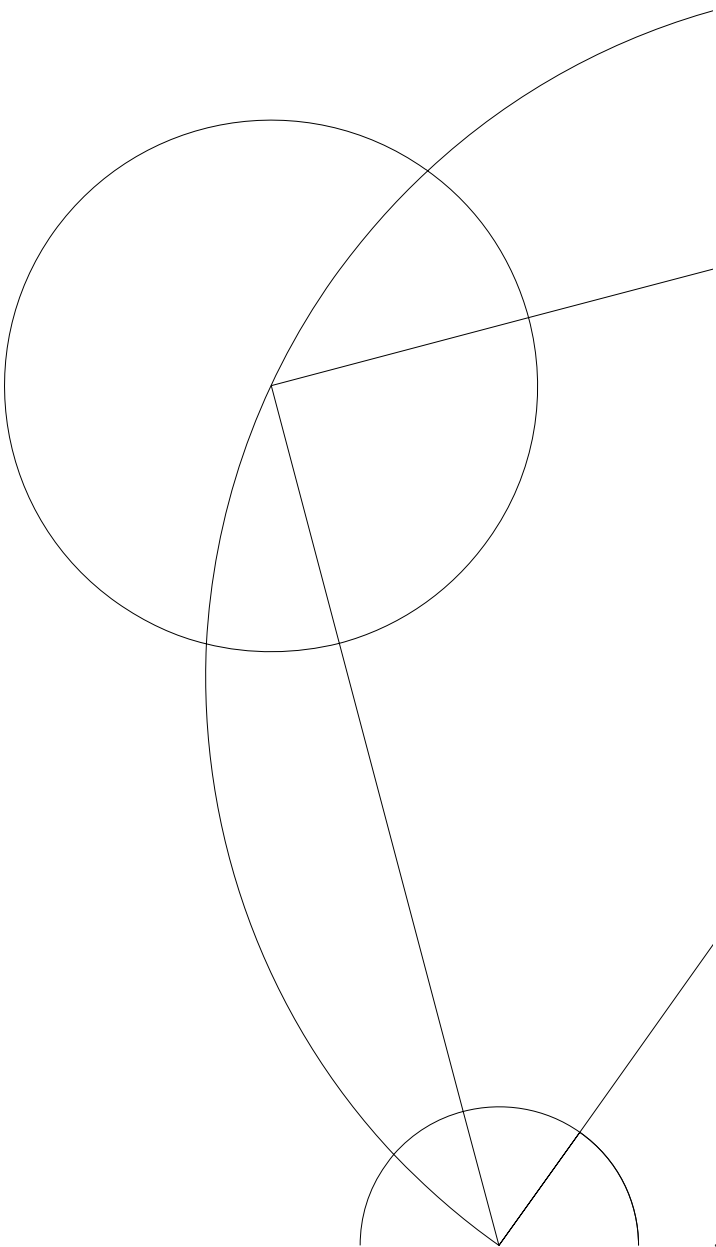
UNIVERSITY OF COPENHAGEN

Procedures for Modeling Dust Evolution

Masters thesis
Written by: Martin Sandberg
July 6, 2021

Under Supervison of:
Åke Nordlund & Troels Haugbølle

Department of Physics
University of Copenhagen



Abstract

New findings in several areas of astrophysics have put dust in focus. Mysteries of how larger dust particles are formed and on what timescales seem to elude us still. Therefore the need to explain the microscopic behavior of dust arises. Large-scale simulations of astrophysical phenomena such as molecular clouds have become highly efficient due to continued advances in computing and software that leverage these improvements. This opens up new possibilities for investigating new solutions to questions hard to answer in traditional ways.

In this thesis, a numeric model of dust evolution is developed, which includes the coagulation of dust particles and their fragmentation in the events of dust-dust collisions. The effects of this evolution is investigated by looking at the distribution of dust particles at a later time. To achieve this, a simple prototype model is developed and its results are tested against similar approaches, here it is found that the results are consistent with analytical approaches and with the results of other authors. The prototype also found that rapid growth occurs within very short timescales of about $\sim 30yr$, whereafter an equilibrium is reached. However, this model has no dynamic component which means the collisions are assumed to happen constantly.

This model is then implemented into the computing framework *Dispatch* as this framework has the necessary physics implemented in order to set up a more realistic simulation experiment. The framework handles hydrodynamics and the dynamic part of the particles. To probe the dust evolution in the cold interstellar medium, an experiment is set up with supersonic turbulence in a $1 \times 1 \times 1$ pc box. The model has a $1/100$ dust-to-gas ratio and the MRN distribution [1], with particle ranges of $5 - 250$ nm and an exponent of -3.5. The system is then evolved for a few turnover times, where a snapshot of the system is taken at every $\sim 1/4$ turnover time, meaning a timescale of $\sim 0.1Myr$. What is seen is the system grows very few larger particles $> 100\mu m$; This is seen already in the first snapshot. Hereafter the number density distribution sees very minute changes throughout several turnover times. This show a balance between the effect of coagulation and fragmentation is reached, especially for larger particles whose fragmentation contribution is high. therefore larger particles are likely to fragment. The fragmentation contribution is highly dependent on the relative velocities and the fragmentation velocity–which is still an area of some debate; thus it would be possible that the fragmentation velocity is not accurate. Perhaps a significant fraction of the particles are crystallized and thus have much higher fragmentation velocity. This would likely result in more significant grain growth. However, from the given results, it seems unlikely that the interstellar medium (ISM) is an ideal medium for grain growth.

In a second experiment. Setting up with the same particle distribution, with a slight adjustment; a fraction of the mass is put into a particle size outside this distribution. Thus the system starts with particles of size $5 - 250nm$ and a population of size $100\mu m$ particles. What is seen is that the distribution evolves very quickly, populating bins in size range $250nm - 100\mu m$ and particles $> 100\mu m$. However, these populations remain small, and the system reaches equilibrium quickly. This results in the distribution not changing significantly through timescales of $\sim 1Myr$, indicating what is put into the system is more or less what you get out millions of years later.

The particle distribution seen in the ISM is therefore dependent on the injected particles or possible sink for the particles. This means it is more likely that contributors such as AGB stars or protostars are more likely to account for the larger dust grains that are seen in the ISM.

Many aspects of dust evolution and its contribution is still up for debate. Hopefully, this implementation of dust evolution in *Dispatch* can assist further investigations into these areas of discussion, ultimately impacting our understanding of the initial dust size distribution available in protoplanetary disks and as a reservoir for planet formation.

Contents

1 Introduction

Dust is an essential ingredient in many processes throughout the universe. Seemingly most of the mass of terrestrial planets and cores of gas giants are made of dust accumulated to a great degree. Most of this dust is primarily made of metals and has its origin in stars created by fusion.

This dust forms much of what we see when we look out into the universe, it shapes the spectra we see from galaxies. As dust absorbs light and re-emits it, it blocks light from stars giving us the characteristic dark patches we seen in the night sky. We still do not really know exactly where this dust comes from. We observe that already in the very early universe copious amounts of dust was present, this timescale is shorter than the timescale for AGB stars, which raises questions as to where the distribution of dust we see came from. Another string contender is supernovae, but it cannot in itself contribute the observed amount of dust.

To form what we see we must have dust in range of different sizes, and to make sense of this; it is proposed that dust-dust interaction is both constructive and destructive. In other words dust have the opportunity to coagulate or fragment in collisions with one another.

In recent years the observations of exoplanets have blown up with new telescopes such as the Kepler space telescope continuously discovering more exoplanets. This leads us to believe that planets are commonplace in the universe and not such a rarity as once thought. However, little is known of how these planets form and indeed how they form so frequently and quickly as it seems to happen.

How planets form continues to be a gap in our understanding, Despite the fact that we are now in a time where we have managed to observe a staggering number of exoplanets. It looks more and more like a common occurrence for stars to have multiple exoplanets of different sizes and masses. Moreover, crucially we no longer only seem to find high mass and close-in planets; we have gotten better and better at detecting smaller Earth-like, low-mass in different orbits, likely terrestrial planets made primarily of rock material. The question is then what drives such prolific planet formation.

In newer studies, how such copious amounts of dust have formed in the early universe have proposed new solutions. Which seems to lead to a discrepancy in the timescales for dust creation and dust destruction. It has been proposed that the interstellar medium (ISM) might be responsible for some grain growth. This leads to questions of what the timescale for dust fragmentation is and if it is possible to coagulate dust in molecular clouds (MC) or the diffuse ISM.

These questions involved with the evolution of dust may be well explained provided a comprehensive model of dust coagulation and fragmentation.

Unfortunately, we do not possess the ability to do universe-scale laboratory experiments in the real world. So we must resort to a virtual reality, where we will have to try and emulate the real world. To do this, we must identify the dominating features of the problem and find a way to formulate them numerically. Luckily, this has been done throughout the history of science since the invention of the computer, and we have gotten quite good at it. Furthermore, computers have continued to improve at almost alarming rates, enabling the possibility of adding new features to existing simulation frameworks. As well as enabling high resolution and larger scales.

The goal of this thesis is to formulate the description of dust so that it can be included in big-scale simulations in the computing framework *Dispatch*. Nevertheless, to make a realistic model of how dust behaves, we must first describe the dynamics of dust and, secondly, the evolution of dust.

2 Dust Origins

To begin we will take a look at where dust originates from such that we have a foundational understanding of the existence of dust in the universe, which we will later build on in terms of dynamics and evolution.

2.1 Dust in the interstellar medium

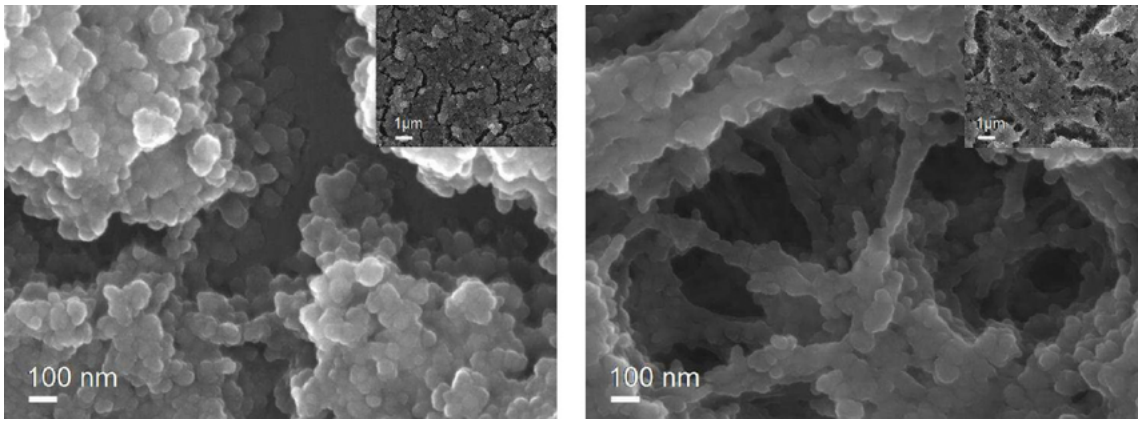


FIGURE 1. *Field Emission Scanning Electron Microscopy (FESEM) images showing the difference between amorphous(left) magnesium silicates and crystalline magnesium material(right) Credit: [2].*

As they are found in molecular clouds and protoplanetary disks, dust particles have their starting point in stars, as stars are responsible for the metals that make up dust particles. It is found that silicate dust, which is mostly made up of silicon, oxygen, iron, and magnesium, is prevalent. Other types of dust include carbonaceous dust, made up of groups of carbon atoms mixed with hydrogen. Lastly, ice dust is made up of simple molecules such as water, methane, and carbon monoxide. These tend to form in very cold mediums, and when they are heated up, they tend to evaporate and release accreted material. This dust type will not be in focus throughout this thesis.

The makeup of a molecular cloud is dependent on the chemistry of the stars from which the cloud is made up. The depletion of different materials is different, however. The depletion of various elements in various mediums has been observed and explained from new data and new models. It is found that substantial depletion of materials Si and Fe in a cold medium is required, meaning these must-have coagulated to form dust grains [3].

Silicate particles can crystallize if heated sufficiently. In the case of silicates, it seems around 1000 K for several days, undergoing annealing, meaning the atoms move to a crystal structure [4]. Crystallization gives the particles a tough and compact outer shell. These crystalline materials can form in high mass-loss oxygen-rich AGB stars, where the envelope is dense [5]. The crystalline material is also found in envelopes of a protostar, where it suggests that the material has started as amorphous silicate in the disk undergoing annealing in the hot inner disk and being transported to the envelope by outflows [6].

The way dusts then end up in molecular clouds and later end up in disks is by contribution from stars, as they are the origins of the necessary metals. One known source of these metals is the processes in asymptotic giant branch (AGB) stars. These types of stars develop circumstellar envelopes where molecules are allowed to cool and form dust particles. Stellar winds then

release these particles into the surrounding medium, the total dust input from Oxygen-rich AGB stars is in the order of $(7.5 - 8.4) \cdot 10^{-7}\ M_{\odot} yr^{-1}$, where for Carbon-rich AGB it is a magnitude slower at $7.8 \cdot 10^{-8}\ M_{\odot} yr^{-1}$, which only accounts for 2% of the dust in the ISM [7].

AGB stars were thought to be the primary source of dust in the present-day Milky Way. However, we know from quasars that the universe was already filled with dust after 700 million years, meaning the timescales of stellar evolution of such stars are too long for them to be considered the most significant contributor of dust in the early universe[8].

Then supernovae were considered, specifically rapidly-evolving massive stars that end in a type-II supernova, rich in heavy elements, and are likely to condense into dust particles. Therefore type II SN was considered likely to be the major source of dust in the early universe [8]. However, it has been found that, meanwhile, SNe are efficient dust producers; they are not enough on their own, as they destroy most of the larger grains they create in reverse shocks. Destruction happens because of various interactions, mainly dust-dust collision, where accelerated grains collide, and gas-dust, where the gas is heated and brings dust back to gas phase. New calculations have shown that only $10 - 20\%$ of silicate grains in size range $\sim 0.25 - 0.5\ \mu m$ survive these shocks [9], meaning from a 19 $M_{\odot}$ star it is possible to get 0.1 $M_{\odot}$ dust, in the right conditions.

This has then lead to the question being posed, is dust produced in the ISM? These have proven to be difficult as the cold diffuse ISM does not possess the ideal conditions for grain growth [10]. In recent times protoplanetary systems have been considered as significant dust contributors.

Mm-sized particles have been found in envelopes of protostars, and it is suggested that these are formed by grain growth in the disk and then being transported to the envelope by outflows [11]. Looking at how much mass is being ejected by a protostar. It is estimated that 1 $M_{\odot}$ goes into the star, 1 $M_{\odot}$ is simply let go, and $0.5 - 1$ $M_{\odot}$ gets ejected after coming close to the protostar, meaning this material could be larger processed grains.

This would suggest that the ISM is constantly being supplied with larger grains. However, larger grains do not seem prevalent in the diffuse ISM, implying that these larger grains fragment relatively fast.

It is well established that the most common way for grain destruction is by shocks in the ISM, generated mainly by supernovae. These can reach velocities of $> 200\ km/s$, depending on the distance to the supernova [12]. The rate of destruction of silicate grains has been calculated given new estimates of supernova rates to be in the order of $\sim 2 - 3\ Gyr$. This results in a discrepancy between the timescale of dust creation and the timescale for dust destruction[12].

As these dust particles then exist in the diffuse ISM, they are now vulnerable to irradiation and shocks. It is, therefore, widely believed that dust is processed such that crystallized material goes to porous amorphous material by amorphization [13]. Much of this material enters star-forming regions and protostellar systems, where it is processed and ends up as crystallized material. However, it is not well known how this distribution looks, how much of the dust is crystalline material and how much is amorphous. But there seems to be a balancing between the destruction and creation of crystallized material. This will significantly impact how easy these particles will fragment, as will be important in later discussions of dust evolution.

2.2 Observing dust in the interstellar medium

Dust can be observed in the ISM in different ways. One immediate way is the way that it blocks out light. If we were to point a telescope at the night sky, big patches of complete darkness appear, as well as patches of reddened stars. This is due to how light is dimmed as it travels clouds of dust, also called extinction. Dust absorbs the light and transmits it in the infrared spectrum. Reddening happens as dust scatters blue light and lets through the red light, thus making stars appear redder. An easy to recognize example of this is sunset, where the sky reddens. This happens because of Rayleigh scattering, which applies to particles smaller than the wavelength of the incoming light. In this case, air molecules are around 1 nm, these scatter the blue light at around 500 nm in the atmosphere, and the sky appears redder. Interstellar dust, however, is shown to have a weaker dependence on the wavelength of the incoming light. The same features are still seen as blue light from stars is scattered.[14]. A well-known cloud that clearly shows these consequences of dust is Barnard 68 shown in figure 2.

The extinction still hints at the size distribution of the dust, in Mathis (1997)[1] it is shown that the extinction fits with sizes going from 5 nm to 500 nm where the number density is distributed according to a power-law with exponent in the range -3.3 to -3.6 for graphite (SiC). The dust-to-gas ratio is a parameter that is relatively well established but can also vary throughout a MC[15]. It is generally around 1/100 meaning the majority of the mass in the ISM is gas.

Dust can be observed by its thermal emission, primarily in the far-infrared and sub-millimeter wavelengths. Properties of the spectral energy distribution (SED) can explain things like what type of dust is present in the system.

The Spitzer telescope observed type II SN ejecta 500-700 days after outburst in MID-infrared and found profiles consistent with dust emission profiles and optical extinction increasing over time after the outburst [8].

AGB stars have been observed to contain the molecules C2H2, HCN, SiS, SiO, SiC2, and CS in their envelope [16]. Meaning that indeed atoms are being released from the star and condense in its envelope, however lacking iron.

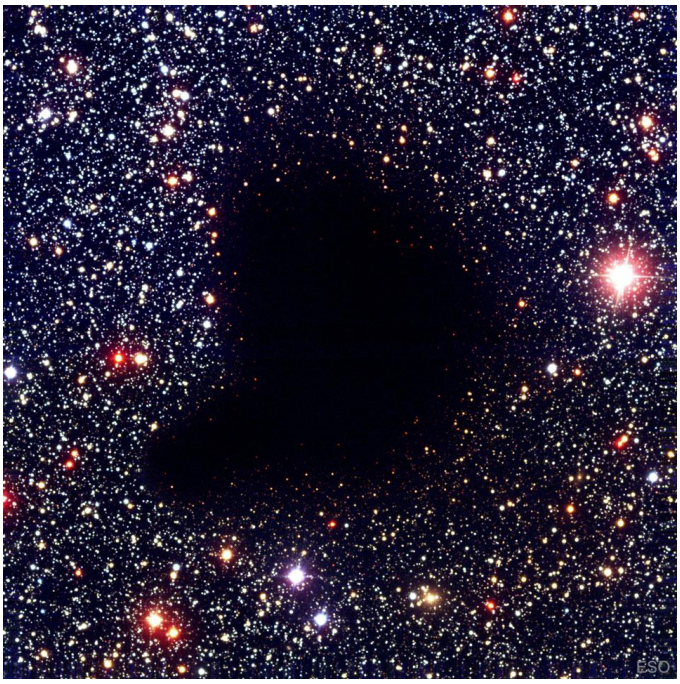


FIGURE 2. *A picture of molecular cloud Barnard 68, significant in that it is close to Earth, meaning there are no bright stars between us and it and it blocks light from star behind it, making a big black spot. It can also be seen at the rim of the cloud that stars appear red, likely due to reddening. Credit: FORS Team, 8.2-meter VLT Antu, ESO*

2.3 Observing dust in protoplanetary disks: DSHARP

In disks of gas and dust in disks around newly formed stars, it is thought to be the ideal conditions for planet formation. These disks and their features have recently been studied in the first large-scale observation program at Atacama Large Millimeter Array (ALMA), dubbed "Disk Substructures at High Angular Resolution Project" (DSHARP). The array focused on 20 specific targets for up to multiple months at a time, thus giving researchers a new and unique look into planet-forming regions in the disks and their structures, providing hints into what the underlying dynamics required to form planets are.

The goal of the project is to find characteristic substructures in the distribution of solid particles in protoplanetary disks. This is achieved by observations using a very high resolution of 5 AU, Full-Width-Half-Maximum in the 240 GHz band.

Dust is heated up by friction, the transport of angular momentum, and, radiation from the star. This heat is radiated from the dust and shows up as a continuum emission or as blackbody radiation. From the shape of this continuum emission, it is possible to determine how the solids are distributed, also features in observations have been used to describe how the dust-to-gas ratio varies in the disk[17].

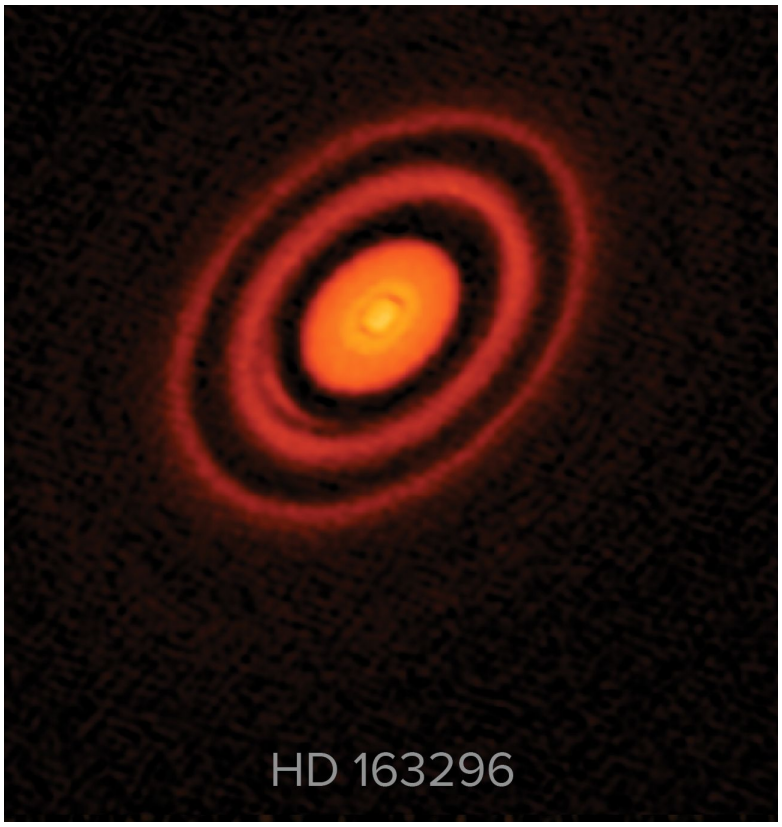


FIGURE 3. *Picture of the system HD163296, from the DSHARP observation program.*

In figure 3, an example of one of the 20 targets that the ALMA telescopes observed in the DSHARP. With remarkably distinct gaps between the core and outer rings. These substructures are shared among the DSHARP targets, some more distinct than others, and found at varying radii (1 AU to 100 AU). They are mostly relatively compact, no thicker than 10 AU, and varying in how distinct they are. These suggest that we have areas of trapping for particles of mm/cm size. These could lead to areas of coagulation, giving rise to larger particles [17]. In previous models of planet formation, these substructures a largely missing. Meaning DSHARP made a considerable impact putting in focus these substructures that likely play an essential role in many aspects of planet formation.

