radial and latitudinal differential rotation. *Monthly Notices of the Royal Astronomical Society*, 524(2):2857–2882, 07 2023. ISSN 0035-8711. doi: 10.1093/mnras/stad1982. URL https://doi.org/10.1093/mnras/stad1982.

A. S. Eddington. Internal circulation in rotating stars. , 90:54, November 1929. doi: 10.1093/mnras/90.1.54.

P. S. Epstein. On the resistance experienced by spheres in their motion through gases. *Phys. Rev.*, 23:710–733, Jun 1924. doi: 10.1103/PhysRev.23.710. URL https://link.aps.org/doi/10.1103/PhysRev.23.710.

A. Esser and S. Grossmann. Analytic expression for taylor–couette stability boundary. *Physics of Fluids*, 8(7):1814–1819, July 1996. doi: 10.1063/1.868963. URL https://doi.org/10.1063/1.868963.

M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer Berlin Heidelberg, 2002. doi: 10.1007/978-3-642-56026-2. URL https://doi.org/10.1007/978-3-642-56026-2.

F. Fraysse and R. Saurel. Automatic differentiation using operator overloading (ADOO) for implicit resolution of hyperbolic single phase and two-phase flow models. *Journal of Computational Physics*, 399:108942, December 2019. doi: 10.1016/j.jcp.2019.108942. URL https://doi.org/10.1016/j.jcp.2019.108942.

K. Fricke. Instabilität stationärer Rotation in Sternen. , 68:317, January 1968.

C. F. Gammie. Layered Accretion in T Tauri Disks. , 457:355, January 1996. doi: 10.1086/176735.

P. Goldreich and G. Schubert. Differential Rotation in Stars. , 150:571, November 1967. doi: 10.1086/149360.

P. Goldreich and W. R. Ward. The Formation of Planetesimals. , 183:1051–1062, August 1973. doi: 10.1086/152291.

T. P. Greene, B. A. Wilking, P. Andre, E. T. Young, and C. J. Lada. Further Mid-Infrared Study of the rho Ophiuchi Cloud Young Stellar Population: Luminosities and Masses of Pre–Main-Sequence Stars. , 434:614, October 1994. doi: 10.1086/174763.

E. Grüneisen. Theorie des festen Zustandes einatomiger Elemente. *Annalen der Physik*, 344(12):257–306, January 1912. doi: 10.1002/andp.19123441202.

G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

W. Hackbusch. *Multi-Grid Methods and Applications*. Springer Berlin Heidelberg, 1985. doi: 10.1007/978-3-662-02427-0. URL https://doi.org/10.1007/978-3-662-02427-0.

E. Hairer, G. Wanner, and S. P. Nørsett. *Solving Ordinary Differential Equations I*. Springer Berlin Heidelberg, 1993. doi: 10.1007/978-3-540-78862-1. URL https://doi.org/10.1007/978-3-540-78862-1.

F. H. Harlow and J. E. Welch. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *The Physics of Fluids*, 8(12):2182–2189, 12 1965. ISSN 0031-9171. doi: 10.1063/1.1761178. URL https://doi.org/10.1063/1.1761178.

C. R. Harris, K. J. Millman, S. J. van der Walt, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

L. Hartmann, N. Calvet, E. Gullbring, and P. D'Alessio. Accretion and the evolution of t tauri disks. *The Astrophysical Journal*, 495(1):385–400, March 1998. ISSN 1538-4357. doi: 10.1086/305277. URL http://dx.doi.org/10.1086/305277.

W. K. Hartmann and A. C. Hartmann. Asteroid collisions and evolution of asteroidal mass distribution and meteoritic flux. *Icarus*, 8(1–3):361–381, January 1968. ISSN 0019-1035. doi: 10.1016/0019-1035(68)90084-5. URL http://dx.doi.org/10.1016/0019-1035(68)90084-5.

C. Hayashi. Structure of the Solar Nebula, Growth and Decay of Magnetic Fields and Effects of Magnetic and Turbulent Viscosities on the Nebula. *Progress of Theoretical Physics Supplement*, 70:35–53, 01 1981. ISSN 0375-9687. doi: 10.1143/PTPS.70.35. URL https://doi.org/10.1143/PTPS.70.35.

A. Heger, N. Langer, and S. E. Woosley. Presupernova Evolution of Rotating Massive Stars. I. Numerical Method and Evolution of the Internal Stellar Structure. , 528(1):368–396, January 2000. doi: 10.1086/308158.

P. F. Hopkins. A new class of accurate, mesh-free hydrodynamic simulation methods. *Monthly Notices of the Royal Astronomical Society*, 450(1):53–110, April 2015. doi: 10.1093/mnras/stv195. URL https://doi.org/10.1093/mnras/stv195.