The coagulation of smaller dust particles to larger ones has long been believed to be the crucial first step to forming planets. It is well established that dust can coagulate to go from sub-micron to several microns in observations looking at Herbig Ae/Be systems by Bouwman et al.(2001)[18] and Boekel(2003)[19] who further go on to suggesting a continuous replenishment of micron-sized particles, at the disk surface. Also, new improvements to the large scale ALMA VLA have given rise to new observations where it has been found that mm size dust particles exist in the outer rings of a protoplanetary disk[20], even $1 > cm$ sized particles have been found in the inner rings of a disk[21]. All these observations point to the significance of dust evolution and seem a likely candidate for the first phases of planet formation.

In the solar system, we find dust particles in size range of 0.3 μm in diameter (Jessberger et al. 2001), even μm sized particles have been found in meteorites. And sample studies of comets also suggest that $\geq 1\mu m$ grains are produced in the solar system (Brownlee et al. 2006; Zolensky et al. 2006). Typically the particle size found in primitive meteorites (chondrites, material from early solar system) is millimeter-sized particles called chondrules, making up 80% of the chondrite (Weisberg, McCoy & Krot 2006).

3 Dust dynamics

In this thesis, our model of coagulation and fragmentation will be driven by the relative velocities of dust particles. For different-sized grains, different sources of relative velocity are essential. The velocities will determine to which degree the effects of coagulation and fragmentation, respectively, are significant. It is then clear that we must investigate the macroscopic dynamics of protoplanetary disks and molecular clouds to determine the dominant source of relative velocity.

Gas is located in molecular clouds, and these can become too dense and cool, which leads to collapse and the possibility of sparking the formation a protostar. Due to the conservation of angular momentum, a disk is formed. Through observation, we know that the disk over a few million years will vanish, and in some cases, leave nothing but planets and some asteroids and meteors behind. We know from observations that molecular clouds are dominated by gas in terms of mass. Therefore It is clear that to understand dust dynamics; we must understand the relationship between gas and dust motion.

The gas can be described by the fluid representation equation of motion. Here mass conservation is represented by the continuity equation

$$\frac{\partial}{\partial t}\rho_g + \nabla(\rho_g \boldsymbol{u}) = 0 \ . \tag{3.1}$$

Here ρ_g is the volume density of the gas and $\boldsymbol{u}$ the velocity of the gas. Momentum conservation in a fluid is given by the momentum equation as

$$\rho_g \left(\frac{\partial}{\partial t}\boldsymbol{u} + \boldsymbol{u}\nabla\boldsymbol{u} \right) = \boldsymbol{F}_{tot} \ . \tag{3.2}$$

Here movement of the gas in time is dependent on the forces acting upon it. The force could as an example be given as

$$\boldsymbol{F}_{tot} = -\nabla P - \rho_g \nabla\phi \ , \tag{3.3}$$

where $\boldsymbol{F}_{tot}$ is force per unit volume, P is the gas pressure, and ϕ the gravitational potential. Other forces can be considered, such as the divergence of the viscous stress tensor or, in the case of magnetohydrodynamics, where the gas is ionized and feels a Lorentz force.

The dynamics of the dust can then be described, in the same representation, essentially the same way, however, with a difference in what forces act upon it. As the dust will be described as a pressureless fluid

$$\frac{\partial}{\partial t}\boldsymbol{u} + \boldsymbol{u}\nabla\boldsymbol{u} = -\nabla\phi + \boldsymbol{f}_{drag} \ . \tag{3.4}$$

Here $\boldsymbol{f}_{drag}$ is the force per unit mass. The dust will feel the gravitational potential. However, the pressure gradient term is the part that separates the dust and the gas. Nonetheless, the dust will not feel this force directly in its interaction with the gas that acts on the dust via drag forces. This we will go into in more detail in the sections below.

3.1 Brownian motion

The simplest form of motion in particles in a gas medium comes from the random fluctuations in such a medium, giving rise to random movement in random directions. This type of motion

was first described by Robert Brown(1827) when he discovered pollen suspended in water would move about randomly[22]. The motion was thus named Brownian motion. There will be no preferred direction of flow within a medium of thermal equilibrium, meaning the fluids' angular and linear momentum is 0 and does not change over time. The relative Brownian motion is dependent on the mass of the particles and is given by

$$\Delta u_{\mathrm{BM}} = \sqrt{\frac{8k_b T(m_i + m_j)}{m_i m_j}} \ . \tag{3.5}$$

Here i and j are the two particles and T the temperature. From this, it is clearly seen that this velocity source will be more dominant and influential for small particles as it decreases with the mass of the objects. Meaning we will likely see a limit for particles of a specific size where Brownian motion is no longer relevant. The movement is random, meaning that over time particles will spread out in the medium. Such that a macroscopic feature of Brownian motion describes a form of diffusion. In our case, this will be relevant as small particles suspended in a gas, either the case for a protoplanetary disk of high density or in a molecular cloud, but mediums that can primarily be described as being locally in thermal equilibrium.

If we look at some examples involving equal-sized particles, the relative velocities caused by Brownian motion in a protoplanetary disk where we assume $T \sim 200\ K$ for 0.1 μm sized particles, then we have $\Delta u_{BM} \sim 4.5\ cm/s$, and for millimeter-sized particles, we already have $\Delta u_{BM} \sim 10^{-6}\ cm/s$. Thus the fall-off is quite significant for large particles.

For a molecular cloud, the temperature could be around $T = 10\ K$ for a cold medium, meaning the velocities are roughly 4-5 times smaller, meaning only $\Delta u_{BM} \sim 1\ cm/s$ for a 0.1 μm sized particle, and even lower for larger particles.

3.2 Gas drag forces

Firstly, we will work under the assumption that the gas in the regions that we examine corresponds to the dominant part of the mass. This significance of gas also means that since the dust velocities are dependent on the gas velocities by way of some coupling via drag forces. We will have to consider what velocities the gas is dominated by.

Dust particles will 'feel' drag from the gas by way of a drag force. This drag force will depend on the size of the dust particles and the velocity difference between the particles and the gas as

$$\boldsymbol{F}_{drag} = C_d(\pi a^2)\left(\frac{1}{2}\rho_g \Delta \boldsymbol{u}^2\right) \ , \tag{3.6}$$

C_d is a drag coefficient that depends on many different parameters involved with fluids, such as the viscosity, flow speed, et cetera. πa^2 represents the cross-section of the particle of radius a, and the last parentheses represent the dynamical pressure, where ρ_g is the gas density and $\Delta \boldsymbol{u}$ is the relative velocity between the gas and the dust particle.

It is beneficial to look at the time it takes for a dust particle to be stopped. This is called the stopping time for a particle and is defined by the particle's momentum divided by the drag force.

$$\tau_s = \frac{m\boldsymbol{v}}{\boldsymbol{F}_{drag}} \ , \tag{3.7}$$

where m is the mass of the dust particle, and v the velocity, there are then different regimes of the drag force. Which will determine the gas-to-dust coupling. These are split into regimes depending on the size of the particles and the mean-free path of the gas. When the mean-free-path λ_{mfp} of gas molecules is larger than the cross-section of the dust grains of radius a such that $\lambda_{mfp} > a\frac{4}{9}$, the particle will 'see' the gas molecules one at a time. And thus, the interaction is treated by the Epstein regime; given that the grains are solid spheres of fixed radius, the effects due to large dust concentrations are neglected. Moreover, we assume the specular reflections on the grains are negligible together with the differential velocity compared to the gas sound speed is negligible[23]. Then we can express the drag force and thus the stopping time as

$$\boldsymbol{F}_{drag} = -\frac{m\Delta\boldsymbol{u}}{\tau_s} \tag{3.8}$$

$$\tau_s = \frac{\rho_s a}{\rho_g c_s} \ , \tag{3.9}$$

where ρ_s is the solid density of a dust grain, a is the grain size, ρ_g is the local gas density, and c_s is the local gas sound speed.

For the cases of bigger particles given if the mean-free-path λ_{mfp} of the gas molecules is smaller than the cross-section of the dust grains such that $\lambda_{mfp} < a\frac{4}{9}$. Then the particles will see many gas molecules at once, and the Stokes regime now describes the interaction. In this case, the dust particle is surrounded locally by a viscous fluid described by the Reynolds number (Re) for the gas, which is the relationship between inertia and viscosity. for $Re \gg 1$ then

$$\tau_s = \frac{6\rho_s a}{\rho_g|\Delta\boldsymbol{u}|} \ , \tag{3.10}$$

where the general drag force (3.6) is valid. The drag coefficient then is dependent on the Reynolds number, which describes how laminar or turbulent a flow will be, helping to predict flow patterns, in this case, it is given as

$$Re = \frac{2a|\Delta\boldsymbol{u}|}{v_g} \ , \tag{3.11}$$

where v_g is the kinematic viscosity of the gas, in the figure4, it is seen how the drag coefficient can range from 0.1-1.5 for given Reynolds numbers, which describe the flow for different regimes.

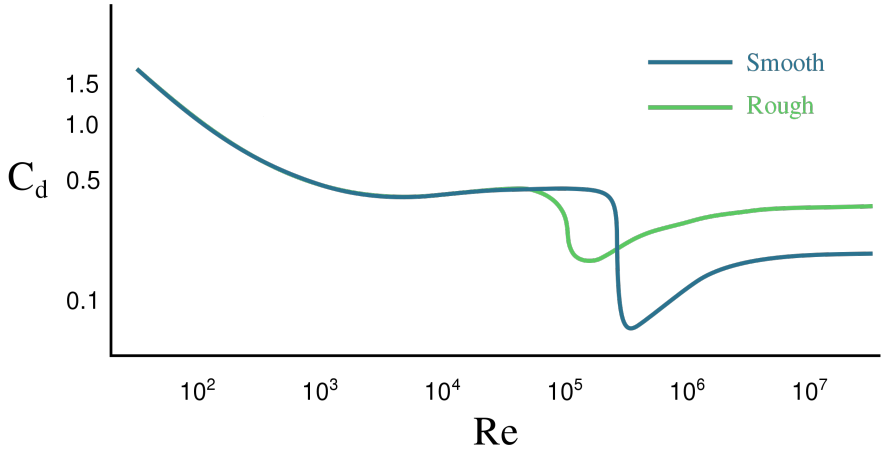


FIGURE 4. *Drag coefficient for a sphere as a function of Reynolds number, from laboratory experiments. 'Smooth' is for a sphere with a smooth surface and 'Rough' is for a sphere with a rougher surface. Credit: [24]*

In a turbulent flow, the coupling between the dust and the gas can be described by the dimensionless constant, the Stokes number which is defined as

$$\mathrm{St} := \frac{\tau_s}{\tau_{ed}} \ , \tag{3.12}$$

where τ_{ed} is the eddy turnover time. Thus it is defined as a ratio between two characteristic times, one for the particle and one for the flow. From this, it is clear that a Stokes number of $\gg 1$ will mean that the particle will detach from the flow. Conversely, a Stokes number of $\ll 1$ would mean a particle will follow the flow closely. The Stokes number is dependent on the drag force on the particle, meaning it is also dependent on which regime we investigate.

3.2.1 Stokes number in a protoplanetary disk

If we focus on the *Epstein regime* and look at eddy turnover time in a protoplanetary disk then, following [25] their calculation provide

$$\tau_{ed} = \alpha^{1-2q} \frac{1}{\Omega_k} \ , \tag{3.13}$$

where α is the turbulent parameter for disk and Ω_k is the Keplerian period of the disk. q is the exponent parameter from the dimensional analysis in Birnstiel (2010). The dust-gas coupling highly depends on the eddy turnover, so does the relative velocities between dust and gas in the disk; thus these will therefore also depend highly on this parameter. We will here follow [25] and set it to 1/2. Simplifying the eddy turnover time to

$$\tau_{ed} = \frac{1}{\Omega_k} \ . \tag{3.14}$$

Then combining Eq. (3.9) and the previous to get the simple term for the Stokes number in a protoplanetary disk assuming the Epstein regime

$$\mathrm{St} = \frac{\pi \rho_s \mathrm{a}}{2\Sigma_g} \quad \text{for } a < \frac{9}{4}\lambda_{mfp} \ . \tag{3.15}$$

From this, we can determine that the coupling between gas and dust is linearly dependent on the size of the dust grains. Big grains will give high Stokes numbers, and small grains will have low Stokes numbers.

In order to examine this further and get an idea of how coupled a specific grain would be in a disk, we must then make some assumptions about the gas in the disk and grain density. Assuming the solid density of the grains to be $\rho_s = 2\ g\ cm^{-3}$[26] and a gas surface density of 18 $g\ cm^{-2}$ [27], we can now make a calculation of the Stokes number in the Epstein regime for given particle size. Micron sized particles will have Stokes number of order 10^{-5}, and particles of a few cm will near St = 1. Meaning in a protoplanetary disk, small dust grains will 'feel' the gas much more and move with it, whereas larger grains will move less with the gas, however not completely decoupled.

3.2.2 Stokes number in a molecular cloud

If we shift our focus to molecular clouds, then to understand the gas-dust coupling, we must understand the eddy turnover time in a molecular cloud (MC) or giant molecular cloud (GMC). To do this, we must look at the dynamics of MCs. MCs are governed by fluctuations in density, self-gravity, magnetic interactions, amongst other processes. Fluctuations in density mean that in regions of MCs, there are opportunities for star formation to happen. What keeps MCs from collapsing as a whole is the presence of gas pressure, rotation and turbulence. What is shown to be the dominant factor in determining collapse in MCs is turbulence. On larger scales, it prevents collapse of the MC; on a small scale, it promotes local collapse [28], giving rise to star-forming regions. Suppose we were to look at easily observable quantities of MCs such as size, velocity dispersion, and mass surface density. Larson (1981)[28] found scaling relationships between these aforementioned observable quantities. These have been used time and time again ever since to describe the dynamics of MCs. These relations are given as [29]

$$\sigma_v \sim 1.2\ km/s \cdot \left(\frac{L}{1\ pc}\right)^{0.38} , \tag{3.16}$$

$$\frac{2\sigma_v L^2}{GM} \sim 1 , \tag{3.17}$$

$$n \sim n_0 \left(\frac{L}{1\ pc}\right)^{-1.1} . \tag{3.18}$$

Here the first describes a power-law relationship between the velocity dispersion σ_v and the size L of the MC. The second describes self-gravitational equilibrium, and the third describes an inverse relationship between mean density n and the size. The sound crossing time in the MC is equal to

$$\tau_{ed} = \tau_{cross} = \frac{L}{2\sigma_v} . \tag{3.19}$$

This gives a Stokes number as a function of stopping time

$$\mathrm{St} = \frac{2\sigma_v \tau_s}{L} . \tag{3.20}$$

At the relatively low densities prevalent in molecular clouds, the most relevant regime is the Epstein regime, which includes the assumption that particles only see gas molecules one at a time. For well-coupled particles, this means that the thermal properties of the gas dominate the relative velocity, this means that the sound speed such that using Eq. (3.9) then

$$\mathrm{St} = \frac{2\rho_s a \sigma_v}{\rho_g c_s L} , \tag{3.21}$$

from this, we can make some estimates with some additional assumptions. We assume that the average number density of an MC is given as $n \sim 1000\ cm^{-3}$, even though these densities can reach as high as 10^5 in high-density regions, such as clumps. Using the molecular mass of the gas is $\mu = 2.37$, we then have $\rho_g = n\mu m_p$, where m_p is the mass of a proton. Thus the sound speed at $T = 10\ K$ is $c_s = (kT/(\mu m_p))^{1/2} \sim 186\ m/s$. Now using the Larson relations if we have a MC of a size $L = 1\ pc$ $\sigma_v = 1.2\ km/s$. When the Stokes number is around 0.1 we find the particle size is around 0.5 μm, thus particles under and around this size should be tightly coupled to the gas flow.

From Mark Falkenstrom's master thesis experiments, he found that around Stokes number of around 0.1 and below, the relative velocities between the gas and dust were dominated by the sound speed.

For particles with a Stokes number around 1, the particles are not well coupled to the gas. Thus the relative velocities are not dominated by the sound speed anymore, as the particles will no longer follow the flow of the gas. This means the relative velocity is dominated by the larger stochastic velocity dispersion as described by the Larson relations. Thus the Stokes number is more accurately described as

$$\mathrm{St} = \frac{2\rho_s a}{\rho_g L}\ . \tag{3.22}$$

Looking for when the Stokes number is around 1 we find that a $\sim 30\ \mu m$ particle is necessary and particles of this size move on their own with little influence from the gas.

3.3 Turbulent motion of gas

There is a differential rotation in a protoplanetary disk, giving rise to shear between elements at different radii. This can power various instabilities in the disk, such as magnetorotational-instabilities (MRI), Kelvin-Helmholtz instabilities, or temperature gradient-related instabilities. These instabilities drive turbulence in the disk.

Based on the coupling between the turbulent gas and the dust particles, it is possible to derive a closed-form expression for the relative velocities for the turbulent motion of dust particles, this was done by Ormel & Cuzzi (2007) [30] and for equal-sized particles results in

$$\Delta u_{\mathrm{TM}} \approx \begin{cases} c_{\mathrm{s}}\sqrt{2\alpha \mathrm{St}} & \text{for St} \ll 1 \\ c_{\mathrm{s}}\sqrt{\frac{2\alpha}{\mathrm{St}}} & \text{for St} \gg 1. \end{cases}\ . \tag{3.23}$$

Here α is the turbulence parameter is an effective way to parameterize the turbulent motion. We will use this for prototyping our fragmentation model and use the commonly used value 10^{-3} for protoplanetary disks. Looking at *DSHARP* data, the value seems a good approximation, even though it can change throughout the disk[17], but this is not our focus. With the relative turbulent velocity, we can see how the different grain sizes would move along with the gas or be less coupled.

For tightly coupled particles in a protoplanetary disk with $T \sim 200\ K$ the relative velocity caused by turbulent motion for a 0.1 μm sized particle is $\Delta u_{\mathrm{TM}} = 4.3\ cm/s$ and for millimeter-sized it will be $\Delta u_{\mathrm{TM}} = 435\ cm/s$, clearly showing that it will dominate for larger particles. From this, it is also possible to show that the turbulent motion of the gas is dominant, as a

source for relative velocity for particles of Stokes number of $> 10^{-6}$ which under the assumptions will be for particles of size $> 0.1\ \mu m$. For particles under this size, the Brownian motion will dominate the relative velocities.

In molecular clouds, the relative turbulent motion between particles is dependent on the Stokes number, as described earlier. There is a cut-off where the sound speed is dominant for highly coupled particles, and for decoupled particles, the velocity dispersion is dominant.

4 Particle descriptions

Here we discuss the different ways of describing and understanding particles. These have different benefits, and limitations that we might require in our model of dust particles. We want different properties of particles included in our model thus we need to apply a combination of methods in order to describe the important factors of our problem.

4.1 Eulerian-Lagrangian descriptions

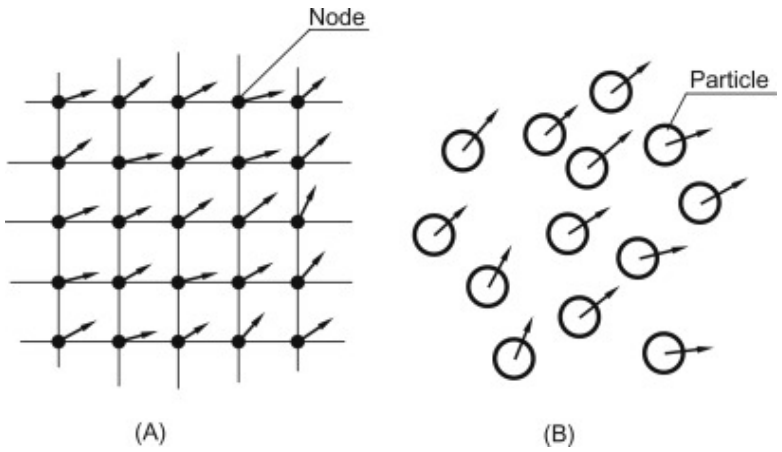


FIGURE 5. *(a) is the eulerian on a mesh description, (b) Lagrangian particle description, graphic from [31]*

In the Lagrangian description, particles are described as points with coordinate vector and velocity vector and perhaps other variables such as temperature and pressure or mass and such. The moving particles' mass is kept constant and the distribution of the mass in the particle is not considered. Meaning that overlapping particles are not considered. Thus mass conservation is simply that no particles are created and destroyed, the number of particles is kept constant. In the Eulerian description, the properties are confined to a mesh grid, meaning their spatial coordinates are constant

4.2 Particle-in-cell with Macro-particles

A combination of Eulerian and Langrangian and one that has had a lot of success and has been used since the '50s is the Particle-in-cell method. It is used to solve a certain class of partial differential equations. Where the particles are tracked in a Langrangian way in continuous phase-space and the densities and such are computed at the same time on Eulerian mesh points.

Particles are thus represented in so-called *macro-particles*. These are ensembles of particles where macro-quantities such as number density are assigned as a weighting. The macro particle can be anywhere in the continuous domain however macro-quantities are only calculated on the discrete mesh points. For a given macro-particle we assume a certain shape, determined by the shape function $S(\boldsymbol{x} - \boldsymbol{x}_p)$, where $\boldsymbol{x}$ is the coordinate of the macro-particle and $\boldsymbol{x}_p$ the observation point.

As our focus is macroscopic trends of evolution in the long term of the particles it makes sense to simplify the many-body problem in this way, as we would like to cut down on computation cost and filling the system with a gigantic number of realistic mass particles is unfortunately

not a feasible way to computationally handle the problem.

This macro-particle is then only allowed to move with an ensemble velocity, such that the shape of the particle is held constant and no deformation occurs. It can be described by kinetic theory via the introduction of a smooth distribution function $f(\boldsymbol{x}, \boldsymbol{v}, t)$ this is then a 6-dimensional problem, given by the Vlasov equation[32]

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \frac{\partial f}{\partial \boldsymbol{x}} + \boldsymbol{a} \cdot \frac{\partial f}{\partial \boldsymbol{v}} = \left.\frac{\partial f}{\partial t}\right|_c . \tag{4.24}$$

Here $\boldsymbol{a}$ is some force, as an example drag-force is an important one in our case. The right-hand side is a collision term describing the transfer of momentum. Considering the dimensionality of the problem and complexity we must find a way to approximate a solution. Firstly the distribution function in our case will be described as a macro-particle given by

$$f_i(\boldsymbol{x}, \boldsymbol{v}, t) = S(\boldsymbol{x} - \boldsymbol{x}_i) P(\boldsymbol{p} - \boldsymbol{p}_i) w_i . \tag{4.25}$$

Here S is the shape function, this decides the geometric extension of the macro-particle. P is the given shape of the particle in momentum space where we would like for all particles within the macro-particle to have same momentum such that there is no deformation meaning $P(\boldsymbol{p} - \boldsymbol{p}_i) = \delta(\boldsymbol{p} - \boldsymbol{p}_i)$. w_i describes the number of physical particles per macro-particle.

The geometric shape of the particle could be constructed in different ways, often involving some symmetry and some smooth function. Often something like a step function is used for its simplicity. We would here like to have the simplest form possible given by the nearest-grid-point scheme where the shape is simply considered to be a delta function[33]

$$S(\boldsymbol{x} - \boldsymbol{x}_i) = \delta(\boldsymbol{x} - \boldsymbol{x}_i) . \tag{4.26}$$

This means that we have basically a linear interpolation for solving the forces on the nearest grid points, as it would only be solved on the grid point it is nearest. With this approach we have more noise as we are not really looking at spreading out the macro-particle, we only evaluate at one point. Other approaches have a smoother transition but with higher-order, spreading the particle out over several points. This however is a very simple implementation that is not very computationally heavy.

4.3 Collisions

Last we discussed how to solve the dynamic side of the Vlasov equation, now we must look at how to handle the collision part of the Vlasov equation, here we must introduce an approximate model as well. This is the essence of this thesis, and we will go into further detail as to how this part is handled in later sections. For now, I will explain the basic principle of the model.

As we for every macro-particle have the mass and the number of particles described, we can then bin these particles into different sizes, such that each size has an associated number of particles. This distribution is then evolved, by looking at what the velocities are and then moving particles around to different sizes in adherence to some model of coagulation and fragmentation. Now we then have a new distribution of particles this can then be brought back to the macro-particle. This is done by simply changing the w_p factor

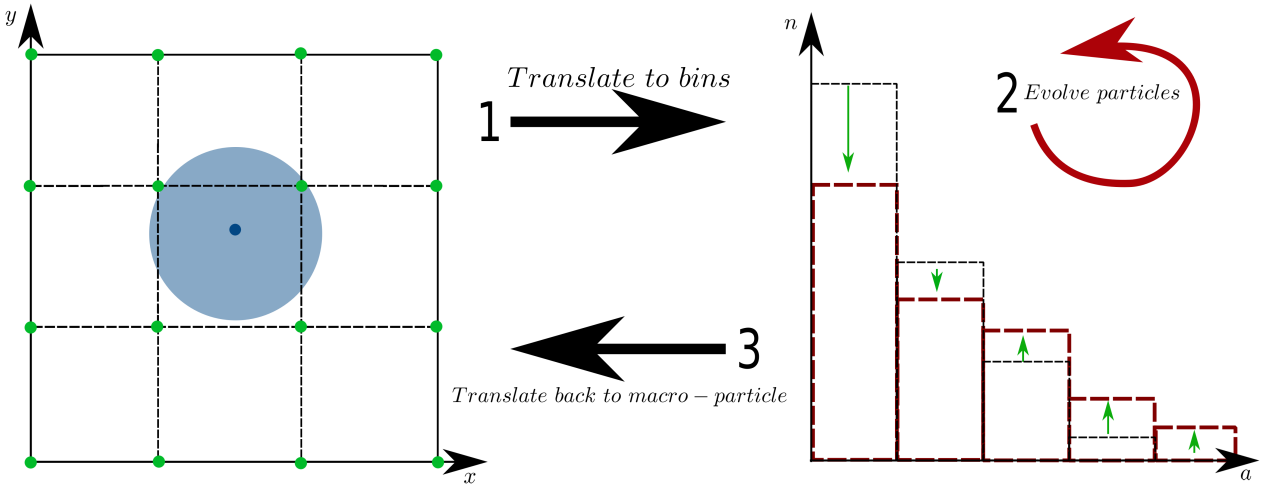


FIGURE 6. *Shown on the left is the PIC scheme with a macro-particle of some shape, here just a representation. Location is given by the blue dot in the center. This is then translated into the binned representation, where the collisions are evaluated in the evolution, this leads to bins being populated or depopulated. This process is then repeated for every timestep.*

5 Dust evolution model

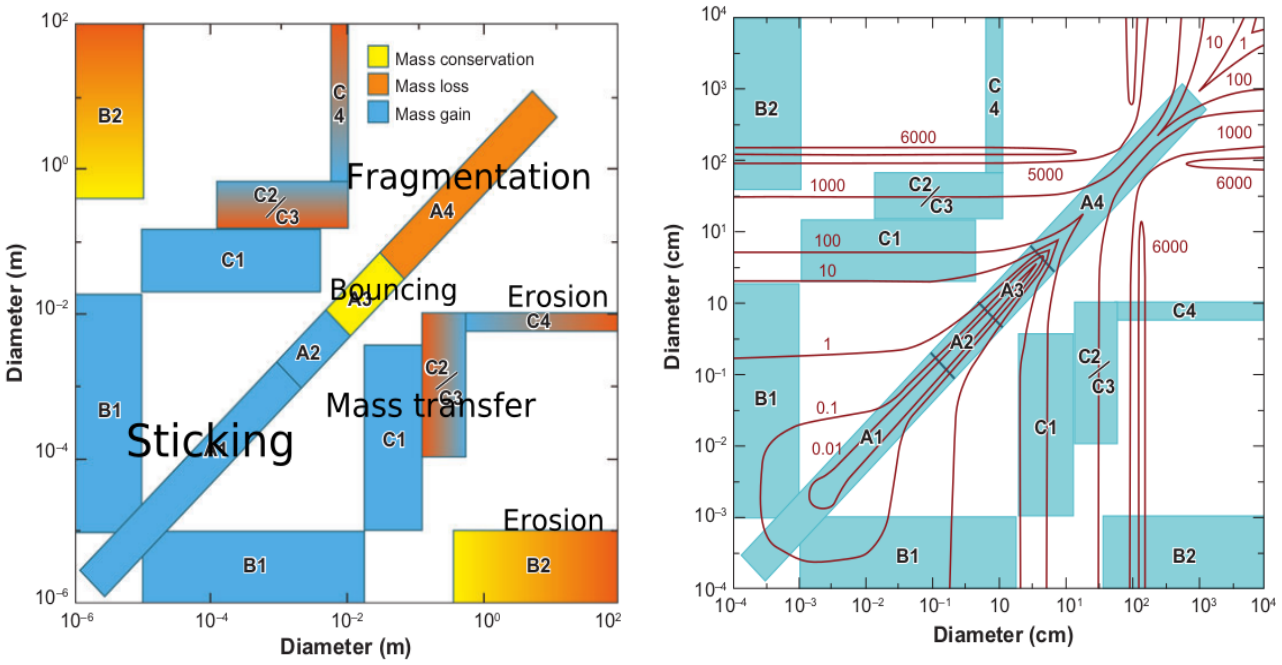
It is still not fully understood how nature processes μm sized particles dust particles to objects like planetesimals and asteroids at km size. With new observations it is becoming increasingly obvious that this is an important question to answer, as it seems very much tied to the dynamics behind planet formation. Several ways of building larger particles have been discussed, such as particles sticking caused by ice in particles outside the ice line of stars or pebble accretion. And several instabilities in protoplanetary disks have been proposed, considering that in the general in the diffuse ISM the dust to gas ratio is often assumed to be 1/100. This ratio is based on observations in the milky way and is often used to assume characteristics of molecular clouds. However this ratio is dependent on the metallicity of the medium and can vary depending on where in the medium one looks, in the outer parts of galaxies, where the ratio can vary greatly[15].

Therefore in general to have dust-dust interactions we must have a local accumulation of dust in regions. This requires instabilities to upset the homogeneity of the medium. For example in protoplanetary disks where dust is shown to accumulate, which happens due to several types of instabilities that can arise in disks. As an example the streaming instability which is based on the initial spontaneous small clumping of dust in a area which in turn slow the radial drift down, meaning faster radial drifting particles will catch up. This leads to a concentration of dust meaning higher dust-to-gas ratio. In general several instabilities arise due to the fact that gas and dust move in different ways. In molecular clouds smaller dust might clump where the gas clumps as these dust particles are well coupled to the gas motion. However larger dust might have other characteristics.

For our model to accurately describe effects as coagulation and fragmentation we must also investigate the zoomed in case of dust particles colliding with one another. The understanding behind this micro-physics of dust grains has been much improved in recent years, it is not easy to study as we cannot simply observe these collisions in space. Therefore we must make laboratory experiments that mimic the right conditions. In recent time several experiments of this kind have been done, using a variety of methods, some using drop towers to mimic the gravity, some have even been done on board the space shuttle in order to make use of the effects of microgravity[26].

Several models of coagulation of particles to bigger aggregates that have been put forth, suggest rapid growth of centimeter-sized particles at only around 1000 years. Simple models that assume perfect sticking do not replenish the reservoir of smaller particles, rendering a protoplanetary disk optically thin after only 1000 years, which is not what we observe. Observations suggests that smaller grains are abundant for millions of years. To fix this the effects of fragmentation has to be included, and this gives a more realistic picture of the outcomes of collisions.

Many different laboratory experiments have been assembled to form figure 7, where the collisions of different grain sizes at different velocities have been analyzed. This was done in using a variety of methods for example using drop towers.



(A) (B)

FIGURE 7. *(a) parameter space of given experiment where the boxes indicate the applicability of the individual experiments to the collision scenario described by Weidenschilling & Cuzzi (1993) of a minimum-mass solar nebula at 1 AU. Shown is the areas where we see 'Sticking' as in two dust particles stick together to form larger aggregates. Here there then is a limit where bouncing is reached, meaning particles bounce of each other and conserve mass. Later comes fragmentation. Note diameter units in [m] (b) is the same plot with contour lines giving the collision velocities in [cm/s]. Note diameter units in [cm] Credit: [26]*

In figure 7 the letters correspond to experiments *Group A*: bigger aggregate dust particles with approximately equal mass are collided. Numerals means higher mass and higher collision velocity (for conditions in a protoplanetary disk). *Group B*: small solid particle (monomer) collide with larger dust aggregate. Numerals means higher collision velocity *Group C*: dust aggregate - dust aggregate collision, numerals are higher collision velocity. It is interesting how there is this clear transition where particles bounce off of each other. We go from sticking to a barrier of bouncing before we reach fragmentation. This gives a nice overview of the possible interactions at given collision velocities and particles sizes. We can use this later to determine how we will handle the question of fragmentation velocity.

To describe dust evolution, we will need to include both coagulation and fragmentation, The Smoluchowski equation is the basic theoretical master equation to do this. It is a population balance equation that describes how the number densities of particles changes over time with interactions. It was first publicized in 1916 by Marian Smoluchowski[34], where it originally only included coagulation as. It was used in statistical physics as way of describing diffusion. This is commonly used equation when investigating grain growth. It is given as

$$\frac{\partial}{\partial t} n(m_k) = \int \int M(m_k, m_i, m_j) n(m_i) n(m_j) dm_i dm_j \ , \tag{5.27}$$

where M is the kernel which describes how particles move in the distribution from one mass to another, this can be any form of kernel one decides. In the case of fragmentation and coagulation it can be defined as

$$\begin{aligned} M(m_k, m_i, m_j) = \frac{1}{2}K(m_i, m_j)\delta(m_i + m_j - m_k) \\ -K(m_i, m_j)\delta(m_j - m_k) \\ +\frac{1}{2}L(m_i, m_j)S(m_k, m_i, m_j) \\ -L(m_i, m_j)\delta(m_k - m_j)\ . \end{aligned} \tag{5.28}$$

Here K is the coagulation kernel, L is the fragmentation kernel and S the distribution of fragments after collision. The first two terms describe the gain and corresponding loss of mass due to coagulation for a specific size. The next two terms describe the fragmentation gain and loss. The distribution of fragments I will go into detail with in 5.2.3. [25]

5.1 Coagulation

Here we want to describe how and under what circumstances dust particles stick to one another. This is described by the coagulation kernel in the Smoluchowski equation, which is given by

$$K(m_i, m_j) = \Delta u_{i,j}\sigma_{geo_{i,j}}p_c\ . \tag{5.29}$$

Here σ_{geo} is the geometrical cross section of the collision, Δu the relative velocity and p_c the probability of coagulation. The cross section is easy enough, it is found by

$$\sigma_{geo_{i,j}} = \pi(a_i a_j)^2\ , \tag{5.30}$$

where a is the radius of the given particles, which are here assumed spherical.

The relative velocities we have talked a bit about already. In the case of *Dispatch*, which is discussed further below, it will be handled in the framework, as there is already a system in place to handle particles.

I will ignore bouncing of particles. Then the probability of coagulation will come from the fact that probability of coagulation and fragmentation must $p_f + p_c = 1$ which means $p_c = 1 - p_f$. I will go into further detail of p_f in 5.2.1.

5.2 Fragmentation

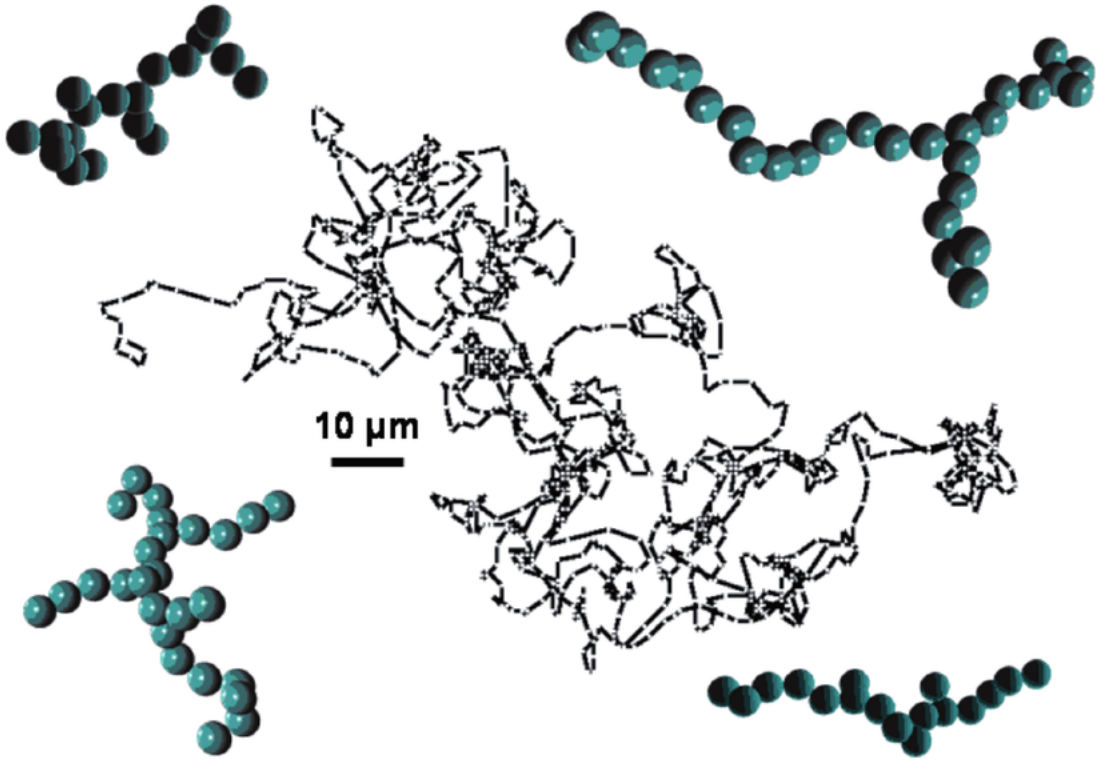


FIGURE 8. *Here a representation of four examples of dust aggregates that are grown by Brownian motion induced collisions. These were reconstructed from 3D microscopic images from space-shuttle experiments. It consists of SiO_2 spheres with 1.9 μm diameter. The centre images shows a trajectory of a single SiO_2 sphere, recorded in a drop-tower experiment. Credit:[35]*

The fragmentation kernel is described by almost the same parameters as the coagulation, except crucially in the probability. Thus

$$L(m_i, m_j) = \Delta u_{i,j} \sigma_{geo_{i,j}} p_f \ , \tag{5.31}$$

where the fragmentation probability is dependent on the relative velocities.

5.2.1 Fragmentation velocity

Exactly when fragmentation happens is a complicated microphysical problem that has to be investigated in the lab or through molecular dynamics modelling. In my models, the probability for fragmentation is controlled by the simplistic function

$$p_f = \begin{cases} 0 & \text{if } \Delta u < u_f - \delta u \\ 1 & \text{if } \Delta u > u_f \\ 1 - \frac{u_f - \Delta u}{\delta u} & \text{else .} \end{cases} \tag{5.32}$$

Here Δu is the velocity difference between the 2 colliding particles and u_f is the velocity at which fragmentation occurs. We also introduce a transition width of δu. Which essentially works like a bouncing parameter, where in the given parameter space the particles neither stick nor fragment, but simply bounce of, of each other. It also provides a parameter space where the

fragmentation probability is being lowered such that far below the bouncing region it goes to zero. This is the approach used in Birnstiel *et al.* [25], which we will later attempt to replicate some test results of. This then allows us to set a specific velocity above which fragmentation will be dominant.

The range at which fragmentation occurs can vary significantly in magnitudes from around 1 m/s [25] to 1 km/s [36]. Thus this fragmentation velocity is subject to some debate. It of course depends on the kinetic energy involved in the collision. However depending also on what type of material the particle is made of whether the particle has been thermally processed and importantly what size of particle we are looking at. Things like the geometric shape of the particles are also shown to be important. There is a lot of micro-physics to consider in grain-grain collisions. There does not seem to be a nice and simple model that captures all the complex components accurately.

Monomers will typically be structurally solid whereas bigger particles might be made up of chains of monomers, forming aggregates that are fluffy, this changes how the collision energy is expended. For example in these aggregates the energy will be spread out between multiple monomers[37]. We also have crystalline structures coming from star forming regions which are thermally processed particles. These are more compact and are held together by sitting in crystal lattice form. Where as these fluffy particles are held together with by simply Van der Waals attraction. Which are comparable to diamonds and the dust accumulating on your shelf respectively. These types will vary significantly in the energy that is needed to break them apart.

With these considerations in mind we can formulate a fragmentation probability function that is dependent on the particles' sizes involved. So that for each possible collision of the bins we have a matching fragmentation velocity we can check against.

If we follow the example of Dominik & Tielens (1997) [38] looking at the kinetic energy involved then the impact energy is given as

$$E = \frac{1}{2} m_\mu (\Delta u)^2 \ . \tag{5.33}$$

Here $m_\mu = m_1 m_2/(m_1 + m_2)$ is the reduced mass of the involved particles and Δu is the collision velocity (The relative velocity between particles). And then using the result from Dominik & Tielens[38] for the breaking energy which is given as

$$E_{br} = A_{br} \frac{\gamma^{5/3} a_\mu^{4/3}}{\mathcal{E}^{\star 2/3}} \ . \tag{5.34}$$

The rolling energy, energy required to roll a contact point between monomers in a aggregate is given as

$$E_{roll} = 6\pi^2 \zeta_{crit} \gamma a_\mu \ , \tag{5.35}$$

where a_μ is the reduced radius of the particles. γ is the surface energy density and $\mathcal{E}^\star$ reduced elastic modulus of the material. The parameter A_{br} and ζ_{crit} in Dominik & Tielens[38] are based on theoretical considerations and assumptions which leads to $A_{br} = 43$ and $\zeta_{crit} = 10^{-8}\ cm$. However later in laboratory experiments by Poppe (1999)[39] it is shown that monomers stick at a order of magnitude higher velocity than assumed previously. This was due to having underestimated the energy dissipation by two orders of magnitude[39]. These factors leads to a much higher A_{br} where I will follow the calculations in [37] where they estimate the parameter

at $A_{br} = 2.8 \times 10^3$. Also the rolling energy E_{crit} is also shown to be an order of magnitude higher than the theory. Meaning that the critical displacement ζ_{crit} is an order of magnitude larger. Here the new estimate is then $\zeta_{crit} = 2 \cdot 10^{-7}\ cm$. From this we can then estimate the breaking velocity or fragmentation velocity. We look at the breaking energy and isolate the velocity such that

$$u_{frag} = \sqrt{\frac{2E_{br}}{m_\mu}} \Rightarrow \tag{5.36}$$

$$u_{frag} = \sqrt{2A_{br}} \frac{\gamma^{5/6} a_\mu^{2/3}}{m_\mu^{1/2} \mathcal{E}^{\star 1/3}} \ . \tag{5.37}$$

Here for silicate grains the material property parameters γ and $\mathcal{E}^\star$ will be set to $\gamma = 25$ erg g^{-2} and $\mathcal{E}^\star = 2.8 \cdot 10^{11}$ dyn g^{-2}. Now we can use this to estimate effects on monomers with the given parameters, such that we find the scaling

$$u_{frag} = \frac{1}{2^{2/3}} \sqrt{\frac{3A_{br}}{\pi}} \gamma^{5/6} \rho_s^{-1/2} a_0^{-5/6} \mathcal{E}^{\star -1/3} \Rightarrow \tag{5.38}$$

$$\begin{aligned} u_{frag} = 8.4\ \frac{m}{s} \left(\frac{A_{br}}{2.8 \cdot 10^3} \right)^{1/2} \left(\frac{\gamma}{25\ \text{erg}\ g^{-2}} \right)^{5/6} \left(\frac{\rho_s}{2\ g\ cm^{-3}} \right)^{-1/2} \\ \left(\frac{a_0}{0.1\ \mu m} \right)^{-5/6} \left(\frac{\mathcal{E}^\star}{3.7 \cdot 10^{11}\ \text{dyn}\ g^{-2}} \right)^{-1/3} \ . \end{aligned} \tag{5.39}$$

Assuming spherical grain such that $m_0 = \frac{4}{3} \rho_s a_0^3$. The subscript '0' describes a monomer. This is then the collision between two equal sized grains. From this we can see that for a 0.1 μm grain the fragmentation velocity will be $9.5 \frac{m}{s}$, for smaller particles as 1 nm it will be $438.6 \frac{m}{s}$ and bigger particles as 1 μm it is $1.4 \frac{m}{s}$. For even bigger particles as 1 mm it will be much lower at 0.4 $\frac{cm}{s}$

This result is indeed only applicable to small particles since in the case of bigger particles as aggregates of monomers, here there is the possibility of distributing the energy over several monomers in the structure. This means we need to look at the rolling energy. By the results of Blum & Wurm (2000)[26] then the regime for sticking is when the energy is below $5E_{roll}$ and above we will have other types of regimes, in Ormel (2008)[37] transition areas of energy is considered, such as restructuring of the aggregate and erosion. Here we will keep it simple and everything above the $5E_{roll}$ limit will give us fragmentation, however these regimes are something that could go into further considerations of the model, also in the case of fragmentation. From this energy threshold however we can again look at the kinetic energy such that

$$u_{frag} = \sqrt{\frac{2(5E_{roll})}{m_\mu}} \Rightarrow \tag{5.40}$$

$$u_{frag} = \sqrt{\frac{60\pi^2 \zeta_{crit} \gamma a_\mu}{m_\mu}} \ . \tag{5.41}$$

Now again we will like to know how this scales in terms of the size of particles. To do this we assume spherical grain and equal size such that $a_\mu = a_0/2$, we get

$$u_{frag} = \sqrt{60}\pi\zeta_{crit}^{1/2}\gamma^{1/2}a_{\mu}^{1/2}\left(\frac{4}{3}\rho_s a_0^3\pi\right)^{1/2} \Rightarrow \tag{5.42}$$

$$u_{frag} = 13.3\frac{m}{s}\left(\frac{\zeta_{crit}}{2\cdot 10^{-7}cm}\right)^{1/2}\left(\frac{\gamma}{25\text{ erg } g^{-2}}\right)^{1/2}\left(\frac{\rho_s}{2\ g\ cm^{-3}}\right)^{-1/2}\left(\frac{a_0}{0.1\ \mu m}\right)^{-1} . \tag{5.43}$$

This is then only applicable to larger aggregates. Clearly this scaling in the size means that for bigger aggregates, the fragmentation velocity will go down quicker than (5.39).