A. Hujeirat. A problem-orientable numerical algorithm for modeling multidimensional radiative mhd flows in astrophysics—the hierarchical solution scenario. *Computer Physics Communications*, 168(1):1–24, 2005. ISSN 0010-4655. doi: https://doi.org/10.1016/j.cpc.2005.01.013. URL https://www.sciencedirect.com/science/article/pii/S0010465505000640.

A. Hujeirat and R. Rannacher. A method for computing compressible, highly stratified flows in astrophysics based on operator splitting. *International Journal for Numerical Methods in Fluids*, 28(1):1–22, July 1998. doi: 10.1002/(sici)1097-0363(19980715)28:1<1::aid-fld690>3.0.co;2-b. URL https://doi.org/10.1002/(sici)1097-0363(19980715)28:1<1::aid-fld690>3.0.co;2-b.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

L. Ingleby, N. Calvet, J. Hernández, L Hartmann, et al. The evolution of accretion in young stellar objects: Strong accretors at 3-10 myr. *The Astrophysical Journal*, 790(1):47, July 2014. ISSN 1538-4357. doi: 10.1088/0004-637x/790/1/47. URL http://dx.doi.org/10.1088/0004-637X/790/1/47.

R. A. James and F. D. Kahn. Angular Momentum Transport in Stars, Following the Goldreich-Schubert Instability. , 12:332, June 1971.

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

A. Johansen and A. Youdin. Protoplanetary disk turbulence driven by the streaming instability: Nonlinear saturation and particle concentration. *The Astrophysical*

Journal, 662(1):627–641, June 2007. ISSN 1538-4357. doi: 10.1086/516730. URL http://dx.doi.org/10.1086/516730.

A. Johansen, T. Henning, and H. Klahr. Dust sedimentation and self-sustained kelvin-helmholtz turbulence in protoplanetary disk midplanes. *The Astrophysical Journal*, 643(2):1219–1232, June 2006a. ISSN 1538-4357. doi: 10.1086/502968. URL http://dx.doi.org/10.1086/502968.

A. Johansen, H. Klahr, and A. J. Mee. Turbulent diffusion in protoplanetary discs: the effect of an imposed magnetic field. *Monthly Notices of the Royal Astronomical Society: Letters*, 370(1):L71–L75, July 2006b. ISSN 1745-3925. doi: 10.1111/j.1745-3933.2006.00191.x. URL http://dx.doi.org/10.1111/j.1745-3933.2006.00191.x.

A. Johansen, M.-M M. Low, P. Lacerda, and M. Bizzarro. Growth of asteroids, planetary embryos, and kuiper belt objects by chondrule accretion. *Science Advances*, 1(3):e1500109, 2015. doi: 10.1126/sciadv.1500109. URL https://www.science.org/doi/abs/10.1126/sciadv.1500109.

W. R. Johnston. Binary minor planets compilation bundle v3.0, 2019. URL https://sbn.psi.edu/pds/resource/doi/binmp_3.0.html.

I. Kant. *Allgemeine Naturgeschichte und Theorie des Himmels oder Versuch von der Verfassung und dem mechanischen Ursprunge des ganzen Weltgebäudes nach Newtonischen Grundsätzen abgehandelt*. Peterson, Königsberg; Leipzig, 1755.

B. W. Keil. Astro-grips, the general relativistic implicit parallel solver for astrophysical fluid flows. 2010. doi: 10.11588/HEIDOK.00010188. URL http://archiv.ub.uni-heidelberg.de/volltextserver/id/eprint/10188.

S. S. Kimura, M. Kunitomo, and S. Z. Takahashi. From birth to death of protoplanetary discs: modelling their formation, evolution and dispersal. *Monthly Notices of the Royal Astronomical Society*, 461(2):2257–2265, 06 2016. ISSN 0035-8711. doi: 10.1093/mnras/stw1531. URL https://doi.org/10.1093/mnras/stw1531.

H. Klahr. Thermal baroclinic instabilities in accretion disks I: Combined dispersion relation for Goldreich-Schubert-Fricke Instability and Convective Overstability in disks around young stars. *Personal Communication*, 2024.

H Klahr and A. Hubbard. Convective overstability in radially stratified accretion disks under thermal relaxation. *The Astrophysical Journal*, 788(1):21, May 2014. ISSN 1538-4357. doi: 10.1088/0004-637x/788/1/21. URL http://dx.doi.org/10.1088/0004-637X/788/1/21.

H. Klahr and A. Schreiber. Turbulence sets the length scale for planetesimal formation: Local 2d simulations of streaming instability and planetesimal formation. *The Astrophysical Journal*, 901(1):54, September 2020. ISSN 1538-

4357. doi: 10.3847/1538-4357/abac58. URL http://dx.doi.org/10.3847/1538-4357/abac58.