Thus using this new expression for fragmentation velocity we can use the same function as we began with (5.32) now the function is just dependent on the particles colliding as the fragmentation is now dependent on the particles' sizes and material properties, thus

$$p_f(i,j) = \begin{cases} 0 & \text{if } \Delta u_{i,j} < u_{frag_{i,j}} - \delta u \\ 1 & \text{if } \Delta u_{i,j} > u_{frag_{i,j}} \\ 1 - \frac{u_{frag_{i,j}} - \Delta u_{i,j}}{\delta u_{i,j}} & \text{else} \end{cases} \tag{5.44}$$

Now we have a matrix that for every possible collision between bins can give us a probability for fragmentation and a for coagulation as we set $p_c = p_f - 1$. Which means they sum to unity and we either have coagulation or fragmentation in collisions.

In Ossenkopf (1993) it is found to be the turbulence that is the dominant force behind fragmentation, except for high density and small particles. Turbulence causes eddies which can given high relative velocities, which are required for fragmentation[40]. This is also clear from the results shown later.

5.2.2 Cratering

Since previous studies have shown that large dust particles tend to acquire large velocity dispersion they therefore also tend to fragment. This is also seen in warm ionized mediums[41].Therefore we introduce 2 regimes of fragmentation, depending on the masses of the dust particles involved in the collision. We want the possibility of cratering to provide the system with smaller particles. This could occur when one big particle and a small particle collide at high speeds. Thus the smaller particle will dig into the bigger particle. First of we need to determine in which specific collisions this would occur. We would here like to introduce a basic model, so what we will do is assume a specific ratio between the masses such that when this is small or large, corresponding to when large and small particles collide, we have cratering

When

$$\frac{m_i}{m_j} > 10 ,$$

or

$$\frac{m_i}{m_j} < 0.1 .$$

Which follow the same example as [42], where masses that differ by order magnitude is assumed to give cratering. Shown in Figure 9 are the possible interactions.

As shown when cratering occurs the smaller particle will dig out material equal to its own mass thereby giving us $2m_j$ mass, here assuming $i > j$. This mass is spread into fragments. Where as in the other case the total mass of the particles will be spread into fragments. This results in a efficient way of limiting the particle sizes and providing small dust grains.

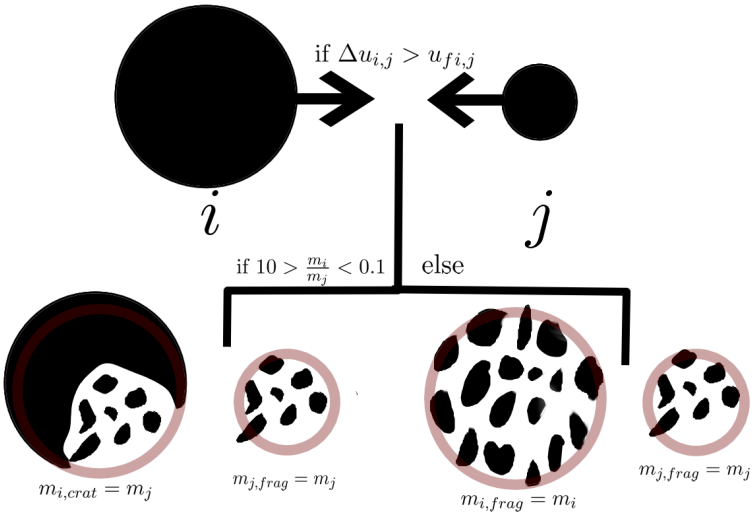


FIGURE 9. *Here we have the two colliding particles. If their relative velocities are above that of the fragmentation velocity u_f one of two scenarios will play out depending on the mass ratio of the two. Where in the first scenario the smaller particle will make a crater in the bigger particle effectively resulting in a fragmentation mass equal to $2m_j$, and leaving the rest of the larger particle. In the second scenario full fragmentation will occur shattering both particles, feeding the system with small particles.*

5.2.3 Distribution of fragments

How the fragments are spread between mass bins is typically described by a power-law. Such that most fragments are of smaller masses. This means that in (5.28) we here need to formulate S which is responsible for distributing fragments after collisions.

Now that we have 2 cases of fragmentation, one that includes cratering. We will similarly handle this distributing in two separate ways. For full fragmentation we want the fragments to be spread across all the mass bins below the largest one that is included in the collision. As we have binned the masses the only thing we need to find is which of the indexes are the biggest in the collision. Thus

$$S(k,i,j) = m_k^{-\xi} \text{ where } k < i \text{ and } i > j, \text{ or where } k < j \text{ and } j > i, \text{ or where } k < i, j \text{ and } i = j \ , \tag{5.45}$$

an easier way of saying this is

$$S(k,i,j) = m_k^{-\xi} \text{ where } k < \max(i,j) \ . \tag{5.46}$$

This gives us an array of the length given by the index of the largest particle in the collision. Thus spreading the fragments across the bin indexes below this. This of course does not include the index of the large particle itself.

In the case of cratering the particles dug out of the larger particle must not exceed the smaller particle in mass, thus we will distribute the fragments in all the bins below the smaller particle in the collision.

$$S(k,i,j) = m_k^{-\xi} \text{ where } k < i, j \ , \tag{5.47}$$

where we will also have to re-normalize the array such that mass is preserved.

The exponent for the power-law has been widely discussed. It ranges between 1 and 2, both experimentally and theoretically[25]. I will here follow the lead of Brauer, Dullemond (2008)[42] and use $\xi = 1.83$.

5.3 Extending the Smoluchowski equation to include momentum

My dust evolution kernel will be integrated into *Dispatch* which has a built-in particle system, . It has to conserve mass and particle number densities, but it also needs to conserve momentum, and in that case we need to be able to transfer momentum with particles that coagulate and fragment. Luckily in the Smoluchowski equation we have a system in place for transfer of mass, so all we need is to have the velocity of each particle, so that in each collision we can also make sure to transfer the momentum to fragments or coagulated particles.

$$\frac{\partial}{\partial t}\boldsymbol{q}(m_k) = \int\int M(m_k, m_i, m_j) n(m_i) n(m_j) \frac{m_i \boldsymbol{v}_i + m_j \boldsymbol{v}_j}{m_i + m_j} dm_i dm_j \ , \tag{5.48}$$

where q is momentum number density can be related to momentum density as $p_k = q_k m_k$. Note that this is in principle only valid given that the individual particle inside a macro-particle are assumed to have the same velocity.

5.4 Analytical solution to Brownian motion

By assuming that the grain size distribution to be a delta function and the coagulation probability to be unity we can approximate the growth rate of particles in collisions. The increase in mass is given by the mass of the particle m divided by the collision time τ

$$\frac{dm}{dt} = \frac{m}{\tau} \ . \tag{5.49}$$

Using that $dm = 4\pi\rho_s a^2 da$ and that $\tau = m/(\rho_d \sigma \Delta u)$, where ρ_d is the dust density, ρ_s is the solid density of a particle, σ the cross section given as $\sigma = 4\pi a^2$ for spherical particles and Δu the relative velocity between particles. If we assume the relative velocity to be given by the Brownian motion as [25]

$$\Delta u_{BM} = \sqrt{\frac{16 k_b T}{\pi m}} \ , \tag{5.50}$$

then integrating (5.49)

$$a(t) = \left(\frac{5}{2} \frac{\rho_\mathrm{d}}{\rho_\mathrm{s} \pi} \sqrt{\frac{12 k_\mathrm{b} T}{\rho_\mathrm{s}}} (t - t_0) + a_0^{\frac{5}{2}} \right)^{\frac{2}{5}} \ . \tag{5.51}$$

This gives us a analytical solution to the grain growth in the simplified case which we can test our model on as a first step. We must first make some assumptions about the disk however. If we assume the vertical structure to be in hydrostatic equilibrium at all times, and if the protoplanetary disk is geometrically thin meaning the pressure scale height over radius $H_p(r)/r \ll r$ and the vertical sound crossing time is much shorter than the radial drift time scale of the gas. The isothermal vertical density structure is then given by

$$\rho(z) = \rho_0 \exp(-\frac{z^2}{2H_p^2}) \ , \tag{5.52}$$

where

$$\rho_0 = \frac{\Sigma_g}{\sqrt{2\pi} H_p} \ , \tag{5.53}$$

$$H_p = \frac{c_s}{\Omega_k} \ , \tag{5.54}$$

$$\Omega_k = \sqrt{\frac{G m_{star}}{r^3}} \ . \tag{5.55}$$

Here m_{star} is the mass of the host star, and G is the gravitational constant. If we make further assumptions, as the temperature in the disk is $T \sim 200\ K$, the radius of the disk is $r = 1\ AU$, the stars mass is half that of the sun and we can use the ideal gas law to simplify the sound speed to

$$c_s = \sqrt{\frac{k_b T}{\mu m_H}} \ . \tag{5.56}$$

Here k_b is the Boltzmann constant, μ is the ratio between hydrogen and other atoms, primarily Helium, and m_H is the mass of a hydrogen atom. We will assume $\mu = 2.37$, leading to a sound speed of $c_s = 823\ m/s$, and assume a dust-to-gas ratio of 1/100 such that $\Sigma_d = \Sigma_g/100$

6 Numerical Algorithm for dust evolution

The essence of our problem, in this case, is the discretization of the mass, and the particle size. We will do this by having an upper and lower bound for the size of the particles. Between these two points, we will then have a logarithmic space with a set number of bins. Thus slicing the particle size space. Even though their physical shape will not have an effect on the simulations, as we do not take this into account in the way we describe the microphysics of dust grain collisions, we will assume spherical dust grains meaning we can relate the particle mass and particle size as

$$m = \frac{4}{3}\rho_s a^3 \pi \ .$$

Where ρ_s is the density of the particles. We will throughout this thesis use $\rho_s = 2$ g/cm for all dust particles.

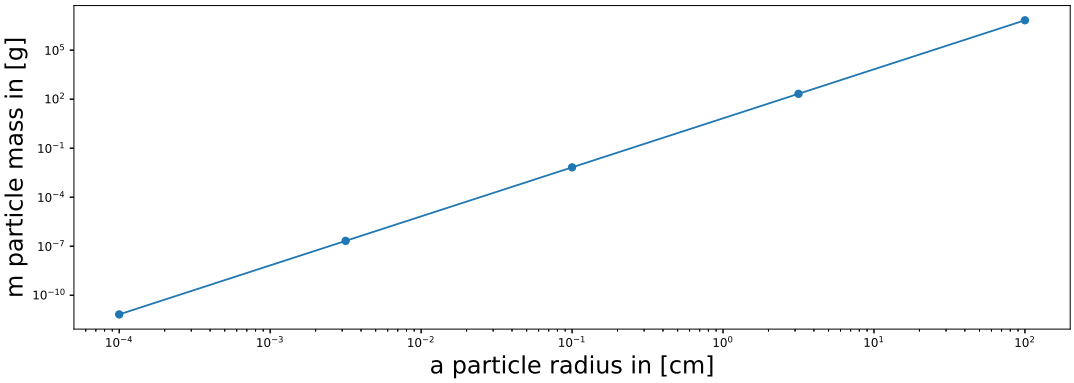


FIGURE 10. *Here in simple terms the binning that we will be using is shown, the sizes and masses are discrete. In this example we use 5 bins and the sizes ranging from* 10^{-4} *[cm] to* 10^{2}*[cm] this is then the limit of our system.*

6.1 Podolak algorithm: conservation of number density and mass

With equation (5.27) we describe how the particle number density evolves with time, in order to solve this numerically, we will have to discretize it to solve the equation at discrete points. This presents problems when a particle mass lands between 2 mass bins, here we must have some way of splitting the mass. This was previously done by Podolak (1980)[43] for investigating dust grain growth and was previously used in meteorological science. It has similarly been used in other works such as Brauer (2008)[42] for studying dust grain growth.

What we require for our dust model first and foremost is to be able to conserve mass and particle number density. This is exactly what the Podolak algorithm achieves and the algorithm is relatively simple to implement in any system. This makes it suitable for our intended purpose in handling.

As shown above we have discretized the sizes of particles and thereby the masses. We can also use this to have an associated number density for each grid point. Such that for a grid point m_i has a number density n_i associated. This means there is a specific number of particles n_i of the given size a_i, where each particle has mass m_i. Thus we bin all particles of a certain

size. Particles from bin i and bin j will then have a coagulation rate from i to j per time is given by

$$Q_{ij} = n_i n_j K_{ij} \ . \tag{6.57}$$

Here K_{ij} is the coagulation kernel also recognized in the Smoluchowski equation. It is given as

$$K_{ij} = \Delta u_{ij} \sigma_{geo.i,j} p_c \ . \tag{6.58}$$

Since we are using a logarithmic space for the mass binning we have put ourselves in the situation where when particle i and j coagulate together their mass $m = m_i + m_j$ will give us a point outside our mass grid bins, it will never actually hit a point on the discrete mass grid. Thus we must divide the interaction between the two nearest bins, in a way that makes physical sense. To split the mass that falls between two grid points we look at the two neighbor bins $m_m < m < m_n$, and we split the coagulation rate linearly between these two such that

$$\begin{aligned} Q_m &= \epsilon Q_{ij}, \\ Q_n &= (1-\epsilon) Q_{ij} \ . \end{aligned} \tag{6.59}$$

Here the number density will be conserved since $Q_m + Q_n = Q_{ij}$, now the key thing we need to ensure is that we also have mass conservation. This we can enforce by setting

$$Q_m m_m + Q_n m_n = Q_{ij}(m_i + m_j) \ , \tag{6.60}$$

which we can in turn use to define our parameter ϵ such that

$$\epsilon = \frac{m_n - (m_i + m_j)}{m_n - m_m} \ . \tag{6.61}$$

This can then be used to define every possible coagulation interaction between particles in bin i and j in a coefficient defined as

$$C_{ijk} = \begin{cases} \epsilon & \text{if } m_k \text{ is the largest mass grid point} < m_i + m_j \\ 1-\epsilon & \text{if } m_k \text{ is the smallest mass grid point} > m_i + m_j \\ 0 & \text{else} \ . \end{cases} \tag{6.62}$$

Replacing the integrals in the Smoluchowski equation with sums over discrete mass bins, coagulation, we now have the following for evolving the particle number density with time

$$\dot{n}_k = \frac{1}{2} \sum_{ij} Q_{ij} C_{ijk} - \sum_i Q_{ik} \Rightarrow \tag{6.63}$$

$$\dot{n}_k = \frac{1}{2} \sum_{ij} K_{ij} C_{ijk} n_i n_j - \sum_i K_{ik} n_i n_k \ . \tag{6.64}$$

Other problems also arise when we consider the bins located at the far end of the mass grid. Here we might have masses that go over the grid limit, due to coagulation. A roundabout way to not have this happen is simply to make the grid large enough that coagulation will not reach the maximum mass bin and/or simply not populate larger bins from the start. However since we interested in evolution, meaning large bins will not be populated, and this is would also only be a concern if coagulation was present, in our case we have the opposite effect of fragmentation. This will break the larger particles, mostly preventing larger bins from being

significantly populated. However, in areas where fragmentation is less prevalent, we might run into trouble. Therefore we will for safety, implement that for all collisions that give us coagulation that would put the particle mass to the right of the largest bin we will set $\sigma_{geo} = 0$ for these specific collisions, thus everything in the coagulation kernel will go to zero and the collision will in essence not happen.

At this point we must look at the change over time for the system, we have set everything up to be able to handle various transfers of mass and conserve it as well. Now we must evolve in time. In (6.64) we have everything related to coagulation now we just have to combine it with the third and fourth term in (5.28) related to fragmentation giving the additional terms

$$\dot{n}_k = \frac{1}{2}\sum_{ij}(K_{ij}C_{ijk}n_i n_j + L_{ij}S_{ijk}n_i n_j) - \sum_i (K_{ik}n_i n_k + L_{ik}n_i n_k) \ . \tag{6.65}$$

Thus we have our time-dependent differential equation. To solve this on the discrete grid we employ two different methods, as they both have their advantages and disadvantages, depending on what you value most, computational speed or accuracy. Luckily much of this equation is something that does not have to be calculated every timestep. Much of this can be calculated beforehand, such as the coefficients C and S. In reality, only the probabilities in our case have to be re-evaluated at each timestep but more on that below.

6.2 Explicit method

The simplest way of solving (6.65) is by using the explicit method. Explicit means to evaluate the system at a later time given the system at the current time. Meaning

$$\frac{n_k(t+\Delta t) - n_k(t)}{\Delta t} = \frac{\Delta n_k}{\Delta t} \Rightarrow \tag{6.66}$$

$$n_k(t+\Delta t) = n_k(t) + \Delta n_k \tag{6.67}$$

$$\Delta n_k = \left(\frac{1}{2}\sum_{ij}(K_{ij}C_{ijk}n_i n_j + L_{ij}S_{ijk}n_i n_j) - \sum_i (K_{ik}n_i n_k + L_{ik}n_i n_k)\right)\Delta t \ . \tag{6.68}$$

So explicit does not take into account the future system when it solves the equation, thus the only way of keeping the system from making too big changes would be to limit the timestep Δt. For our system, this could occur in regions with high velocities or in regions with high turbulence, where fragmentation could make significant changes to the system in a short time. This can give rise to more severe issues, beyond accuracy. If we were to break conservation of mass and momentum, or at least not maintain it as well as we would like, which is a likely scenario, then maintenance of the system is no longer guaranteed. Luckily we can check for this at runtime, to see whether the mass going in matches the mass going out of the evolution process. We can make checks at different Δt's to see how small we would need to go before mass conservation is guaranteed.

However, at some point, this is no longer practical and we would like a method that enables us to have higher accuracy in scenarios with large Δt.

6.2.1 Solving momentum explicitly

We need momentum conservation for our model of coagulation, to be able to translate from the binned description to a particle description. To do this we must keep track of the average velocities for each size bin, such that bin i has an average velocity $\boldsymbol{v}_i$, in each physical direction. Now to maintain conservation of momentum, we must at all times have that the total momentum of the entire system before the evolution step of coagulation and fragmentation be the same after. To achieve this we must have the same principle for every collision, such that the momentum before a collision is the same as after a collision. If we look at the first term of Eq. (6.68)

$$\sum_{ij}(K_{ij}C_{ijk} + L_{ij}S_{ijk})n_i n_j \ .$$

This term describes the creation of particles, for a given number density bin, given by the coagulation and fragmentation for a collision. We can reformulate this to find out how much momentum these new particles will have by weighing by looking at what the momentum in the colliding bins were before collision and then weighing this by the mass of the size bins, such that the change in velocity number density is

$$\sum_{ij}(K_{ij}C_{ijk} + L_{ij}S_{ijk})n_i n_j \frac{m_i\boldsymbol{v}_i + m_j\boldsymbol{v}_j}{m_i + m_j} \ .$$

This thus determines the momentum for the new particles created, given by the momentum of the old particles. For the second term of Eq. (6.68) we have the term that describes the removal of particles from the system, after they have been coagulated or fragmented. In this case, looking at the momentum then the momentum removed from the system is given by the momentum of the particles removed from the system, such that the removal of momentum is given by

$$\sum_{i}(K_{ik} + L_{ik})n_i n_k \boldsymbol{v}_k \ . \tag{6.69}$$

This means that the explicit solution to the momentum is given by

$$\Delta\boldsymbol{q}_k = \left(\frac{1}{2}\sum_{ij}(K_{ij}C_{ijk} + L_{ij}S_{ijk})n_i n_j \frac{m_i\boldsymbol{v}_i + m_j\boldsymbol{v}_j}{m_i + m_j} - \sum_{i}(K_{ik} + L_{ik})n_i n_k \boldsymbol{v}_k\right)\Delta t \ . \tag{6.70}$$

Alternatively

$$\Delta\boldsymbol{q}_k = \left(\frac{1}{2}\sum_{ij}(K_{ij}C_{ijk} + L_{ij}S_{ijk})\frac{n_j m_i\boldsymbol{q}_i + n_i m_j\boldsymbol{q}_j}{m_i + m_j} - \sum_{i}(K_{ik} + L_{ik})n_i \boldsymbol{q}_k\right)\Delta t \ . \tag{6.71}$$

This can then be solved separately in all directions x, y and z. Meaning we are now for every timestep solving 4 equations. One for number density and 3 for momentum.

6.3 Implicit method

Also as I am implementing this into a bigger framework that handles various other physics (MHD, gravity, etc.) which runs its own timestep that fits the scenarios of large-scale physics.

These timescales in terms of dust evolution can be very large. This is another reason we would want the possibility of doing implicit steps such that we can do any size of timesteps and still maintain conservations.

An implicit method involves solving the equation involving both the current state of the system and the later state. This makes the solution much more involved and does cost extra computation time. However in many situations where arbitrarily large Δt is required it will be beneficial.

Now the numerical change in n_k in a given change in time $\Delta t = t_{i+1} - t_i$, would be given by $\Delta n_k = n_k^{i+1} - n_k^i$ where n_k^{i+1} is the number density at a later time t_{i+1} and n_k^i is the system at current t_i. This we can use in our dicretized equation (6.65), such that

$$\begin{aligned}\frac{\Delta n_k}{\Delta t} = \frac{1}{2}\sum_{ij}(K_{ij}C_{ijk} + L_{ij}S_{ijk})(n_i + \Delta n_i)(n_j + \Delta n_j) \\ - \sum_i (K_{ik} + L_{ik})(n_i + \Delta n_i)(n_k + \Delta n_k) \ ,\end{aligned} \tag{6.72}$$

giving

$$\begin{aligned}\frac{\Delta n_k}{\Delta t} = \frac{1}{2}\sum_{ij}(K_{ij}C_{ijk} + L_{ij}S_{ijk})(n_i n_j + n_i\Delta n_j + n_j\Delta n_i + \cancel{\Delta n_i \Delta n_j}) \\ - \sum_i (K_{ik} + L_{ik})(n_i n_k + n_i\Delta n_k + n_k\Delta n_i + \cancel{\Delta n_i \Delta n_k}) \ .\end{aligned} \tag{6.73}$$

Omitting second-order terms, as they are very small, we are left with the same starting terms $n_i n_j$ and some extra terms. We will define

$$A_k := \frac{1}{2}\sum_{ij}(K_{ij}C_{ijk} + L_{ij}S_{ijk})(n_i n_j) - \sum_i (K_{ik} + L_{ik})(n_i n_k) \ , \tag{6.74}$$

and

$$W_{ijk} := K_{ij}C_{ijk} + L_{ij}S_{ijk} \ . \tag{6.75}$$

Here A_k is the explicit source term, as it is just the explicit solution. Simplifying (6.73) to

$$\frac{\Delta n_k}{\Delta t} = A_k + \frac{1}{2}\sum_{ij}(W_{ijk} + W_{jik})n_j\Delta n_i - \sum_i (K_{ik} + L_{ik})(n_i\Delta n_k + n_k\Delta n_i) \ . \tag{6.76}$$

Now we can gather the rest of the terms in a matrix of the form

$$J_{ik} := \frac{1}{2}\sum_j (W_{ijk} + W_{jik})n_j - (K_{ik} + L_{ik})n_k \ , \tag{6.77}$$

$$J_{kk} := -\sum_i (K_{ik} + L_{ik})n_i \ . \tag{6.78}$$

Thus if we write this with vectors $\mathbf{n}$ and $\Delta\mathbf{n}$ then we can write this in the complete matrix form as

$$\left(\frac{\mathbb{1}}{\Delta t} - \mathbf{J}\right)\Delta\mathbf{n} = \mathbf{A} \ . \tag{6.79}$$

Here $\mathbf{J}$ is the Jacobian of the source function and $\mathbb{1}$ is the unity matrix. Now from (6.79) we can find the solution to the system at a later time as the difference between the solutions at current and later times, as the solutions for each bin are independent, the solutions can be written simply by inverting the matrix such that

$$\mathbf{n}^{i+1} = \mathbf{n}^i + \Delta\mathbf{n} = \mathbf{n}^i + \left(\frac{\mathbb{1}}{\Delta t} - \mathbf{J}\right)^{-1} \mathbf{A} \ . \tag{6.80}$$

What we get is a solution to the eq. (6.65) that is more consistent in time and goes asymptotically to the right solutions. Since using this method requires inverting a matrix it can potentially be quite computationally heavy.