H. Klahr and A. Schreiber. Testing the Jeans, Toomre, and Bonnor-Ebert Concepts for Planetesimal Formation: 3D Streaming-instability Simulations of Diffusion-regulated Formation of Planetesimals. , 911(1):9, April 2021. doi: 10.3847/1538-4357/abca9b.

H. Klahr, T. Pfeil, and A. Schreiber. *Instabilities and Flow Structures in Protoplanetary Disks: Setting the Stage for Planetesimal Formation*, page 2251–2286. Springer International Publishing, 2018. doi: 10.1007/978-3-319-55333-7_138. URL http://dx.doi.org/10.1007/978-3-319-55333-7_138.

H. Klahr, H. Baehr, and J. D. Melon-Fuksman. Thermal instabilities in accretion disks ii: Numerical experiments for the goldreich-schubert-fricke instability and the convective overstability in disks around young stars, 05 2023.

W. Kley. On the treatment of the Coriolis force in computational astrophysics. , 338:L37–L41, October 1998. doi: 10.48550/arXiv.astro-ph/9808351.

D. G. Korycansky. Two-dimensional Hydrodynamic Calculations of the Nonlinear Development of the Goldreich-Schubert-Fricke Instability in a Rotating Annulus. , 381:515, November 1991. doi: 10.1086/170675.

K. K. Kuo and R. Acharya. *Applications of Turbulent and Multiphase Combustion*. Wiley, April 2012. doi: 10.1002/9781118127575. URL https://doi.org/10.1002/9781118127575.

C. J. Lada and B. A. Wilking. The nature of the embedded population in the rho Ophiuchi dark cloud : mid-infrared observations. , 287:610–621, December 1984. doi: 10.1086/162719.

R Landshoff. *A Numerical Method for Treating Fluid Flow in the Presence of Shocks*. January 1955. doi: 10.2172/4364774. URL http://dx.doi.org/10.2172/4364774.

P. S. Laplace. *Exposition du système du monde*. Paris: imprimerie du cercle-social, 1796.

B. P. Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer Methods in Applied Mechanics and Engineering*, 19(1):59–98, June 1979. doi: 10.1016/0045-7825(79)90034-3. URL https://doi.org/10.1016/0045-7825(79)90034-3.

G. Lesur and J. C. B. Papaloizou. The subcritical baroclinic instability in local accretion disc models. *Astronomy and Astrophysics*, 513:A60, April 2010. ISSN 1432-0746. doi: 10.1051/0004-6361/200913594. URL http://dx.doi.org/10.1051/0004-6361/200913594.

G. Lesur, M. W. Kunz, and S. Fromang. Thanatology in protoplanetary discs: The combined influence of ohmic, hall, and ambipolar diffusion on dead zones. *Astronomy amp; Astrophysics*, 566:A56, June 2014. ISSN 1432-0746. doi: 10.1051/0004-6361/201423660. URL http://dx.doi.org/10.1051/0004-6361/201423660.

R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, August 2002. ISBN 9780511791253. doi: 10.1017/cbo9780511791253. URL http://dx.doi.org/10.1017/CBO9780511791253.

M.-K. Lin and A. N. Youdin. Cooling requirements for the vertical shear instability in protoplanetary disks. *The Astrophysical Journal*, 811(1):17, September 2015. ISSN 1538-4357. doi: 10.1088/0004-637x/811/1/17. URL http://dx.doi.org/10.1088/0004-637X/811/1/17.

D. Lynden-Bell and J. E. Pringle. The evolution of viscous discs and the origin of the nebular variables. , 168:603–637, September 1974. doi: 10.1093/mnras/168.3.603.

W. Lyra. Convective overstability in accretion disks: Three-dimensional linear analysis and nonlinear saturation. *The Astrophysical Journal*, 789(1):77, June 2014. ISSN 1538-4357. doi: 10.1088/0004-637x/789/1/77. URL http://dx.doi.org/10.1088/0004-637X/789/1/77.

W. Lyra and O. M. Umurhan. The Initial Conditions for Planet Formation: Turbulence Driven by Hydrodynamical Instabilities in Disks around Young Stars. , 131(1001):072001, July 2019. doi: 10.1088/1538-3873/aaf5ff.

C. F. Manara, G. Rosotti, L. Testi, A. Natta, et al. Evidence for a correlation between mass accretion rates onto young stars and the mass of their protoplanetary disks. *Astronomy amp; Astrophysics*, 591:L3, June 2016. ISSN 1432-0746. doi: 10.1051/0004-6361/201628549. URL http://dx.doi.org/10.1051/0004-6361/201628549.