6.3.1 Solving momentum implicitly

If we use the same approach as before looking at the numerical change for a timestep $\Delta t = t_{i+1} - t_i$, the change in momentum would be given by $\Delta \boldsymbol{q}_k = \boldsymbol{q}_k^{i+1} - \boldsymbol{q}_k^i$. Where $\boldsymbol{q}_k^{i+1}$ is the momentum at a later time t_{i+1} and $\boldsymbol{q}_k^i$ is the system at current t_i. Since the number density also changes in time we also have that $\Delta n_k = n_k^{i+1} - n_k^i$. This leads to the change given by Eq. (6.71) and (6.65) as

$$\begin{aligned}\frac{\Delta \boldsymbol{q}_k}{\Delta t} = \frac{1}{2}\sum_{ij}(K_{ij}C_{ijk} + L_{ij}S_{ijk}) \left[\frac{m_i(\boldsymbol{q}_i + \Delta\boldsymbol{q}_i)(n_j + \Delta n_j) + m_j(\boldsymbol{q}_j + \Delta\boldsymbol{q}_j)(n_i + \Delta n_i)}{m_i + m_j}\right] \\ - \sum_i (K_{ik} + L_{ik})(n_i + \Delta n_i)(\boldsymbol{q}_k + \Delta\boldsymbol{q}_k) \ .\end{aligned} \tag{6.81}$$

Then expanding this

$$\begin{aligned}\frac{\Delta \boldsymbol{q}_k}{\Delta t} = \frac{1}{2}\sum_{ij}(K_{ij}C_{ijk} + L_{ij}S_{ijk}) \Bigg[&\frac{m_i\boldsymbol{q}_i n_j + m_j\boldsymbol{q}_j n_i}{m_i + m_j} + \\ &\frac{m_i}{m_i + m_j}(\boldsymbol{q}_i\Delta n_j + \Delta\boldsymbol{q}_i n_j + \cancel{\Delta\boldsymbol{q}_i\Delta n_j}) \\ &+\frac{m_j}{m_i + m_j}(\boldsymbol{q}_j\Delta n_i + \Delta\boldsymbol{q}_j n_i + \cancel{\Delta\boldsymbol{q}_j\Delta n_i}) \\ - \sum_i (K_{ik} + L_{ik})(n_i\boldsymbol{q}_k + \Delta n_i\boldsymbol{q}_k + n_i\Delta\boldsymbol{q}_k + \cancel{\Delta n_i\Delta\boldsymbol{q}_k}) \ .\end{aligned} \tag{6.82}$$

It is thus clear that q is coupled to n, as the change in p will depend upon the change in n. However n does not depend on q. If we again omit second order terms, we can split these terms into matrix form such that we first have a source matrix given as

$$A_k^p := \frac{1}{2}\sum_{ij}(K_{ij}C_{ijk} + L_{ij}S_{ijk})\frac{m_i\boldsymbol{q}_i n_j + m_j\boldsymbol{q}_j n_i}{m_i + m_j} - \sum_i (K_{ik} + L_{ik})n_i\boldsymbol{q}_k \ . \tag{6.83}$$

Again using W as defined in Eq. (6.75) we can define

$$J_{ik}^p := \frac{1}{2}\sum_j (W_{ijk} + W_{jik})\frac{m_j n_j}{m_i + m_j} \ , \tag{6.84}$$

and

$$J^p_{kk} := -\sum_i (K_{ik} + L_{ik}) n_i \ . \tag{6.85}$$

Lastly we have the terms that describe the coupling between momentum and the change in n

$$M^p_{ik} = \frac{1}{2}\sum_j (W_{ijk} + W_{jik}) \frac{\boldsymbol{q}_j m_j}{m_i + m_j} - \sum_i (K_{ik} + L_{ik}) \boldsymbol{q}_k \ . \tag{6.86}$$

Now we have every part of the system describe by the matrices. We can now use vectors $\mathbf{n}, \Delta\mathbf{n}, \mathbf{q}, \Delta\mathbf{q}$ to describe the full coupled system, it will be given by the equation

$$\left(\begin{array}{c|c} \frac{\mathbb{1}}{\Delta t} - \mathbf{J} & \mathbf{0} \\ \hline \mathbf{M}^p & \frac{\mathbb{1}}{\Delta t} - \mathbf{J}^p \end{array}\right) \begin{pmatrix} \Delta\mathbf{n} \\ \Delta\mathbf{q} \end{pmatrix} = \begin{pmatrix} \mathbf{A} \\ \mathbf{A}^p \end{pmatrix} \ .$$

From this it is clear that to solve the momentum part we must use the solution of the number density. As we have already gone through the change in number density is then given by

$$\left(\frac{\mathbb{1}}{\Delta t} - \mathbf{J}\right)^{-1} \mathbf{A} = \Delta\mathbf{n} \ , \tag{6.88}$$

the next terms given by the matrix multiplication would be

$$\mathbf{M}^p \Delta\mathbf{n} + \left(\frac{\mathbb{1}}{\Delta t} - \mathbf{J}^p\right) \Delta\mathbf{q} = \mathbf{A}^p \ , \tag{6.89}$$

isolating the change in momentum we get

$$(\mathbf{A}^p - \mathbf{M}^p \Delta\mathbf{n}) \left(\frac{\mathbb{1}}{\Delta t} - \mathbf{J}^p\right)^{-1} = \Delta\mathbf{q} \ . \tag{6.90}$$

However this only solves the momentum for one direction, we must do this in all directions giving us three systems of equations for momentum. In total we have four systems of equations to solve that involves inverting a matrix. Each system is $nbin \times nbin$ in size. We note that by first solving for the number density and then for the three momenta, we have elegantly reduced the size of the matrix by a factor of four at the expense of solving four systems with similar coefficients.

7 The Dispatch framework

In order to use our model of dust evolution in the context of molecular clouds or planet formation, we need a framework that can handle a setup with all the underlying physics of such a system. Crucially we must have setup of gas and dust that can evolve in time. This will require solutions to equations of Hydrodynamics (HD), and a system for handling particles. Here the *Dispatch* framework [44] is used.

7.1 Ideas and principles

Dispatch is a code framework, written in *Fortran*, that is made for high-performance computing and is used to simulate large-scale physics confined in a box. It has some key features that allows it to be highly scaleable and highly modular in terms of solvers and what physics to implement. Meaning it is relatively easy to add a system to handle dust evolution on top. It is designed to run in a highly optimized way, allowing us good resolution in simulations.

The basic idea of the framework is that the computational domain is decomposed into small semi-independent patches, allowing asynchronous timsteps and it has its physics solvers sitting in the middle of it all and this allows the solvers to only care about 2 things.

- It figures out the time step, taking local variables into account
- It will then update the local variables of its given datacube. This means there is no need for parallelization in the solver as it does not care about any other pieces of data.

The framework supports many different solvers such as; HD, MHD, RMHD, non-ideal MHD, and/or particle-in-cell, that can be used at will. Since no implementation of Message-Passing-Interface (MPI), for implementation on clusters, or multiprocessing, for example, Open-multiprocessing (openMP) is required in the solver, this makes it easy to implement new solvers to the framework.

Solvers are ported from well known well documented astrophysical fluid codes and are tested. From the RAMSES code the Godunov-type Riemann solver, the HLLC solver. This solver solves the Euler equations of motions.

Dispatch introduces a mesh-based system where the computational space is split into problems of cache-friendly and easily vectorized chunk sizes. It is also GPU-friendly task sizes such that many can be updated simultaneously. These tasks can be run asynchronously and semi-independently. This gives the ideal conditions for obtaining optimized solver speed. The mesh space can be either cartesian or orthogonal curvilinear, tasks do not have to be grid-based. Patches can be moving or stationary and mesh can be dynamically refined or statically.

FIGURE 11. *The architecture of the framework is such that the solvers sit in the middle and only handle the specific task of solving the physics in its given task area, everything else is then services functions around the solver. The solvers do not call any interface routines. Credit: Åke Nordlund*

The motivation for building the framework has come about to reduce the computation cost of a given problem, such that it is possible to run experiments on smaller setups or simply increase the resolution of experiments while maintaining the same computation time.

The framework is built on the same idea as many other traditional frameworks of the same purpose with a mesh grid, meaning splitting the problem up. The mesh is divided into patches which in turn are split into cells within each patch.

Increasing the spatial resolution in a mesh will always result in a cost increase, scaling with the smallest resolution element's inverse fourth-order power. This scaling is, however, somewhat mitigated in *Dispatch* by using local time steps.

To setup an experiment the fortran "namelist" format is used. This namelist file is used as an input to set parameters for the different solvers that the user wants to include. This input file also decides the size of the experiment and scaling parameters. It is also possible to start different tasks at different times such that the gas turbulence could evolve from the start and then at a later time particles can be enabled.

7.2 Task-based computing

Traditionally in a mesh based approach the space is split into large equally sized chunks which are then split further into pieces that are split across available compute nodes. This means that the timesteps must be globally communicated as the smaller pieces must wait for the larger chunks, as the structure is of a hierarchical form.

In *Dispatch* it is handled in a way that is more dynamic, as the space is split into many smaller only loosely coupled, only locally interdependent tasks, if the system is grid based these are called *patches*. These patches will then have to communicate with their neighbouring patches as there is a dependence between tasks. For a given patch if its neighbouring patches have advanced sufficiently in time. The evaluation is then done using a ready-to-update queue

or a dynamic neighbor list, where the patch is put into queue if all of its neighbors are up to date. Thus the first in queue is first to be updated. This queue is serviced by *openMP* threads, where a thread computes a patch, meaning we can have simultaneous updates. However there may be critical regions where one patch depends upon another and therefore cannot be updated simultaneously, these regions are therefore assigned to a single threads instead.

With this we achieve ideal conditions for optimizing solvers as the problems are divided into cache-friendly sizes. Also a framework with great scalability. And as there no longer is need of global communications as we have a framework where each patch can have its own timestep based upon local variables, and are no longer waiting for larger parts finishing.

7.3 Guard zones

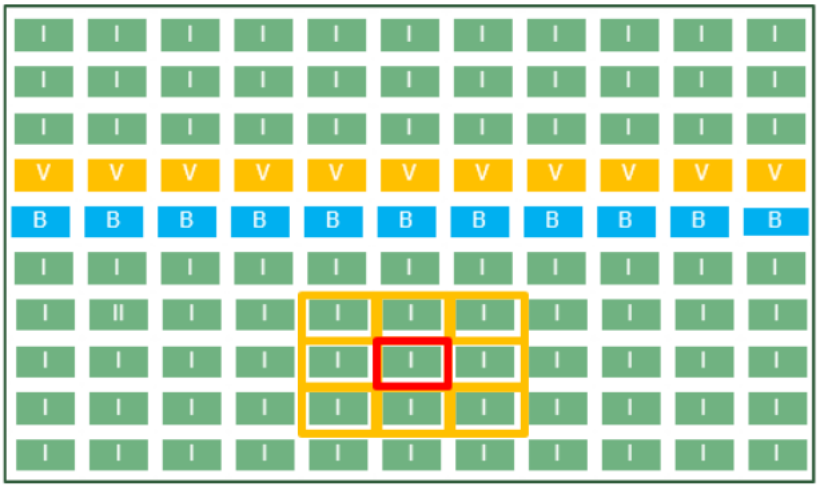


FIGURE 12. *Internal patches labelled "I", virtual patches labelled "V" and boundary patches labelled "B". Marked in red is a patch and yellow is its neighbours. Credit: Åke Nordlund*

Traditionally in a mesh based approach the space is split into large equally sized chunks which are then split further into pieces that are split across available compute nodes. This means that the timesteps must be globally communicated as the smaller pieces must wait for the larger chunks, as the structure is of a hierarchical form. In *Dispatch* it is handled in a way that is more dynamic, as the space is split into many smaller only loosely coupled, only locally interdependent tasks, if the system is grid based these segments are called "patches". These patches are further split into very small chunks called cells. When dividing space into patches as done in *Dispatch*, one is bound to run into trouble solving physics at the boundaries of the patches, as every patch has its own time. What one must therefore do is implement a way to make sure that a patch and its neighbors exist in the same time before trying to do physics across boundaries. A patch will in 2D have 8 neighbours and 26 in 3D, from these neighbours we fetch something called Guard zones. These are an overlap of cells 2-3 cells thick. You will then have a number of relevant cells, depending on how the neighbour is situated. As seen in figure 12 you can have neighbours on the corners, neighbours along the faces and in 3D you also have edge neighbours. These will have a different number of relevant cells. If a uniform grid is used, no variation in patch sizes, then if a patch is of size $N \times N \times N$. Then in 3D a face neighbour will have $N \times N \times 2$ relevant cells. A edge neighbour will then have $N \times 2 \times 2$ relevant cells and finally a corner neighbour will have $2 \times 2 \times 2$ cells to pass on. This can be changed if one were to use adaptive-mesh-refinement (AMR), changing sizes of patches in relevant regions. These guard zones are then used to obtain consistency across boundaries.

This means that for a given patch to update it must fetch guard zones and to do this a given patch must check if its neighbouring patches have advanced sufficiently in time. After an update the particles that have moved outside of the patch and into the guard zones, they are put in a "export buffer", such that when a neighbour starts its update it will look in these 26 export buffers and it fetches the particles that now belong to it. This can also work across multiple nodes, here there will be a boundary between the different nodes, and thus a node will have boundary patches at the geometric edge, neighbouring patches from the other node is here called virtual patches. As soon as patches marked as boundary are updated a MPI package is prepared and send to all neighbours that require it.

7.4 Treating particles in Dispatch

In the Dispatch framework, we handle particles with a particle-in-cell method as previously described. Macro-particles are implemented to describe an ensemble of points where the shape function is given by Eq. (4.26). Meaning we use the NGP approach where the particle is evaluated in the nearest grid point.

The particle trajectories are then integrated using a kick-drift-kick leap-frog integrator. From the input file it is decided how many particles there will be per cell. Here it is also decided how many size bins. For the given cell there is a dust density assigned, this total density is then spread out over the particles using a weight, that is normalized to this total density. Meaning

$$\sum_{i}^{N_{bin}} w_i = \rho_d^{cell} \ , \tag{7.91}$$

where N_{bin} is the number of bins specified, w_i the weight for a size bin and ρ_d^{cell} the dust density of a cell. Thus to achieve a specific starting distribution of dust particle sizes, we can use this weight to specify in the cell how much of each size bin we want. We do this by a power-law such that the exponent of this can be decided in the input file

$$w_i = \frac{\rho_g \ d2g}{N_{bin}} \left(\frac{a_i}{a_0} \right)^{\alpha} \ , \tag{7.92}$$

where ρ_g is the gas density, $d2g$ is the dust-to-gas ratio, a_0 is the smallest size specified and α can then be set to anything of desire. It is also possible to set a cut off for the distribution, such that bins above this size will be empty. Thus the bins will exist in the range given, but will only be populated by the above equation up till a cut off size, after which the remaining bins will be empty, ready to later be populated.

In figure 6, it is seen how *Dispatch* is set up to handle particles for the case of the implementation of coagulation and fragmentation. We split the problem up into two parts, the dynamics, and the evolution. The dynamics part is solved by the PIC method moving particles around, the evolution part is solved by the dust evolution model.

7.5 Modeling dust evolution in a molecular cloud

In order to then bring evolution of particles into the framework there are a couple of critical areas that must be handled in a specific way in order to maintain the functionality present in the prototype model.

7.5.1 Translation from particle description to bin description

To implement a dust evolution model as described we must translate between the particle description in *Dispatch* and the bin description of the dust evolution model.

This is done by passing the dust density of a given cell, as well as the velocity in every direction, to the module that handles dust evolution. This module then translates this into a density per bin size, and in turn a number density.

The number density is then used throughout the module to calculate the evolution of the number density distribution as well as the new velocities depending on the momentum conservation.

For every timestep the particles are then integrated and passed to the evolution module changing the distribution of the number density which is implemented in the particles.

7.5.2 Velocity difference in Dispatch

The velocity difference must be calculated between particles to compare with the fragmentation velocity. Every particle has a x, y and z velocity, we find the bulk difference by

$$\Delta u = \sqrt{(v_x^i - v_x^j)^2 + (v_y^i - v_y^j)^2 + (v_z^i - v_z^j)^2} \ , \tag{7.93}$$

where v is the velocity in a given direction, and i, j denotes the specific particles. We add the v_{rms} in square such that

$$\Delta u = \sqrt{(v_{i,x} - v_{j,x})^2 + (v_{i,y} - v_{j,y})^2 + (v_{i,z} - v_{j,z})^2} + \sqrt{v_{i,rms}^2 + v_{j,rms}^2} \ . \tag{7.94}$$

This gives us the total relative velocity for two colliding particles in a very simple way.

7.5.3 Fragmentation velocity

We decided to handle the fragmentation velocity in *Dispatch* to say that monomers would need very high velocities even to be considered for fragmentation. Thus we set a cut-off for small sizes. Then from the cut-off, we set the velocity as given in Eq. (5.41) where the fragmentation velocity is dependent on the size of the particles. At the top of the particle sizes, we set another cut-off for particles, such as when massive aggregates collide, they will always fragment at a certain point, since this is also what we see in laboratory experiments, see figure 7

The easiest way to achieve this was to set the fragmentation depending on the two particles in the collisions such that

$$u_f(i,j) = \begin{cases} u_{f,M} & \text{if } a_i,\ a_j < a_M \\ u_{f,A} & \text{if } a_i,\ a_j > a_A \\ \sqrt{\frac{60\pi^2 \zeta_{crit} \gamma a_\mu}{m_\mu}} & \text{else .} \end{cases} \tag{7.95}$$

Here $a_\mu = a_i a_j/(a_i + a_j)$ and $m_\mu = m_i m_j/(m_i + m_j)$. Here a_M is the monomer size and is a cut-off parameter the user can set (1 μm per default) as the sizes at which the fragmentation velocity is $u_{f,M}$, which is can also be set by the user but is set to 1 km/s by default. In the same way a_A is the size of aggregate and is a cut off at which the fragmentation velocity is given by $u_{f,A}$, default size is 10 cm and default fragmentation velocity is 10 m/s. This means that

below the cut off fragmentation is very hard to achieve and above the other cut off it is very easily achieved. In between these two extremes the fragmentation velocity is handled by the equation with size dependence, where particles are assumed to have aggregate like structure, as presented in Eq. (5.41).

It would be possible to instead implement Eq. (5.37) below a certain range, such that in this range particles are handled as monomers with a size dependent fragmentation velocity. We chose to do a cut off with a user specified fragmentation velocity such that it would be possible for the user to decide. Also meant to be a way of handling cases of crystallized particles, which would have very high fragmentation velocity, higher than predicted by (5.37). As discussed above it would be closer to 1 km/s

7.5.4 Picking the right solving method

We have a explicit method and implicit method implemented in our model. However picking which one to use is tricky as both have their advantageous. Explicit is a rather fast and simple method, but it can easily overshoot if we are using too large timesteps. In those cases an implicit method would be better as it handles bigger steps better in terms of approaching the correct solution. It tends to smear out the density more than the explicit solution. However, the implicit solution has a much higher computational intensity due to the required matrix inversions. Thus it is a balancing act between taking large timestep vs. taking easy explicit steps.

The way we decide to handle this in *Dispatch* is by taking a single explicit step and examining how big a change it would give to the system in terms of dn/n. We then have a set fraction δ_{frac1} such that if $dn/n > \delta_{frac1}$ we will consider the change too large to be acceptable and we instead take an implicit step. Along with this fraction we also consider the mass conservation to a respectable limit.

$$\sum_{i}^{N_{bin}} \frac{n_i^e m_i}{n_i^s m_i} - 1 < 10^{-6} \,, \tag{7.96}$$

such that we are guaranteed not to violate mass conservation to a significant degree, as this could be a accumulating problem over time. Here n^e is the number density at the end of a evolution step and n^s is the number density at the start. N_{bin} is the number of bins. Thus this is simply matching the total mass before and after the evolution step. A third criterion for taking a implicit step is if the explicit step has any bin with negative number density. So if any of these 3 criteria is met we instead take an implicit step. If these are not met we have another δ_{frac2} where if $dn/n > \delta_{frac2}$ we will instead divide the given timestep Δt into smaller substeps such that $\Delta t_{sub} = \frac{\Delta t}{N_{sub}}$, where N_{sub} is decided by the user. The different fractions are also decided by the user in a given input file.

8 Results

In preceding chapters I have explained the background, the theory and numerical model for investigating dust evolution. Now, I will turn my attention to the actual implementation, and results that cast light on how dust evolves in the interstellar medium.

8.1 Prototyping the model

To ensure that the formulated dust evolution model gives reasonable results, it was decided to compare results against previously published results [25]. This lead to building a prototype version of the total numerical dust evolution model before the implementation into the Dispatch framework. The programming language *Python* was decided for prototyping as it offers a lot of flexibility in terms of handling units and linear algebra, which are essential features for a working model. As the prototype is isolated from other possible code issues, and written in *Python*, it makes it considerably easier to debug. Model implementation is similar to the one described in Birnstiel, Dullemond (2010)[25] in the appendix.

A foundation of the numeric model is the logarithmic binning of the grain sizes, and these are then related to mass. We use a starting density given by (5.53) with parameters $T = 196\ K$, $\Sigma_g = 18\ g/cm^2$ and we also assume as dust-to-gas ratio of 1/100, meaning $\Sigma_d = 0.18\ g/cm^2$. The solid density of the dust grain is $\rho_s = 1.6\ g/cm^3$. The initial condition in the test is that all dust is placed in the first bin, which in this case corresponds to the grain size $a = 1\ \mu m$. Different sources of the differential velocity that is driving the dust evolution and first pure coagulation and then also fragmentation are added one by one to the system and the dust evolution is recorded as a function of time—testing both an explicit method and an implicit method.