N. Manger. High resolution simulations of structure formation in turbulent protoplanetary disks: A case study of the vertical shear instability. 2019. doi: 10.11588/HEIDOK.00026898. URL http://archiv.ub.uni-heidelberg.de/volltextserver/id/eprint/26898.

N. Manger and H. Klahr. Vortex formation and survival in protoplanetary discs subject to vertical shear instability. *Monthly Notices of the Royal Astronomical Society*, 480(2):2125–2136, 07 2018. ISSN 0035-8711. doi: 10.1093/mnras/sty1909. URL https://doi.org/10.1093/mnras/sty1909.

N. Manger, H. Klahr, W. Kley, and M. Flock. High resolution parameter study of the vertical shear instability. *Monthly Notices of the Royal Astronomical Society*, 499(2):1841–1853, October 2020. doi: 10.1093/mnras/staa2943. URL https://doi.org/10.1093/mnras/staa2943.

N. Manger, T. Pfeil, and H. Klahr. High-resolution parameter study of the vertical shear instability – ii: dependence on temperature gradient and cooling time. *Monthly Notices of the Royal Astronomical Society*, 508(4):5402–5409, September 2021. ISSN 1365-2966. doi: 10.1093/mnras/stab2599. URL http://dx.doi.org/10.1093/mnras/stab2599.

W. McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.

A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, et al. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, January 2017. ISSN 2376-5992. doi: 10.7717/peerj-cs.103. URL https://doi.org/10.7717/peerj-cs.103.

J. Meyer and H. Klahr. Simulations of asteroids and kuiper-belt objects ii: Merging history. in prep.

J. Meyer, J. Tjoa, and H. Klahr. Formation of asteroids and kuiper-belt objects via pebble cloud collapse i: Mass distribution and binary systems. in prep.

J. Meyer, J. D. Melon-Fuksman, and H. Klahr. Matrics: The implicit matrix-free eulerian hydrodynamics solver. *Astronomy & Astrophysics*, in production. doi: 10.1051/0004-6361/202348746.

G. Meynet and A. Maeder. Stellar evolution with rotation. I. The computational method and the inhibiting effect of the μ-gradient. , 321:465–476, May 1997.

A. Miotello, I. Kamp, T. Birnstiel, L. C. Cleeves, and A. Kataoka. Setting the Stage for Planet Formation: Measurements and Implications of the Fundamental Disk Properties. In S. Inutsuka, Y. Aikawa, T. Muto, K. Tomida, and M. Tamura, editors, *Protostars and Planets VII*, volume 534 of *Astronomical Society of the Pacific Conference Series*, page 501, July 2023. doi: 10.48550/arXiv.2203.09818.

A. Morbidelli and D. Vokrouhlický. The yarkovsky-driven origin of near-earth asteroids. *Icarus*, 163(1):120–134, 2003. ISSN 0019-1035. doi: https://doi.org/10.1016/S0019-1035(03)00047-2. URL https://www.sciencedirect.com/science/article/pii/S0019103503000472.

Y. Nakagawa, M. Sekiya, and C. Hayashi. Settling and growth of dust particles in a laminar phase of a low-mass solar nebula. , 67(3):375–390, September 1986. doi: 10.1016/0019-1035(86)90121-1.

R. P. Nelson, O. Gressel, and O. M. Umurhan. Linear and non-linear evolution of the vertical shear instability in accretion discs. *Monthly Notices of the Royal Astronomical Society*, 435(3):2610–2632, September 2013. doi: 10.1093/mnras/stt1475. URL https://doi.org/10.1093/mnras/stt1475.

D. Nesvorný, R. Li, J. B. Simon, A. N. Youdin, D. C. Richardson, R. Marschall, and W. M. Grundy. Binary planetesimal formation from gravitationally collapsing pebble clouds. *The Planetary Science Journal*, 2(1):27, February 2021. doi: 10.3847/psj/abd858. URL https://doi.org/10.3847/psj/abd858.

U. Nowak and L. Weimann. A family of newton codes for systems of highly nonlinear equations. Technical Report TR-91-10, ZIB, Takustr. 7, 14195 Berlin, 1992.

M. Omang, S. Børve, and J. Trulsen. Sph in spherical and cylindrical coordinates. *Journal of Computational Physics*, 213(1):391–412, March 2006. ISSN 0021-9991. doi: 10.1016/j.jcp.2005.08.023. URL http://dx.doi.org/10.1016/j.jcp.2005.08.023.

C. W. Ormel and J. N. Cuzzi. Closed-form expressions for particle relative velocities induced by turbulence. *Astronomy amp; Astrophysics*, 466(2):413–420, February 2007. ISSN 1432-0746. doi: 10.1051/0004-6361:20066899. URL http://dx.doi.org/10.1051/0004-6361:20066899.

B. Paczynski. Evolutionary processes in close binary systems. *Annual Review of Astronomy and Astrophysics*, 9(1):183–208, September 1971. doi: 10.1146/annurev.aa.09.090171.001151. URL https://doi.org/10.1146/annurev.aa.09.090171.001151.

The pandas development team. pandas-dev/pandas: Pandas, feb 2020. URL https://doi.org/10.5281/zenodo.3509134.

E.N. Parker. Dynamical theory of the solar wind. *Space Science Reviews*, 4(5–6), September 1965. ISSN 1572-9672. doi: 10.1007/bf00216273. URL http://dx.doi.org/10.1007/BF00216273.

B. Paxton, M. Cantiello, P. Arras, L. Bildsten, et al. Modules for Experiments in Stellar Astrophysics (MESA): Planets, Oscillations, Rotation, and Massive Stars. , 208(1):4, September 2013. doi: 10.1088/0067-0049/208/1/4.

J. W. Pearson and J. Pestana. Preconditioners for krylov subspace methods: An overview. *GAMM-Mitteilungen*, 43(4), October 2020. doi: 10.1002/gamm.202000015. URL https://doi.org/10.1002/gamm.202000015.

K. Pearson. Note on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London Series I*, 58:240–242, January 1895.

F. Pedregosa, G. Varoquaux, A. Gramfort, and V. others Michel. Scikit-learn: Machine learning in Python. 12:2825–2830, 2011.

B. Polak and H. Klahr. High-resolution study of planetesimal formation by gravitational collapse of pebble clouds. *The Astrophysical Journal*, 943(2):125, February 2023. doi: 10.3847/1538-4357/aca58f. URL https://doi.org/10.3847/1538-4357/aca58f.

H. S. Price, R. S. Varga, and J. E. Warren. Application of oscillation matrices to diffusion-convection equations. *Journal of Mathematics and Physics*, 45(1-4): 301–311, April 1966. doi: 10.1002/sapm1966451301. URL https://doi.org/10.1002/sapm1966451301.

A. Recktenwald, M. Lücke, and H. W. Müller. Taylor vortex formation in axial through-flow: Linear and weakly nonlinear analysis. *Phys. Rev. E*, 48:4444–4454, Dec 1993a. doi: 10.1103/PhysRevE.48.4444. URL https://link.aps.org/doi/10.1103/PhysRevE.48.4444.

A. Recktenwald, M Lücke, and H. W. Müller. Taylor vortex formation in axial through-flow: Linear and weakly nonlinear analysis. *Physical Review E*, 48(6): 4444–4454, December 1993b. doi: 10.1103/physreve.48.4444. URL https://doi.org/10.1103/physreve.48.4444.

G. Rüdiger, R. Arlt, and D. Shalybkov. Hydrodynamic stability in accretion disks under the combined influence of shear and density stratification. , 391:781–787, August 2002. doi: 10.1051/0004-6361:20020853.

Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, January 2003. doi: 10.1137/1.9780898718003. URL https://doi.org/10.1137/1.9780898718003.

Y. Saad and M. H. Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986. doi: 10.1137/0907058. URL https://doi.org/10.1137/0907058.

V. S. Safronov. On the gravitational instability in flattened systems with axial symmetry and non-uniform rotation. *Annales d'Astrophysique*, 23:979, February 1960.

V. S. Safronov. *Evolution of the Protoplanetary Cloud and Formation of the Earth and the Planets*. Israel Program for Scientific Translations, 1972.

U. Schäfer, C.-C. Yang, and A. Johansen. Initial mass function of planetesimals formed by the streaming instability. *Astronomy amp; Astrophysics*, 597:A69, January 2017. ISSN 1432-0746. doi: 10.1051/0004-6361/201629561. URL http://dx.doi.org/10.1051/0004-6361/201629561.

A. Schreiber. Diffusion limited planetesimal formation. 2018. doi: 10.11588/HEIDOK.00024579. URL http://archiv.ub.uni-heidelberg.de/volltextserver/id/eprint/24579.

A. Schreiber and H. Klahr. Azimuthal and vertical streaming instability at high dust-to-gas ratios and on the scales of planetesimal formation. *The Astrophysical Journal*, 861(1):47, June 2018. doi: 10.3847/1538-4357/aac3d4. URL https://doi.org/10.3847/1538-4357/aac3d4.

M. Schrimpf, J. Esteban, H. Warmeling, T. Färber, A. Behr, and A. Vorholt. The taylor-couette reactor – principles, design and applications. *AIChE Journal*, 67, 02 2021. doi: 10.1002/aic.17228.