8.1.1 Explicit vs. Implicit

The prototyping is meant to test the results and see whether it was possible to replicate the results of previous papers with the same assumptions. The results in Birnstiel, Dullemond where the numerical simulation results are compared to an analytical solution provide a simple first benchmark to test the coagulation prototype—testing the foundational coagulation model using an explicit method. The system starts with only the first bin populated.

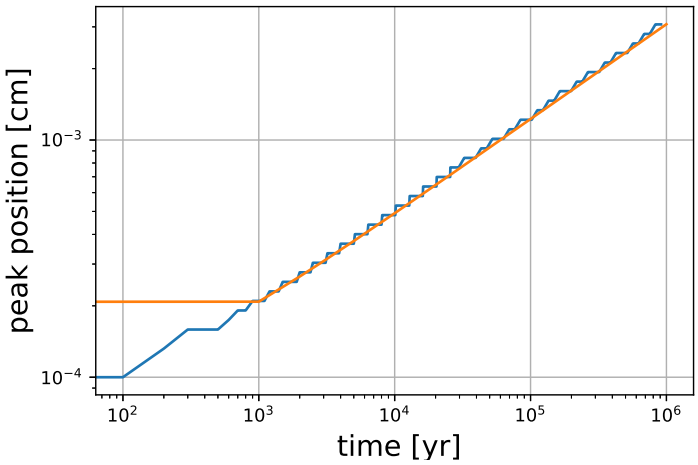


FIGURE 13. *Here we have an example of our test of our coagulation model with only Brownian motion and explicit method. Here we used 200 bins and a timestep of* 100 *yrs. The blue line denotes the peak position of the grain size distribution. Orange is the analytical solution given in* (5.51)

In Figure 13 it is seen that our results match well with those of Birnstiel, Dullemond. Meaning the prototype model can replicate previous results and matches well with analytical calculations. Next, the implicit implementation is tested to ensure that the same results are obtained, regardless of the method.

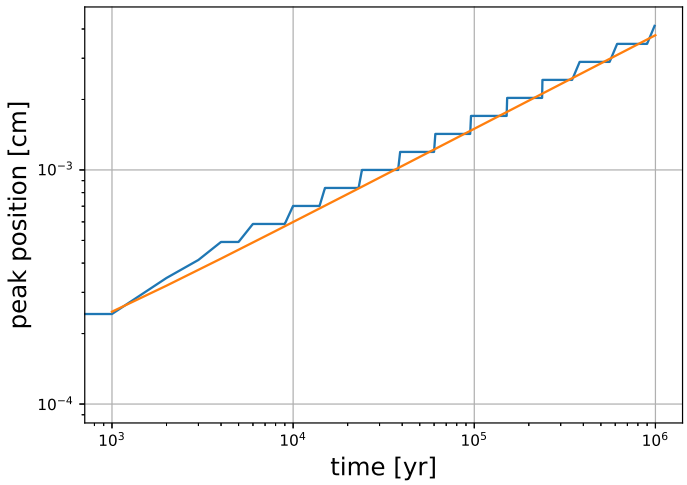
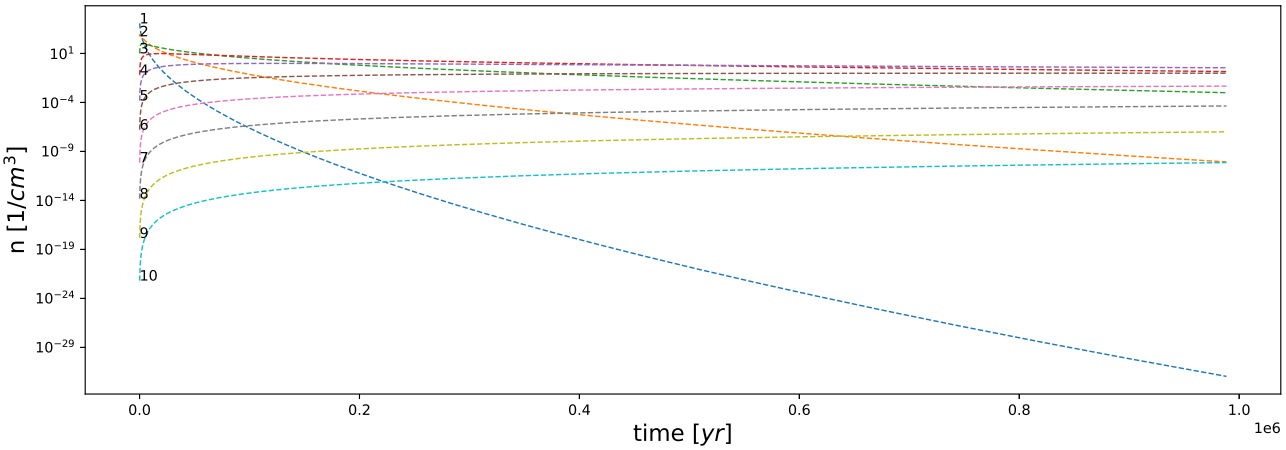


FIGURE 14. *Here the coagulation model with only Brownian motion and, in this case, an implicit method. Here we used 40 bins and timestep of* 1000 *yrs. The blue line denotes the peak position of the grain size distribution. Orange is the analytical solution given in* (5.51)

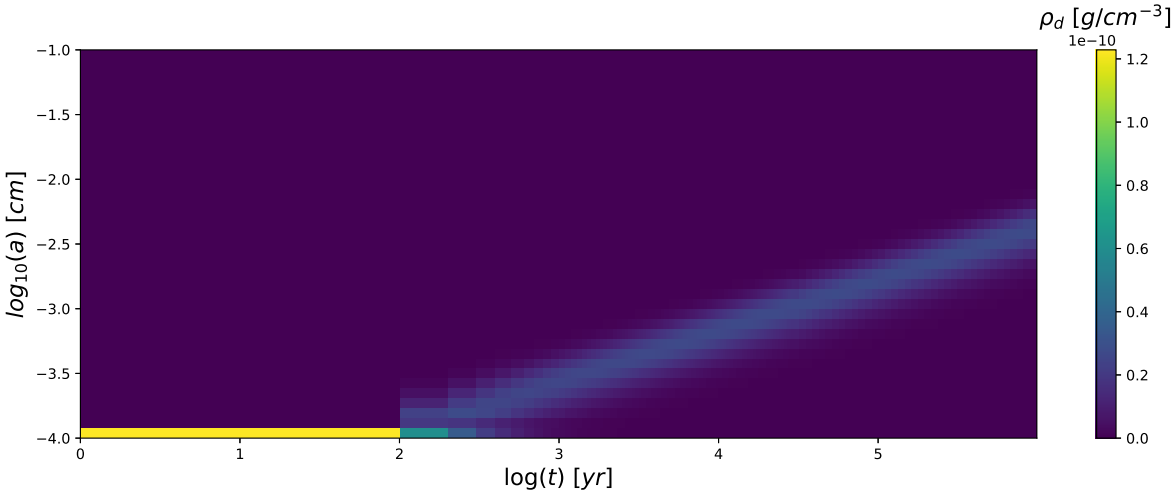
Here we see that we indeed end up with approximately the same results regardless of method. Even though some resolution is lost, the correct trend is present. This result gives us the confidence that thus far, the model lives up to the requirements and can be expanded. We dropped the bin resolution down to 40 bins instead of 200 as we had for the explicit method. This is because the implicit solution requires us to solve a matrix of size equal to the number of bins squared, making it significantly slower for larger number of bins. However, the implicit method enables us to go to higher Δt without having issues with numerical instabilities.

8.1.2 Coagulation model

We can look at the effects the coagulation has on the evolution of each bin — how mass is moved around in the bins. This will give us some insight as to how the model is working. We again run with the implicit method and the same parameters as mentioned above.



(A)



(B)

FIGURE 15. *(a) This shows a plot of the density of every size bin's evolution during the time, here we used* 10 *bins with sizes* $a = 4$ μm *to* $a = 0.1$ cm *and timesteps of* 100 *yrs. Thus we see the effects of coagulation with driven by Brownian motions. Annotated is the number of each bin. (b) is a pictographic representation of the densities for every bin in the form of a heatmap, here 40 bins are used for higher resolution. Otherwise the runs are identical.*

Here, all particles first exist in the smallest size bin, and slowly, this bin loses mass as other bins are populated. It is also clear that the smaller bins are populated right at the start and shortly after losing particles to larger bins. It seems that the system is slowly settling, to the effect that coagulation is slowing down, which makes sense as it rapidly loses small particles, which drive most of the coagulation.

8.1.3 Adding fragmentation

For fragmentation we set the a fragmentation velocity $u_f = 1\ m/s$ and a transition width of $\delta u = 0.001\ m/s$. In the prototype we also use the simple model for the fragmentation velocity with a set u_f with no dependency on size, meaning using eq. (5.32).

Brownian motion, in general, is not enough for fragmentation to occur, at least not with the given parameters. Thus we need another source of relative velocity between grains. We will here add turbulent motion from(3.23). Turbulent motion gives high velocities and is an important source of velocities for fragmentation both in protoplanetary disks and in the interstellar medium.

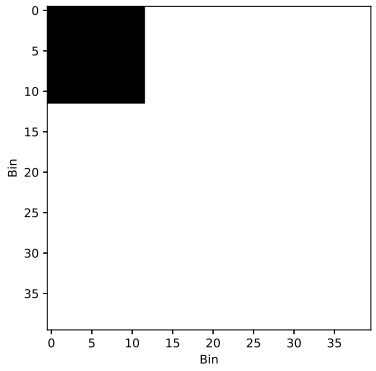


FIGURE 16. *A pictographic of the matrix p_f, where black is the value 0 and white the value 1, shows that for the given bins i, j equivalent x, y what the probability for fragmentation is throughout, where black would be 0 and white 1.*

In figure 16 we see that when turbulent motion is present, fragmentation will occur everywhere except for grain sizes of $< 0.005\ cm$, which will give a constant supply of fragments to the lower bins.

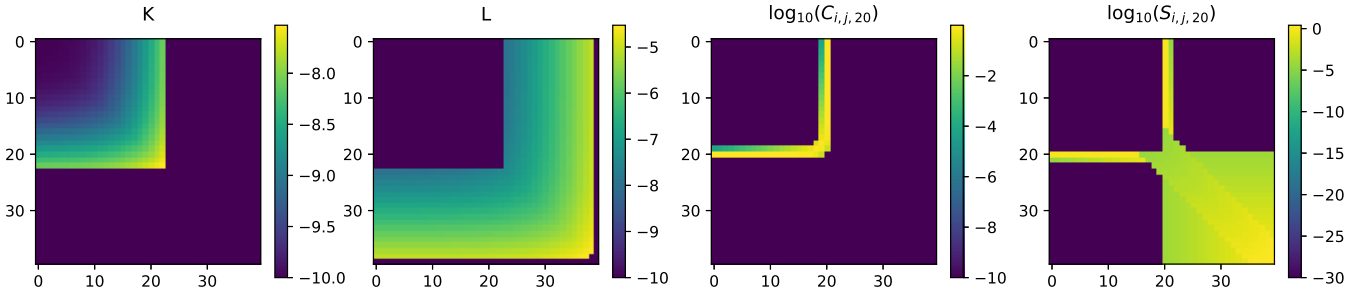
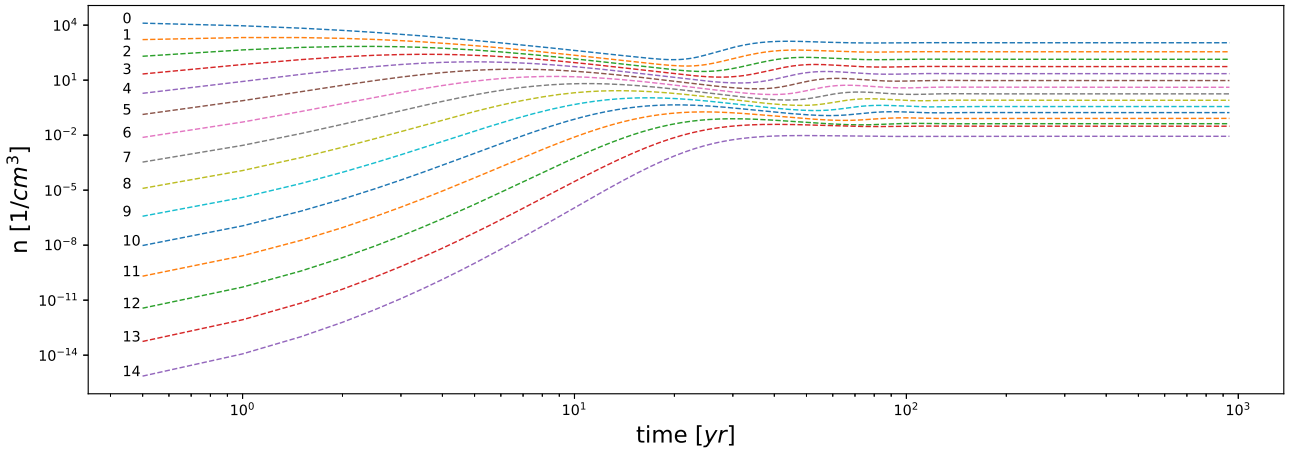


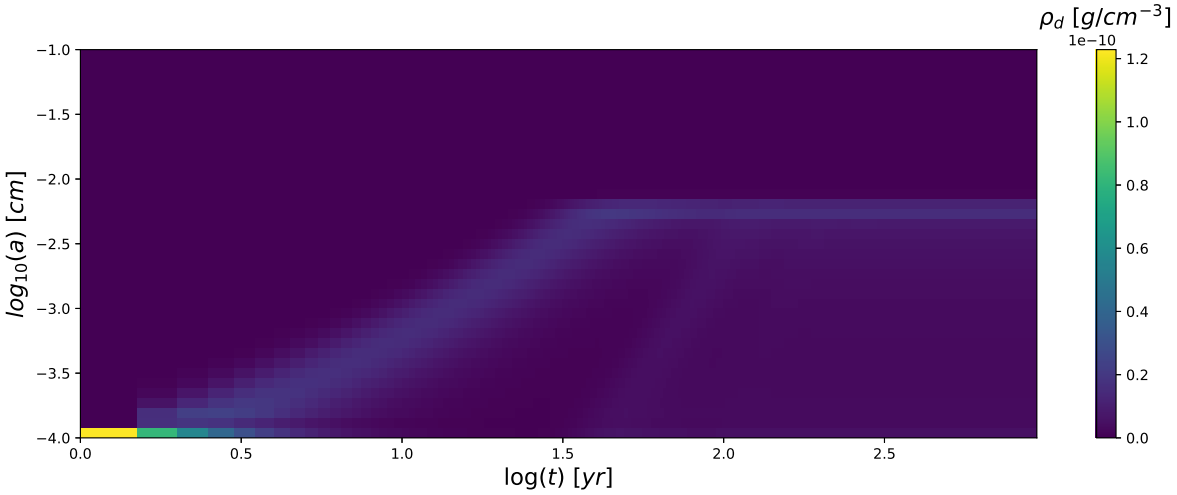
FIGURE 17. *A pictographic of the various matrices that dominate the Smoluchowski equation. C and S is shown for the specific bin 20.*

In figure 17 we see the matrices responsible for the coagulation and fragmentation kernel given in eq. (5.28). K and L show the transfer for the collision of the two bins. There is a clear cut-off where the fragmentation kernel L is dominant. The matrix C is shown for the specific bin 20 corresponding to size $a = 34.5\mu m$. What can be seen from the color is how much the two bins contribute to bin 20—most of the contribution coming from small particles colliding with a particle of size $34.5\mu m$. S shows the contribution from the two bins to that specific fragment in a collision. So to have the fragment of size $34.5\mu m$, the contribution mostly comes

from two of the largest particles colliding. But also from cratering with small particles colliding with the particle of size $34.5\mu m$. Here the matrices C and S will be constant, and K and L depend on the relative velocities and what the fragmentation velocity is, and it will therefore change if the velocities change. However, in this prototype as velocities do not change nor do these matrices.



(A)



(B)

FIGURE 18. *(a) Using 40 bins from $a = 4$ μm to $a = 0.1$ cm and a timestep of 0.5 yrs. Annotated on the lines are the bin number that is followed on the line. And given in the loglog plot. (b) is a pictographic of the densities for every bin in the form of a heatmap for the same run.*

We see that the system quickly reaches a steady state where fragmentation provides fragments and coagulation provides larger particles. We can also see a ceiling where the particles do not seem to coagulate further—a limit to the size driven by fragmentation of larger particles.

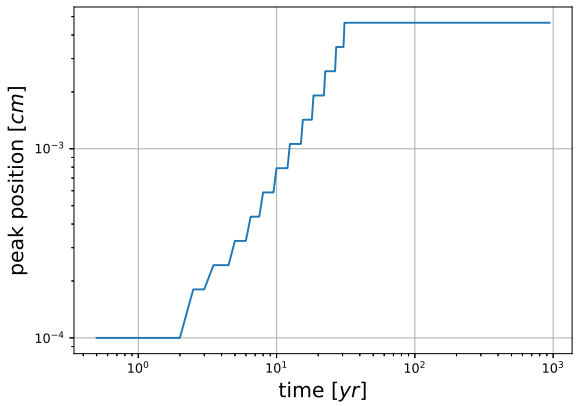


FIGURE 19. *Similar plot to those in the coagulation section, looking at the peak position of the grain size distribution.*

It is also seen from the peak position that the system reaches an equilibrium, and the distribution peak stays constant over time.

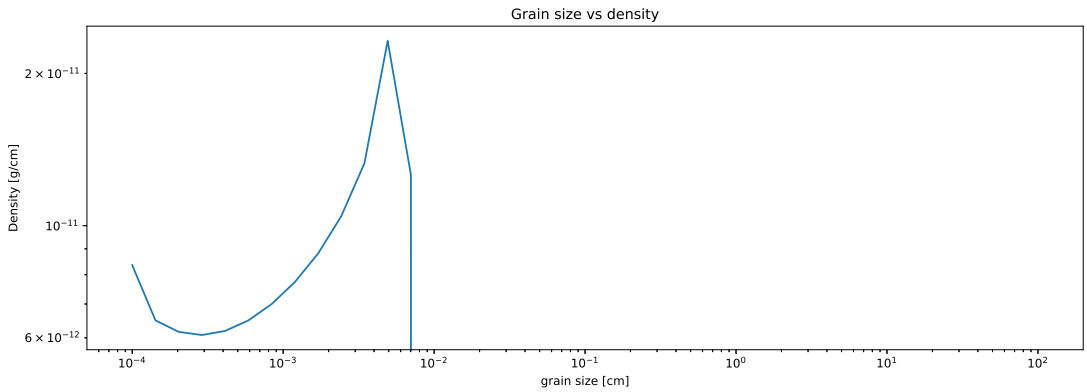


FIGURE 20. *The distribution of the density among the grain sizes, at time* $t = 10^6 yr$

Here we can see how the density is spread out over the grain sizes. Most of the mass has moved to higher bins. Mass is still present in smaller grains; there is, however, a point where the mass does not reach any further at around 50 μm, this peak is already met after only about 30 $years$.

An essential step in the testing was to ensure that we conserve particle number and mass during the evolution. Of course, some mass will be lost or gained due to rounding error, but we made sure the loss was well below significance.

8.1.4 Momentum conservation

To test whether momentum was conserved, we gave the system some start momentum. This is done by a simple one-dimensional velocity given to the smallest bin, which is also the only populated bin from the start. Every particle in this bin thus has a velocity of 1 cm/s.

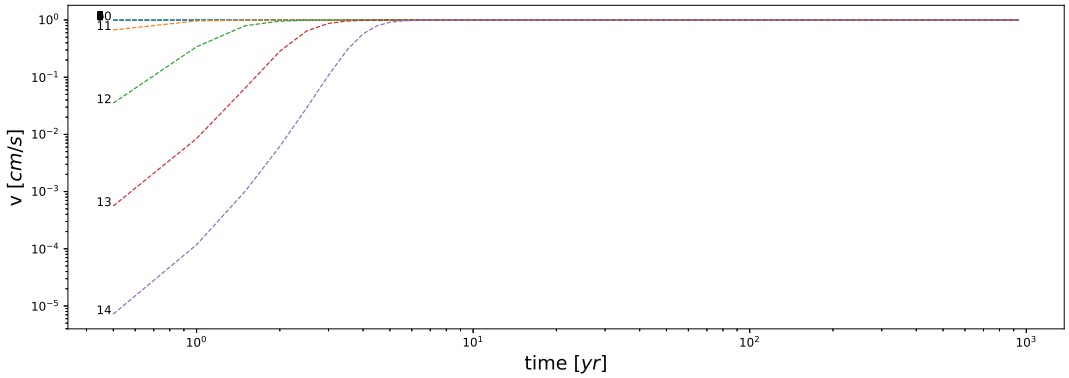


FIGURE 21. *Following the velocity of every bin here, we have 40 bins $a = 4$ μm to $a = 0.1$ cm, most bins do not have sufficient velocity to show up.*

Here it is seen how the velocity spreads out to every bin, and soon all bins have the same velocity, but at some point, the fragmentation and coagulation reaches an equilibrium, and higher bins are not reached.

8.2 Dispatch dust evolution implementation

The way we decide on an experimental setup in *Dispatch* is by an input file that specifies some starting parameters. This means the user chooses what features to turn on and what to keep off. For our purposes, we need to set some parameters to initiate the dust evolution, which means we must also have the particle solver enabled. To involve the dynamics of the dust, we also include gas and hydrodynamics.

The system starts with a dust number density distribution given by the famous MRN distribution [1], between $5 - 250$ nm with an exponent of -3.5. This is run for about 4 turnover times. A snapshot is taken of the system at every $t = 0.1$ where this time is based on the Larson relation as given in Eq.(3.19), meaning that we use the scaling in the code such that the box is $1 \times 1 \times 1$ pc and $\sigma_v = 1.2$ km/s means that the turnover time is $\tau_{ed} = L/(2\sigma_v) = 1\ pc/(2 \cdot 1.2\ km/s) \sim 0.4\ pc\ /km$, meaning in the code the largest eddy has a turnover time of $t = 0.4.$ in years that is around $0.4\ Myr$. The snapshots will then contain everything necessary about the system, such as the various densities in all the cells.