M. A. Shadab, D. Balsara, W. Shyy, and K. Xu. Fifth order finite volume weno in general orthogonally - curvilinear coordinates. *Computers Fluids*, 190:398–424, 2019. ISSN 0045-7930. doi: https://doi.org/10.1016/j.compfluid.2019.06.031. URL https://www.sciencedirect.com/science/article/pii/S0045793018307989.

N. I. Shakura and R. A. Sunyaev. Black holes in binary systems. Observational appearance. , 24:337–355, January 1973.

H. Sierks, C. Barbieri, P. L. Lamy, R. Rodrigo, et al. On the nucleus structure and activity of comet 67p/churyumov-gerasimenko. *Science*, 347(6220), jan 2015. ISSN 1095-9203. doi: 10.1126/science.aaa1044. URL http://dx.doi.org/10.1126/science.aaa1044.

L. Siess, S. Goriely, and N. Langer. Nucleosynthesis of s-elements in rotating agb stars. *Astronomy amp; Astrophysics*, 415(3):1089–1097, February 2004. ISSN 1432-0746. doi: 10.1051/0004-6361:20034281. URL http://dx.doi.org/10.1051/0004-6361:20034281.

J. B. Simon, P. J. Armitage, A. N. Youdin, and R. Li. Evidence for universality in the initial planetesimal mass function. *The Astrophysical Journal Letters*, 847, 2017. URL https://api.semanticscholar.org/CorpusID:118969826.

G. D. Smith. *Numerical solution of partial differential equations*. Oxford University Press, London, England, 2 edition, October 1978.

H. A. Snyder. Stability of rotating couette flow. II. comparison with numerical results. *Physics of Fluids*, 11(8):1599, 1968. doi: 10.1063/1.1692167. URL https://doi.org/10.1063/1.1692167.

G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27(1):1–31, April 1978. doi: 10.1016/0021-9991(78)90023-2. URL https://doi.org/10.1016/0021-9991(78)90023-2.

J. R. Spencer, S. A. Stern, J. M. Moore, H. A. Weaver, et al. The geology and geophysics of kuiper belt object (486958) arrokoth. *Science*, 367(6481), February 2020. ISSN 1095-9203. doi: 10.1126/science.aay3999. URL http://dx.doi.org/10.1126/science.aay3999.

V. Springel. E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. *Monthly Notices of the Royal Astronomical Society*, 401(2):791–851, 01 2010. ISSN 0035-8711. doi: 10.1111/j.1365-2966.2009.15715.x. URL https://doi.org/10.1111/j.1365-2966.2009.15715.x.

S. A. Stern, H. A. Weaver, J. R. Spencer, et al. Initial results from the New Horizons exploration of 2014 MU_{69}, a small Kuiper Belt object. *Science*, 364(6441): aaw9771, May 2019. doi: 10.1126/science.aaw9771.

S. A. Stern, O. L. White, W. M. Grundy, B. A. Keeney, J. D. Hofgartner, D. Nesvorný, et al. The properties and origin of kuiper belt object arrokoth's large mounds. *The Planetary Science Journal*, 4(9):176, September 2023. ISSN 2632-3338. doi: 10.3847/psj/acf317. URL http://dx.doi.org/10.3847/PSJ/acf317.

G. G. Stokes. On the Effect of the Internal Friction of Fluids on the Motion of Pendulums. *Transactions of the Cambridge Philosophical Society*, 9:8, January 1851.

P. A. Sweet. The importance of rotation in stellar evolution. , 110:548, January 1950. doi: 10.1093/mnras/110.6.548.

G. I. Taylor. VIII. stability of a viscous liquid contained between two rotating cylinders. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 223(605-615):289–343, January 1923. doi: 10.1098/rsta.1923.0008. URL https://doi.org/10.1098/rsta.1923.0008.

The HDF Group. Hierarchical Data Format, version 5. URL https://github.com/HDFGroup/hdf5.

J. H. Tillotson. Metallic Equations of State For Hypervelocity Impact. General Atomic Report GA-3216. 1962. Technical Report, July 1962.

A. Toomre. On the gravitational stability of a disk of stars. , 139:1217–1238, May 1964. doi: 10.1086/147861.

E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics*. Springer, Berlin, Germany, 3 edition, April 2009.

J. K. Truelove, R. I. Klein, C. F. McKee, et al. Self-gravitational hydrodynamics with three-dimensional adaptive mesh refinement: Methodology and applications to molecular cloud collapse and fragmentation. *The Astrophysical Journal*, 495(2): 821–852, March 1998. doi: 10.1086/305329. URL https://doi.org/10.1086/305329.