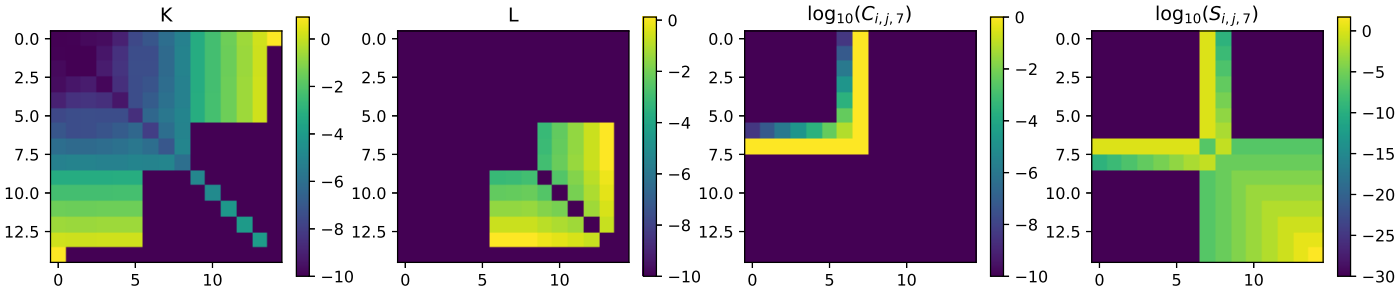


FIGURE 22. *Here the pictographic example of the various matrices that dominate the Smoluchowski equation at $t = 0.6$. C and S are shown for the specific bin 7. Here we have 15 bins in total. This is sampled from a corner of a random patch.*

In 22 we show the different matrices that contribute to the dust evolution for bin nr. 7, corresponding to dust particles of size $0.4\mu m$, this is a sampling of a corner in a random patch,

thus it is just an example. We can see that matrix L is dominant for the larger particle sizes. This means the relative velocity is higher than the fragmentation velocity for these bins. We also see how the coagulation matrix K contributes to smaller particles, mostly collisions of larger particles with small particles. The 3 dimensional matrices C and S here are constant the same for every given cell. It is therefore clear that the matrices fit well with the matrices shown in Figure 17; this indicates that the model is working as intended.

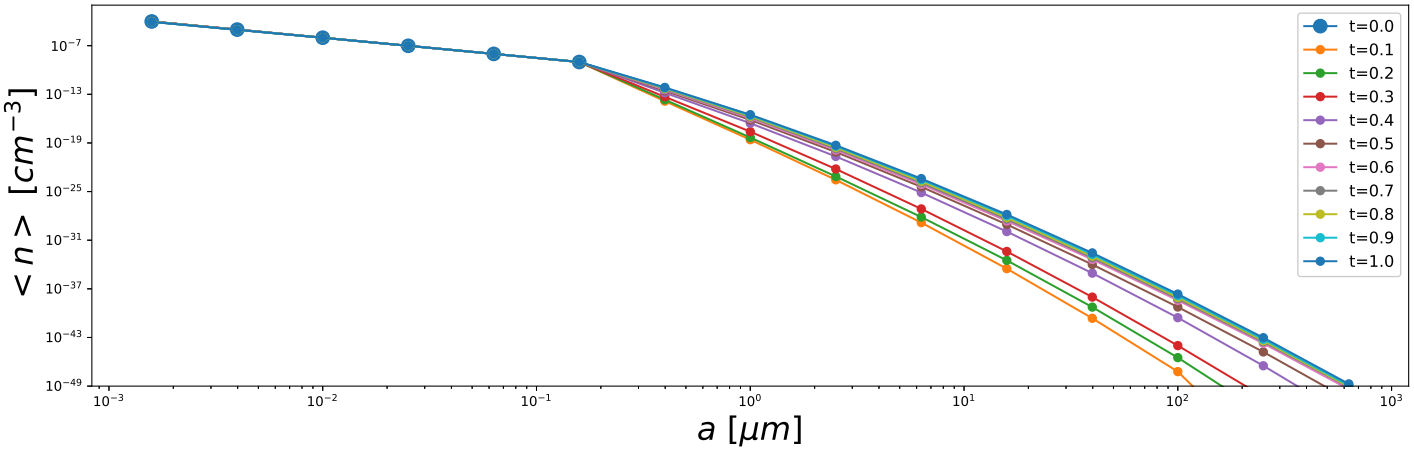


FIGURE 23. *Shown is the number density averaged over all cells for every particle size. All snapshots are shown.*

In Figure 23 it is clear that first the defined distribution is set, then the system quickly defines it as wider distribution with larger particles. Then after about a turnover time, the system evolves to have a slightly different distribution which is held more or less constant through the rest of the time. Large bin sizes have been populated with very small number densities; these densities are so small that it means that in the ISM, there would not be enough particles of larger sizes to be observable. This means that over time no new larger particles appear in the system. Also that the distribution will mostly depend on what particles are inputted into the system and what particles are taken out of the system.

This balance in the distribution can happen if the timescales for building larger particles are of much larger times scales; here, we are looking at timescales of $\sim 1\ Myr$. Another more likely reason and what is also seen in the prototyping results is that the system quickly reaches a steady state as the coagulation and fragmentation is balanced, meaning larger particles are destroyed almost as soon as they are made, reaching an equilibrium between coagulation and fragmentation.

In order to further investigate this balance, we can sample some areas where the velocities are high.

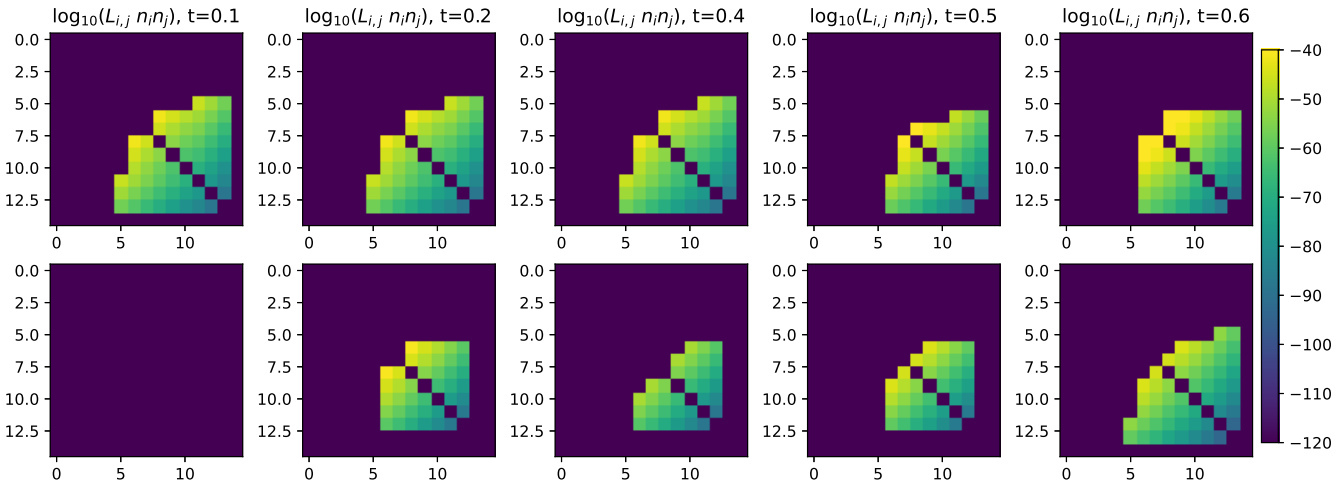
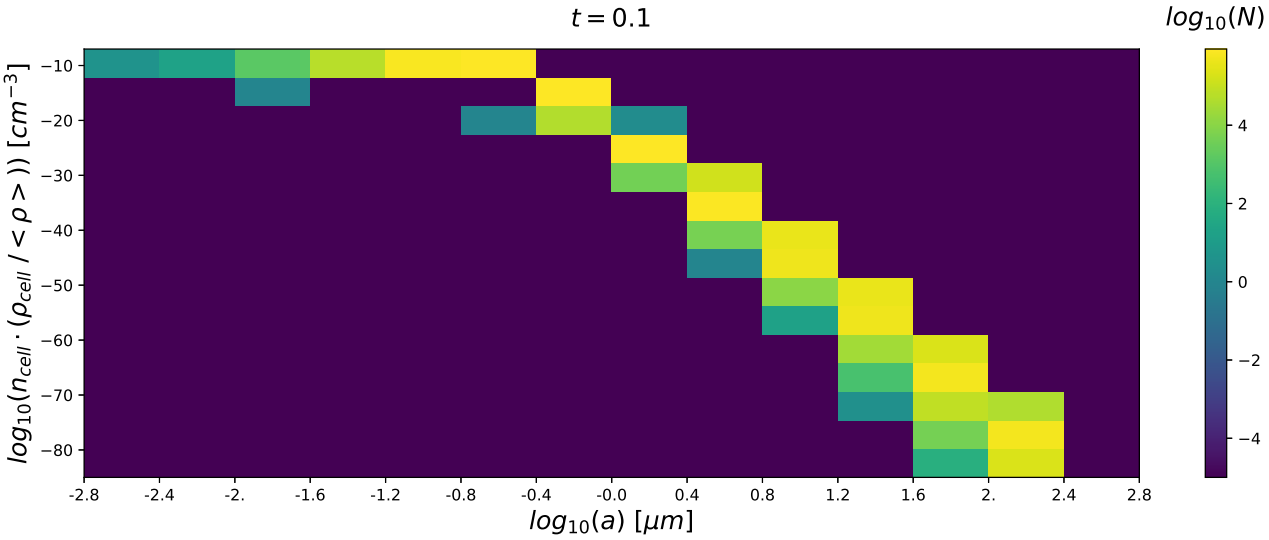
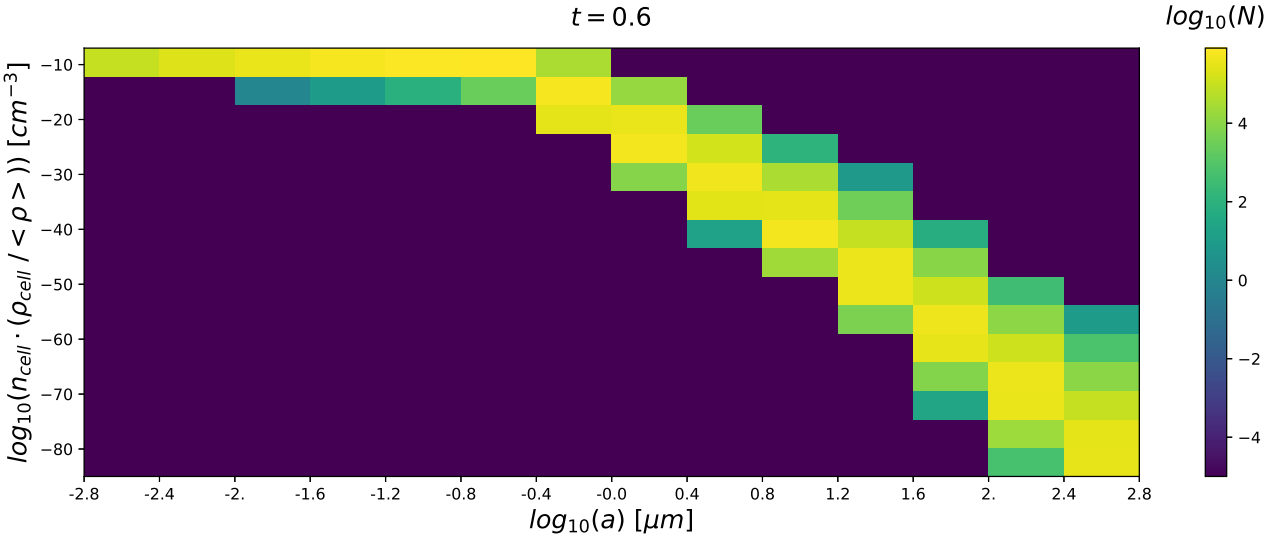


FIGURE 24. *A sampling of the contribution to the fragmentation is shown by the fragmentation kernel matrix L and the number density distribution. In the top row, the sampling is done in a cell with the highest dust root-mean-square velocity. In the bottom row, the sampling is done from a random cell.*

In Figure 24 we see such a sampling done. We can see that in the high v_{rms} regions, the contribution of coagulation is large for larger particle bins at all times. We can see that there is much less contribution found for the randomly sampled areas to begin with, and then the picture starts to change. This is likely due to the turbulence having a large influence on the velocities, meaning the fragmentation becomes more likely throughout the system. This then supports that fragmentation is rampant in the system for large dust particles.



(A)



(B)

FIGURE 25. *A binned heatmap for the number densities. Counts (N) for how many cells have a density within a specific density range.*

From figure 25 we can see exactly how the distribution of densities evolve over time. We see that they start relatively slim and well defined for some of the size bins, meaning less variation in the number density. Later the distributions have all become wider, and larger size bins have been slightly populated.

That the distribution of the number density is widening like this points to the particles moving to less dense regions, especially for larger particles; this seems true as the tail of the distribution is wider for the larger particles.

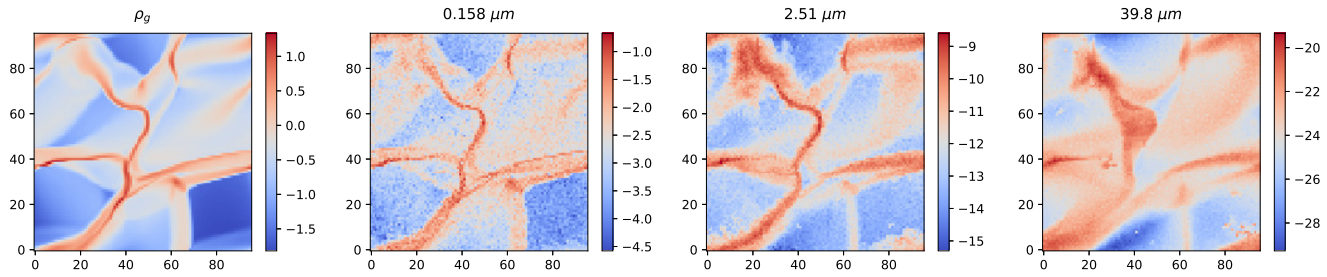


FIGURE 26. *At $t = 0.6$, this shows a slice of the box and all the densities in the cells, where the color red is high-density areas, and blue is low density. This is shown for the gas ρ_g and some chosen size bins.*

In figure 26 we see how the different size particles are distributed throughout the system and how closely these follow the gas densities. In the second panel corresponding to particle size 0.16 μm, we see that the high densities regions very closely resembles that of the gas density regions. This is also what we would expect as these particle sizes are below the sizes we estimated for a Stokes number indicating tight coupling in eq. (3.21). In the third panel, we see the particle start to diffuse further out from the filaments. By 40 μm, there is minimal coupling left between the gas and dust. This indeed also seems to agree with estimates made in (3.22).

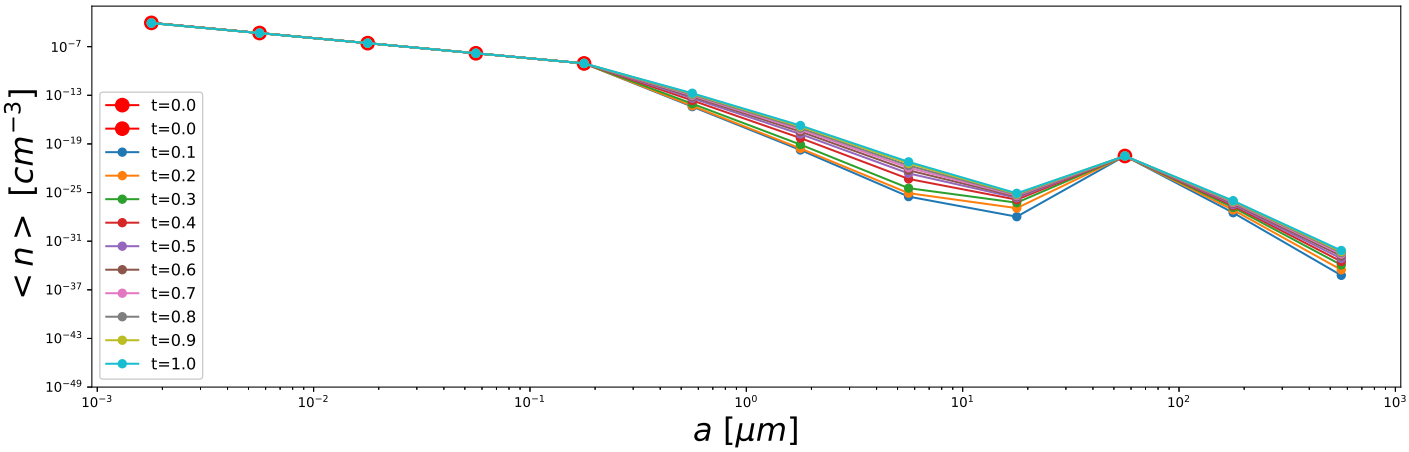


FIGURE 27

In figure 27 we have a slightly different simulation run. Here the same starting distribution is used, with a fraction of the mass put in a size bin outside the otherwise specified range. The specific bin size populated is $\sim 100 \mu m$. Here it is clear that more or less the same thing occurs; a slight evolution of the distribution happens until an equilibrium is reached. However, this also shows that if very large particles are injected into the system, we can expect slight grain growth, but more or less, what is injected into the system will remain there for large timescales. We also see that the particles sizes we start with are also those most likely to survive, as they stay most populated throughout time.

9 Discussion

Here a few thoughts in regards to the implemented model itself and its given results. And its hopefully continuous usefulness.

9.1 Results of Dispatch simulations

The simulation results show the ISM to be relatively static in dust evolution on timescales of order dynamical times, and hence also of order star formation times.

On the other hand it is suspected that the protoplanetary disks that form around newborn stars are important sites of dust processing, and therefore it would be interesting to run these simulations with adaptive mesh refinement (AMR), sufficient to resolve the formation of protoplanetary disks, and eventually also with sufficient resolution to determine how much processed dust they return to the ISM.

With respect to the dust processing, it would be interesting to see how tweaking the fragmentation velocity will affect the grain growth in the ISM. I suspect that setting a higher fragmentation velocity would cause large grains to have a better chance of surviving and thus affecting the timescales to grow significant number densities of, for example, 100 μm sized particles.

However, since the fragmentation velocity is based on laboratory experiments, this is not likely to be consistent with reality. However, it is possible that a significant fraction of the particles in the ISM is thermally processed material, meaning they would have higher fragmentation velocities.

9.2 Further development of the Dispatch implementation

There are a lot of possible improvements to be made to this model. These can increase the functionality of the model and thus the *Dispatch* framework. Improvements to the computational cost and accuracy might also be possible, even though a lot has already been done to improve the module's computation speed.

It would be possible to give particles an attribute *"Crystallinity"*, or simply a boolean flag indicating if the dust has been thermally processed. This could play into how the model should handle the fragmentation velocity. In the case of such particles, crystalline particles will be extremely difficult to shatter, as they are essential diamond-like. Adding this would add the possibility of including particles processed in Class-0 protostars or AGB processed material as these materials will have significantly higher fragmentation velocities than amorphous particles. It would also be possible to add a population of such materials and see how the distribution would end up after fragmentation has run its course, and how many of these particles would survive compared to the amorphous particles.

9.3 Energy consideration in cratering regime amongst others

In our model, we have set a simple cutoff for the regime of cratering. Meaning cratering occurs if there is an order of magnitude or more difference in the mass between the two colliding particles. In the outcomes predicted by Dominik & Tielens (1997), which we used to determine fragmentation velocities, they consider the energy in the collision instead. Splitting the evolution into finer regimes as

- $E < 5E_{roll}$ Sticking without restructuring
- $E \approx E_{roll}$ Onset of restructuring at impact site
- $E \approx N_c E_{roll}$ Maximum compression
- $E \approx 3N_c E_{br}$ Onset of erosion (losing monomers in aggregate)
- $E > 10N_c E_{br}$ Full fragmentation

Where N_c is the number of contacts in the aggregate, it would then be possible to look at the energies in collisions and determine some form of cratering based on both the mass of the particles and energies involved in the specific collisions.

9.3.1 Geometry of particles

This might be a consideration of a computationally heavier nature. But the shape of particles and whether they are smooth or rough, and how porous they are, have considerable impacts firstly on how the particles will move with the gas as the coupling is determined in part by the drag coefficient also shown in figure 4. Secondly, the microphysics of collisions between particles would also change if we were to consider other ways of particles dissipating energies in collisions by way of their structure.

9.3.2 Numerical diffusion

Often the method of solving the Smoluchowski equation is used for the dust evolution. Another prevalent way of describing the dust evolution process, numerically, is using a Monte-Carlo approach [45], wherein the life of an ensemble representative particle is followed. These particles are treated as representing a much larger volume. The simulation then follows these particles as they collide and fragment or coagulate. Random numbers are used to determine which particles collide, and depending on the properties of the colliding particles, the outcome is decided. The advantage of this model is it is not very computationally heavy, and it is easy to add features to the particles, and diffusion is not as much as a problem as in the Smoluchkowski method as particles are individual masses, whereas in the Smoluchkowski method diffusion can happen as we spread the particles into bins of number density.

This problem of numerical diffusion can show up due to lucky collisions and has been shown to, in some cases, cause a massive overestimate of the rate at which particles break through the bouncing barrier and an underestimate of the timescale [46]. Monte-Carlo methods have the opposite problem due to their limited dynamic range of density; they tend to underestimate the timescale.

However, newer models of the Smoluchkowski type aim to mitigate this diffusion issue [47].

9.3.3 Optimizing the choice of method

When we solve our Smoluchowski equation(5.27) in the discrete form, we use either an explicit method or implicit method, each with their computational benefit. In our implementation, we decide to have a set limit for choosing the implicit method over the explicit method. However, one could find the optimal choice in terms of computing time compared to the order of mass conservation desired. This could be done in the initialization of the problem, such that depending on what compiler, libraries, etc., were used, the optimal choice of when the framework would pick explicit over implicit would be set as a start parameter.

9.3.4 Adaptive bin refinement

Adaptive bin refinement, where the number and particle size range of bins was determined locally, for each computational patch in DISPATCH, could be a simple solution to the problem of numeric diffusion, mentioned above. Also, it would help the issue of high memory usage by the particle solver and the dust solver. This would also allow the user to specify a range of bin sizes of interest and use higher resolution in these areas. From (J. Drazkowska, 2018)[47] the conclusion is that the high-mass bins with low number density are the important area to have higher resolution in to mitigate diffusion and overestimates. This would also be possible if using an adaptive bin refinement method.

10 Summary and Conclusion

I have described the dust contributors, what the defining features of dust are, and what observational evidence there is of dust accumulating and agglomerating in different mediums. I discussed how large dust grains might form, what some of the main issues are, and implemented a way for these issues to be resolved numerically.

I introduced the dynamics of gas and dust and how the two interact, which is necessary to understand how the collisions of dust grains play out. I discussed several aspects of the micro-physics involved in a single dust-dust collision, where the interaction can be split into different regimes as; coagulation, bouncing, and fragmentation. These regimes are then defined by various key aspects of the collision, primarily the difference in size and the relative velocity between particles. Here if the relative velocity between particles is above a certain threshold, I define a regime of fragmentation. If the particle size difference is large enough, I handle the collision as a cratering event, where the smaller particle digs out fragments of the larger particle.

With these definitions and considerations, I formulated a dust evolution model that could handle both the processes of coagulation and fragmentation in a numeric simulation. To achieve this, I first implemented a prototype version of the model in *Python*. Here a number density distribution is created from an estimate of the total dust mass in a model protoplanetary system. This number density distribution is then evolved in time, assuming collisions happen continuously. The model then has no dynamics to take into account. To test the code results I compared them with the results of previous authors and found that our model matched well. In this prototype, using the effects of coagulation and fragmentation, the dust seemed to coagulate very quickly, reaching a peak in the distribution at around 50 μm within 30 years; whereafter the system remains in equilibrium at this peak.