V. Urpin. A comparison study of the vertical and magnetic shear instabilities in accretion discs. *Astronomy amp; Astrophysics*, 404(2):397–403, June 2003. ISSN 1432-0746. doi: 10.1051/0004-6361:20030513. URL http://dx.doi.org/10.1051/0004-6361:20030513.

V. Urpin and A. Brandenburg. Magnetic and vertical shear instabilities in accretion discs. *Monthly Notices of the Royal Astronomical Society*, 294(3):399–406, March

1998. ISSN 1365-2966. doi: 10.1111/j.1365-8711.1998.01118.x. URL http://dx.doi.org/10.1111/j.1365-8711.1998.01118.x.

H. A. van der Vorst. Preconditioning. In *Iterative Krylov Methods for Large Linear Systems*, pages 173–204. Cambridge University Press, April 2003. doi: 10.1017/cbo9780511615115.014. URL https://doi.org/10.1017/cbo9780511615115.014.

E. van Henson. Multigrid methods nonlinear problems: an overview. In Charles A. Bouman and Robert L. Stevenson, editors, *SPIE Proceedings*. SPIE, June 2003. doi: 10.1117/12.499473. URL https://doi.org/10.1117/12.499473.

B. van Leer. Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method. *Journal of Computational Physics*, 32(1):101–136, 1979. ISSN 0021-9991. doi: https://doi.org/10.1016/0021-9991(79)90145-1. URL https://www.sciencedirect.com/science/article/pii/0021999179901451.

G. Van Rossum and F. L. Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

P. Virtanen, R. Gommers, T. E. Oliphant, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

H. J. Voelk, F. C. Jones, G. E. Morfill, and S. Roeser. Collisions between Grains in a Turbulent Gas. , 85(3):316–325, May 1980.

C. F. von Weizsäcker. Die entstehung des planetensystems. *Naturwissenschaften*, 33 (1):8–14, January 1946. ISSN 1432-1904. doi: 10.1007/bf00643364. URL http://dx.doi.org/10.1007/BF00643364.

J. VonNeumann and R. D. Richtmyer. A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics*, 21(3):232–237, March 1950. ISSN 1089-7550. doi: 10.1063/1.1699639. URL http://dx.doi.org/10.1063/1.1699639.

S. Wang and E. Johnsen. High-order schemes for the euler equations in cylindrical/spherical coordinates. 01 2017.

M. L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. doi: 10.21105/joss.03021. URL https://doi.org/10.21105/joss.03021.

S. J. Weidenschilling. Aerodynamics of solid bodies in the solar nebula. , 180:57–70, July 1977. doi: 10.1093/mnras/180.2.57.

S. J. Weidenschilling. Dust to planetesimals: Settling and coagulation in the solar nebula. *Icarus*, 44(1):172–189, 1980. ISSN 0019-1035. doi: https://doi.org/10.1016/0019-1035(80)90064-0. URL https://www.sciencedirect.com/science/article/pii/0019103580900640.

F. L. Whipple. On certain aerodynamic processes for asteroids and comets. In Aina Elvius, editor, *From Plasma to Planet*, page 211, January 1972.

J. P. Williams and L. A. Cieza. Protoplanetary disks and their evolution. *Annual Review of Astronomy and Astrophysics*, 49(1):67–117, September 2011. ISSN 1545-4282. doi: 10.1146/annurev-astro-081710-102548. URL http://dx.doi.org/10.1146/annurev-astro-081710-102548.

F. Windmark, T. Birnstiel, C. Güttler, J. Blum, C. P. Dullemond, and T. Henning. Planetesimal formation by sweep-up: how the bouncing barrier can be beneficial to growth. *Astronomy amp; Astrophysics*, 540:A73, March 2012a. ISSN 1432-0746. doi: 10.1051/0004-6361/201118475. URL http://dx.doi.org/10.1051/0004-6361/201118475.

F. Windmark, T. Birnstiel, C. W. Ormel, and C. P. Dullemond. Breaking through: the effects of a velocity distribution on barriers to dust growth(corrigendum). *Astronomy amp; Astrophysics*, 548:C1, November 2012b. ISSN 1432-0746. doi: 10.1051/0004-6361/201220004e. URL http://dx.doi.org/10.1051/0004-6361/201220004e.

A. Youdin and A. Johansen. Protoplanetary disk turbulence driven by the streaming instability: Linear evolution and numerical methods. *The Astrophysical Journal*, 662(1):613–626, June 2007. ISSN 1538-4357. doi: 10.1086/516729. URL http://dx.doi.org/10.1086/516729.