This *Python* prototype was then translated into *Fortran* and added into the code framework *Dispatch*, in which particles have a dynamic component. Here a few difficulties arise as the framework handles particles with a macro-particle representation. This meant the model had to be adapted since the dust evolution handles the particles in a bin description, which means a translation between the particle description and the bin description and back was necessary. Also, the dust evolution had significant computational cost, which had to be circumvented to run a reasonable experiment in a reasonable time. In the end, these challenges have been overcome, and now a working model of dust evolution in *Dispatch* exists, which runs reasonably well. The computational cost of the dust evolution module is around the same as the dynamic solver of the particles, meaning it can easily run large-scale simulations.

I ran simulations with the new implementation, using supersonic turbulence representing the cold interstellar medium in a $1 \times 1 \times 1$ pc box, with an initial dust particle distribution given by the famous MRN distribution [1]. The results show a slight initial evolution of the number distribution, but after about a turnover time, when the supersonic turbulence is dominant in the system, meaning fragmentation has high contributions. This leads to a more or less steady state of the system where fragmentation and coagulation has found an approximate equilibrium, with very small rates of net change. The conclusion is that the time scales for grain growth and fragmentation in the cold ISM are much longer than the the typical dynamical time scales we look at here, $\sim 1 Myr$.

This confirms that the ISM is not the ideal medium for the dust to evolve to larger grain sizes. It is more likely that the ISM dust distribution is defined by a mass balance among

the particles that are injected into it and removed from it; i.e., by the sources and sinks of dust particles in the ISM. Sources could be supernovae, AGB stars or PPDs. If the fraction of thermally processed material is high, this conclusion would be even more robust, since these particle types require significantly larger collision velocities to fragment.

The main result of this thesis is a well tested dust evolution module, which can be used for future large-scale scientific investigations, is now present in the *Dispatch* framework. This is likely to be particularly important in the context of dust processing in proto-planetary disks, which may turn out to an important source of processed dust, in addition to clearly being an important sink of largely unprocessed dust.

References

1. Mathis, J. S., Rumpl, W. & Nordsieck, K. H. The size distribution of interstellar grains. **217,** 425–433 (Oct. 1977).
2. Sabri, T. *et al.* Interstellar Silicate Analogs for Grain-surface Reaction Experiments: Gas-phase Condensation and Characterization of the Silicate Dust Grains. *The Astrophysical Journal* **780,** 180– (Jan. 2014).
3. Zhukovska, S., Henning, T. & Dobbs, C. Iron and Silicate Dust Growth in the Galactic Interstellar Medium: Clues from Element Depletions. *The Astrophysical Journal* **857,** 94. ISSN: 1538-4357. http://dx.doi.org/10.3847/1538-4357/aab438 (Apr. 2018).
4. Murata, K. *et al.* Crystallization Experiments on Amorphous Silicates with Chondritic Composition: Quantitative Formulation of the Crystallization. **668,** 285–293 (Oct. 2007).
5. Suh, K.-W. Crystalline silicates in the envelopes and discs around oxygen-rich asymptotic giant branch stars. *Monthly Notices of the Royal Astronomical Society* **332,** 513–528. ISSN: 0035-8711. eprint: https://academic.oup.com/mnras/article-pdf/332/3/513/6399587/332-3-513.pdf. https://doi.org/10.1046/j.1365-8711.2002.05303.x (May 2002).
6. Poteet, C. A. *et al.* A SPITZER INFRARED SPECTROGRAPH DETECTION OF CRYSTALLINE SILICATES IN A PROTOSTELLAR ENVELOPE. *The Astrophysical Journal* **733,** L32. https://doi.org/10.1088/2041-8205/733/2/l32 (May 2011).
7. Boyer, M. L. *et al.* THE DUST BUDGET OF THE SMALL MAGELLANIC CLOUD: ARE ASYMPTOTIC GIANT BRANCH STARS THE PRIMARY DUST SOURCE AT LOW METALLICITY? *The Astrophysical Journal* **748,** 40. https://doi.org/10.1088/0004-637x/748/1/40 (Mar. 2012).
8. Sugerman, B. E. K. Massive-Star Supernovae as Major Dust Factories. *Science* **313,** 196–200. ISSN: 1095-9203. http://dx.doi.org/10.1126/science.1128131 (July 2006).
9. Slavin, J. D., Dwek, E., Mac Low, M.-M. & Hill, A. S. The Dynamics, Destruction, and Survival of Supernova-formed Dust Grains. *The Astrophysical Journal* **902,** 135. ISSN: 1538-4357. http://dx.doi.org/10.3847/1538-4357/abb5a4 (Oct. 2020).
10. Ferrara, A., Viti, S. & Ceccarelli, C. The problematic growth of dust in the interstellar medium (June 2016).
11. Wong, Y. H. V., Hirashita, H. & Li, Z.-Y. Millimeter-sized grains in the protostellar envelopes: Where do they come from? *Publications of the Astronomical Society of Japan* **68.** 67. ISSN: 0004-6264. eprint: https://academic.oup.com/pasj/article-pdf/68/4/67/6848217/psw066.pdf. https://doi.org/10.1093/pasj/psw066 (June 2016).
12. Slavin, J. D., Dwek, E. & Jones, A. P. DESTRUCTION OF INTERSTELLAR DUST IN EVOLVING SUPERNOVA REMNANT SHOCK WAVES. *The Astrophysical Journal* **803,** 7. https://doi.org/10.1088/0004-637x/803/1/7 (Apr. 2015).
13. Jones, A. P. The mineralogy of cosmic dust: astromineralogy. *European Journal of Mineralogy* **19,** 771–782. http://dx.doi.org/10.1127/0935-1221/2007/0019-1766 (Dec. 2007).
14. Xiao, T. *et al.* Dust reddening in star-forming galaxies. *Proceedings of the International Astronomical Union* **8,** 291–291 (2012).
15. Giannetti, A. *et al.* Galactocentric variation of the gas-to-dust ratio and its relation with metallicity. *A&A* **606,** L12. https://doi.org/10.1051/0004-6361/201731728 (2017).

16. Fonfría, J. P. *et al.* The complex dust formation zone of the AGB star IRC+10216 probed with CARMA 0.25 arcsec angular resolution molecular observations. **445,** 3289–3308. arXiv: 1409.5833 [astro-ph.SR] (Dec. 2014).

17. Andrews, S. M. *et al.* The Disk Substructures at High Angular Resolution Project (DSHARP). I. Motivation, Sample, Calibration, and Overview. *The Astrophysical Journal* **869,** L41. ISSN: 2041-8213. http://dx.doi.org/10.3847/2041-8213/aaf741 (Dec. 2018).

18. Bouwman, J. *et al.* Processing of silicate dust grains in Herbig Ae/Be systems. **375,** 950–962 (Sept. 2001).

19. van Boekel, R. *et al.* Grain growth in the inner regions of Herbig Ae/Be star disks. **400,** L21–L24 (Mar. 2003).

20. Carrasco-González, C. *et al.* The Radial Distribution of Dust Particles in the HL Tau Disk from ALMA and VLA Observations. *The Astrophysical Journal* **883,** 71. ISSN: 1538-4357. http://dx.doi.org/10.3847/1538-4357/ab3d33 (Sept. 2019).

21. Guidi, G. *et al.* Dust properties across the CO snowline in the HD 163296 disk from ALMA and VLA observations. **588,** A112. arXiv: 1601.07542 [astro-ph.SR] (Apr. 2016).

22. V.P.L.S., R. B. F. H. M. R. A. XXVII. A brief account of microscopical observations made in the months of June, July and August 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies. *The Philosophical Magazine* **4,** 161–173. eprint: https://doi.org/10.1080/14786442808674769. https://doi.org/10.1080/14786442808674769 (1828).

23. Epstein, P. S. On the Resistance Experienced by Spheres in their Motion through Gases. *Physical Review* **23,** 710–733 (June 1924).

24. NASA. *Drag of a sphere* https://www.grc.nasa.gov/WWW/k-12/airplane/dragsphere.html.

25. Birnstiel, T., Dullemond, C. P. & Brauer, F. Gas- and dust evolution in protoplanetary disks. *Astronomy and Astrophysics* **513,** A79. ISSN: 0004-6361, 1432-0746. http://www.aanda.org/10.1051/0004-6361/200913731 (2021) (Apr. 2010).

26. Blum, J. & Wurm, G. The Growth Mechanisms of Macroscopic Bodies in Protoplanetary Disks. *Annual Review of Astronomy and Astrophysics* **46,** 21–56. eprint: https://doi.org/10.1146/annurev.astro.46.060407.145152. https://doi.org/10.1146/annurev.astro.46.060407.145152 (2008).

27. Crida, A. Minimum Mass Solar Nebulae and Planetary Migration. **698,** 606–614. arXiv: 0903.5077 [astro-ph.EP] (June 2009).

28. Larson, R. B. *The evolution of molecular clouds* in *The Structure and Content of Molecular Clouds 25 Years of Molecular Radioastronomy* (eds Wilson, T. L. & Johnston, K. J.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 1994), 13–28. ISBN: 978-3-540-49035-7.

29. Heyer, M., Krawczyk, C., Duval, J. & Jackson, J. Re-Examining Larson's Scaling Relationships in Galactic Molecular Clouds. *Astrophysical Journal - ASTROPHYS J* **699** (Sept. 2008).

30. Ormel, C. W. & Cuzzi, J. N. Closed-form expressions for particle relative velocities induced by turbulence. **466,** 413–420. arXiv: astro-ph/0702303 [astro-ph] (May 2007).

31. Koshizuka, S., Shibata, K., Kondo, M. & Matsunaga, T. in *Moving Particle Semi-implicit Method* (eds Koshizuka, S., Shibata, K., Kondo, M. & Matsunaga, T.) 1–23 (Academic Press, 2018). ISBN: 978-0-12-812779-7. https://www.sciencedirect.com/science/article/pii/B9780128127797000011.

32. Dendy, R. *Plasma Physics: An Introductory Course* ISBN: 9780521484527. https://books.google.dk/books?id=puuQM4Dx0zYC (Cambridge University Press, 1995).

33. Hockney, R. W. & Eastwood, J. W. *Computer simulation using particles* (Bristol: Hilger, 1988, 1988).

34. Smoluchowski, M. V. Drei Vortrage uber Diffusion, Brownsche Bewegung und Koagulation von Kolloidteilchen. *Zeitschrift fur Physik* **17,** 557–585 (Jan. 1916).

35. Blum, J., Levasseur-Regourd, A.-C., Muñoz, O., Slobodrian, R. J. & Vedernikov, A. Dust in space. *Europhysics News* **39,** 27–29 (May 2008).

36. Jones, A. P., Tielens, A. G. G. M. & Hollenbach, D. J. Grain Shattering in Shocks: The Interstellar Grain Size Distribution. en. *The Astrophysical Journal* **469,** 740. ISSN: 0004-637X, 1538-4357. http://adsabs.harvard.edu/doi/10.1086/177823 (2021) (Oct. 1996).

37. Ormel, C. W. *The early stages of planet formation: how to grow from small to large* English. OCLC: 256461865. ISBN: 9789036735599. http://irs.ub.rug.nl/ppn/314552375 (2021) (s.n.] ; University Library Groningen] [Host, S.l.; Groningen, 2008).

38. Dominik, C. & Tielens, A. G. G. M. The Physics of Dust Coagulation and the Structure of Dust Aggregates in Space. en. *The Astrophysical Journal* **480,** 647–673. ISSN: 0004-637X, 1538-4357. https://iopscience.iop.org/article/10.1086/303996 (2021) (May 1997).

39. Poppe, T., Blum, J. & Henning, T. New experiments on collisions of solid grains related to the preplanetary dust aggregation. en. *Advances in Space Research* **23,** 1197–1200. ISSN: 02731177. https://linkinghub.elsevier.com/retrieve/pii/S0273117799001829 (2021) (Jan. 1999).

40. Ossenkopf, V. Dust coagulation in dense molecular clouds : the formation of fluffy aggregates. en. *Astronomy and Astrophysics* **280,** 617–646. ISSN: 0004-6361. https://ui.adsabs.harvard.edu/abs/1993A&A...280..617O/abstract (2021) (Dec. 1993).

41. Hirashita, H. & Yan, H. Shattering and coagulation of dust grains in interstellar turbulence. en. *Monthly Notices of the Royal Astronomical Society* **394,** 1061–1074. ISSN: 00358711, 13652966. https://academic.oup.com/mnras/article-lookup/doi/10.1111/j.1365-2966.2009.14405.x (2021) (Apr. 2009).

42. Brauer, F., Dullemond, C. P. & Henning, T. Coagulation, fragmentation and radial motion of solid particles in protoplanetary disks. *Astronomy & Astrophysics* **480,** 859–877. ISSN: 0004-6361, 1432-0746. http://www.aanda.org/10.1051/0004-6361:20077759 (2021) (Mar. 2008).

43. Podolak, M. & Podolak, E. A numerical study of aerosol growth in Titan's atmosphere. en. *Icarus* **43,** 73–84. ISSN: 00191035. https://linkinghub.elsevier.com/retrieve/pii/0019103580900895 (2021) (July 1980).

44. Nordlund, Å., Ramsey, J. P., Popovas, A. & Küffmeier, M. dispatch: a numerical simulation framework for the exa-scale era – I. Fundamentals. *Monthly Notices of the Royal Astronomical Society* **477,** 624–638. ISSN: 1365-2966. http://dx.doi.org/10.1093/mnras/sty599 (Mar. 2018).

45. Zsom, A. & Dullemond, C. P. A representative particle approach to coagulation and fragmentation of dust aggregates and fluid droplets. **489,** 931–941. arXiv: 0807.5052 [astro-ph] (Oct. 2008).

46. Drażkowska, J., Windmark, F. & Dullemond, C. P. Modeling dust growth in protoplanetary disks: The breakthrough case. **567,** A38. arXiv: 1406.0870 [astro-ph.EP] (July 2014).

47. Booth, R. A., Meru, F., Lee, M. H. & Clarke, C. J. Breakthrough revisited: investigating the requirements for growth of dust beyond the bouncing barrier. **475,** 167–180. arXiv: 1712.03171 [astro-ph.EP] (Mar. 2018).

Appendix

A Algorithm of the prototype implementation

Here the full python script for running the basic simulations of the dust evolution model.

```
import numpy as np
import astropy.units as u
from astropy import constants as const
import matplotlib.pyplot as plt
from pylab import rcParams
rcParams['figure.figsize'] = 15, 5



basis = u.si.bases

print("Initializing starting conditions")
T = 196 *u.K
cs = (np.sqrt(const.k_B*T/(2.37*1.008*const.m_p))).decompose(basis)
r = 1 * u.au
msolar = 0.5 *u.solMass
Omega_K = np.sqrt(const.G*msolar/r**3)
Hp = (cs / Omega_K).decompose()
sigma_d = 0.18 * u.g/u.cm**2
rho_d = (sigma_d/(np.sqrt(2*np.pi)*Hp)).decompose(basis)
rho_s = 1.6 *u.g * u.cm**(-3)
a_s = 1*u.micron
m_s = (4/3*np.pi*a_s**3 * rho_s).decompose(basis)
u_f = (1*u.m/u.s).decompose(basis)
trans_width = (0.001*u.m/u.s).decompose(basis)
alpha = 1e-3
Cdt = 0.1
xi = 1.83

dt = 1000 * 60*60*24*365 * u.s
t_end = 1e6* 60*60*24*365 * u.s
save_time = 1 *u.yr
eps_time = 1.10



print("Calculating bins")

nbin = 40
a = np.logspace(-4, -1, nbin) * u.cm
st = (rho_s * a * np.pi/(2*sigma_d*100)).value
v = np.ones(nbin)
```

```
m = (4/3 * a**3 * rho_s * np.pi).decompose(basis)
n = np.zeros(len(m)) * u.m**(-3)
n[0] = (rho_d / m[0]).decompose()
n = n.decompose(basis)
ms = np.add.outer(m,m)
mv = m.value * v
p = n.value * v
q = m.value * p


print("Calculating velocities and sigma_col")
ubm = (np.sqrt(8 * const.k_B*T * (np.add.outer(m, m)) /
              (np.pi*np.outer(m, m)))).decompose(basis)
utm = cs * np.sqrt(2*alpha*st)
utm[st>1] = cs*np.sqrt(2*alpha/st[st>1])

utm = np.maximum.outer(utm,utm)
du = ubm + utm

sigma_col = np.pi * np.add.outer(a, a)**2

print("Setting up probabilities and fragmentation matrix ")

size = len(a)
mn = np.zeros((size,size))
mm = np.zeros((size,size))
pf = np.zeros((size,size))

crat_ind = (np.divide.outer(m,m) > 10) | (np.divide.outer(m,m) < 0.1)
SS = np.zeros((size,size,size))

for i in range(len(m)):
    for j in range(len(m)):
        kn = (np.where(m<ms[i][j])[0])[-1

        if crat_ind[i][j]:
            #cratering
            SS[:min(i,j)+1,i,j] = a[:min(i,j)+1] ** (-3*xi)
            Stot = np.sum(m.value * SS[:,i,j] ) / (m[min(i,j)].value *2)
            SS[:min(i,j)+1,i,j] /= Stot
            m_crat =  np.abs(m[i].value-m[j].value)
            l = np.where(m.value < m_crat )[0][-1]
            eps = (m_crat - m[l].value)/(m[l+1].value - m[l].value)
            SS[l,i,j] = 1-eps
            SS[l+1,i,j] = eps

        else:
            #fragmentation
            SS[:max(i,j)+1,i,j] = a[:max(i,j)+1] ** (-3*xi)
```

```
            Stot = np.sum(m.value * SS[:,i,j] ) / (m[i].value + m[j].value)
            SS[:max(i,j)+1,i,j] /= Stot

        if kn+1 < nbin:
            mm[i][j] = m[kn].value
            mn[i][j] = m[kn+1].value

        else:
            sigma_col[i][j] = 0


        if du[i][j] < u_f-trans_width:
            pf[i][j] = 0
        elif du[i][j] > u_f:
            pf[i][j] = 1
        else:
            pf[i][j] = 1-(u_f-du[i][j])/trans_width

pc = 1-pf



ep = (mn - ms.value) / (mn - mm)
ep[sigma_col==0] = 0

C = np.zeros((size,size,size))
for k in range(size):
    imm = m[k].value == mm
    imn = m[k].value == mn
    C[k][imm] = ep[imm]
    C[k][imn] = 1 - ep[imn]

K = (np.multiply(np.multiply(du, sigma_col), pc)
     ).decompose(basis)
L = (np.multiply(np.multiply(du, sigma_col), pf)
     ).decompose(basis)


print("Time evolving the number density")
print("Starting n: ", (n.to(1/u.cm**3)))
print("Initial velocity: ", p)
n = n.value
K = K.value
L = L.value
dt = dt.value
m = m.value
ms = ms.value
t_end = t_end.value
```

```
t = 0
p_old = np.copy(p)
vel = np.max(p/(n+1e-10))
x = t
y = np.argmax(n*m)
y2 = np.argmax(n)
n_save = np.copy(n)
vel2 = p/(n+1e-10)
time_int = 0
has_run = False
while t < t_end:
    if np.any(n < 0) or np.abs(np.sum(n*m)/rho_d.value - 1.) > 2e-1:
        if np.any(n<0):
            print("Negative number particles, aborting", np.argmin(n), np.min(n))
        if np.abs(np.sum(n*m)/rho_d.value - 1.) > 2e-1:
            print("Mass not conserved, aborting", (np.sum(n*m)/rho_d.value - 1.))
        print(f"Time elapsed: {t/(60*60*24*365)} yr")
        break

    dn = np.zeros(size)
    dn_coag = np.zeros(size)
    dm = np.zeros(size)
    Sp = np.zeros(size)
    dq = np.zeros(size)
    S = np.zeros(size)
    J = np.zeros((size,size))
    Jp = np.zeros((size,size))
    Jm = np.zeros((size,size))
    JJ = np.zeros((size*2,size*2))
    nn = np.outer(n,n)
    Q = np.multiply(K,nn)
    Q_frag = np.multiply(L,nn)

    mp = m * p
    pn = (np.outer(mp, n) + np.outer(n, mp))/ms

    for k in range(size):

        KC_k = np.multiply(K, C[k])
        LS_k = np.multiply(L, SS[k])

        Kn_k = np.multiply(K[k], n)
        Ln_k = np.multiply(L[k], n)

        Sp[k] = 1/2 * np.sum(np.multiply(KC_k, pn ) ) - np.sum(Kn_k * p[k]) \
            + 1/2 * np.sum(np.multiply(LS_k, pn)) - np.sum(Ln_k * p[k])

        S[k] = (0.5 * np.sum(np.multiply(Q, C[k])) - np.sum(Q[k])) + \
```

```
            (0.5 * np.sum(np.multiply(Q_frag, SS[k])) - np.sum(Q_frag[k]))

        Jp[k] = (np.dot(KC_k/ms, n) + np.dot(LS_k/ms, n)) * m
        Jp[k][k] -= np.sum(Kn_k) + np.sum(Ln_k)
        Jm[k] = (np.dot(KC_k/ms, (p*m) ) + np.dot(LS_k/ms, (p*m))) \
            - p[k]*K[k] - p[k]*L[k]
        J[k] = (np.dot(KC_k, n) - n[k]*K[k]) + \
              (np.dot(LS_k, n) - n[k]*L[k])
        J[k][k] -= np.sum(Kn_k) + np.sum(Ln_k)

    for i in range(size):
        J[i,i]=0.0
        J[i,i] = J[i,i] - np.sum(m*J[:,i])/m[i]


    JJ[:size, :size] = (np.identity(size)/dt - J)
    JJ[size:, size:] = (np.identity(size)/dt - Jp)
    JJ[:size, size:] = Jm
    SSS = np.concatenate((S, Sp))

    dd = np.linalg.solve(JJ,SSS)
    dn = dd[:size]
    dp = dd[size:]
    n += dn
    p += dp
    t += dt


    if t > save_time.value:
        time_int += 1
        save_time *= eps_time
        y = np.append(y, np.argmax(n*m) )
        y2 = np.append(y2, np.argmax(n))
        n_save = np.append(n_save, n)
        x = np.append(x,t)
        vel = np.append(vel, np.max(p/(n+1e-10)))
        vel2 = np.append(vel2, p/(n+1e-10))
        if time_int % 10==0:
            print("Time left:", (t_end-t)/(60*60*24*365), \
            n[0], "Stepping: ", dt/(60*60*24*365))
            print("p", p[0])


print("Lost mass to total ratio : ")
print(((sum(n*m)-rho_d.value)/rho_d.value))
print("Final n: ", n)
print("Final velocity: ", p/(n+1e-30))

print("Momentum conservation: ", np.sum(p*m) -
```

```
np.sum(p_old*m), (np.sum(p*m)-np.sum(p_old*m))/np.sum(p_old*m))
```