A. N. Youdin and J. Goodman. Streaming instabilities in protoplanetary disks. *The Astrophysical Journal*, 620(1):459–469, February 2005. ISSN 1538-4357. doi: 10.1086/426895. URL http://dx.doi.org/10.1086/426895.

J. P. Zahn. Circulation and turbulence in rotating stars. , 265:115–132, November 1992.

D. Zhang, C. Jiang, D. Liang, and L. Cheng. A review on tvd schemes and a refined flux-limiter for steady-state calculations. *Journal of Computational Physics*, 302:114–154, 2015. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2015.08.042. URL https://www.sciencedirect.com/science/article/pii/S0021999115005707.

Y. Zhang, P. Michel, O. S. Barnouin, et al. Inferring interiors and structural history of top-shaped asteroids from external properties of asteroid (101955) bennu. *Nat. Commun.*, 13(1):4589, August 2022.

Danksagung

Die Erstellung dieser Arbeit ist in vielerlei Hinsicht einem nicht linearen Pfad gefolgt und so manche Schwierigkeit musste auf dem Weg überwunden werden. Bei vielen Problemen die aufgetreten sind wäre es mir alleine nicht möglich gewesen sie zu lösen und so ist diese Arbeit, das Produkt gemeinsamer Anstrengungen. Ich möchte mich an dieser Stelle bei all denen Bedanken, die mich auf meinem Weg unterstützt und mir so unglaublich viel Freundlichkeit entgegen gebracht haben.
Allem voran möchte ich mich bei Hubert Klahr für seine Betreuung bedanken. Hubert, ich bin dir sehr dankbar dafür, wie viel Freiheit du mir bei der Durchfürhung meiner Projekte, besonders der Entwicklung von MATRICS gelassen hast. Gleichzeitig hast du immer wieder Impulse gesetzt die zu einer Thesis führen, auf die ich stolz bin. Das hätte ohne dich nicht geklappt. Besonders dankbar bin ich dir natürlich dafür, dass du mir überhaupt erst die Möglichkeit gegeben hast meine Promotion in deiner Gruppe weiter zu führen obwohl zu diesem Zeitpunkt so ziemlich alles was ich über Planeten wusste war, dass es mindestens acht gibt. Auch möchte ich mich bei dir für deine ansteckende Begeisterung für Fluid-Instabilitäten, Hydrodynamik im allgemeinen und natürlich die Entstehung von Planeten Bedanken. Es hat immer Spass gemacht sich mit dir zu unterhalten. Mir hat zwar nach jedem Gespräch der Kopf geraucht aber ich war immer ein bischen motivierter als vorher. Ich bin dir sehr dankbar, dass ich von dir lernen durfte.
Als nächstes möchte ich mich bei Matthias Bartelmann bedanken. Als erstes dafür, dass du dich bereit erklärt hast diese Arbeit zu begutachten. Unbeschreiblich dankbar bin ich dir dafür, dass du nicht weniger getan hast als diese Arbeit auf halbem Wege zu retten. Ich kann mit sicherheit sagen, dass wenn du nicht gewesen wärst, diese Thesis nicht zustande gekommen wäre. Deswegen, danke! In dem gleichen Kontext möchte ich mich auch sehr herzlich bei Björn Malte Schäfer, Friedrich Röpke und Christian Fendt bedanken. Mein besonderer Dank gilt Kees Dullemond dafür dass er mich an Hubert vermittelt und auch dass er mir damit wieder Vertrauen in meine Arbeit gegeben hat. I also wish to give my special thanks to David Melon Fuksman for supporting me with my code paper and this thesis in general. I am very gratefuly, that you, not even once, hesitated to offer help.
Ich möchte mich auch bei meinen Großeltern bedanken, dafür, dass sie mir auf so vielfältige Art und Weise Hilfe geleistet haben. Die Urlaube die Janine und ich

durch eure Hilfe machen konnten haben sehr gut getan und auch einen wichtigen Beitrag zum Zustandekommen dieser Arbeit geleistet. Ich möchte mich auch bei den Hempflings; Matthias, Alexandra und Vanessa für ganz viel Unterstützung bedanken. Auch möchte ich meinem Bruder Lucas dafür danken, dass er mich immer wieder daran erinnert hat, dass es noch ein Leben ausserhalb der Promotion gibt. Zuletzt, und trotzdem am wichtigsten, möchte ich mich bei meiner Freundin Janine bedanken. Du hast mir in den letzten vier Jahren so viel Kraft gegeben und so viel Verständnis und Geduld entgegen gebracht, wie ich es nicht für möglich gehalten hätte. Auch ohne dich hätte ich diese Arbeit nicht verfassen können. Du bist der wunderbarste Mensch den ich je getroffen habe. Ich liebe dich, bleib so wie du bist.