




A Monte Carlo method for tracking dust properties during coagulation in protoplanetary disks

Nerea Gurrutxaga^{*}, Vignesh Vaikundaraman, and Joanna Drążkowska

Max Planck Institute for Solar System Research, Justus-von-Liebig-Weg 3, 37077 Göttingen, Germany

Received 19 February 2026 / Accepted 7 April 2026

ABSTRACT

Dust growth is a crucial step in planet formation, and the efficiency of this process is controlled by the physical and chemical properties of the dust grains. Monte Carlo-based methods are commonly used to follow the collisional evolution of dust while tracking their properties. However, current Monte Carlo methods in planet formation do not strictly conserve the global inventory of dust properties across the protoplanetary disk, causing fluctuations that can grow over time and affect predictions of dust evolution. Here we present a coagulation algorithm that ensures the global conservation of dust properties while resolving the spatial evolution of dust. The method is validated against analytical solutions for standard coagulation kernels and benchmarked in a two-dimensional disk. We show that the method reproduces standard results, resolves the full dust population, and improves the resolution of the small-grain regime compared to other Monte Carlo methods for modeling global dust evolution. Finally, using a test case that includes sublimation and condensation of water interacting with silicates, we demonstrate strict conservation of each component's mass during coagulation, establishing the method as a valuable tool for tracking dust properties in protoplanetary disks.

Key words. methods: numerical – methods: statistical – planets and satellites: composition – planets and satellites: formation – planets and satellites: general – protoplanetary disks

1. Introduction

Planet formation requires dust particles to grow across many orders of magnitude in size within protoplanetary disks. Micrometer-sized dust coagulates into centimeter-sized pebbles before it can form kilometer-sized planetesimals (Youdin & Goodman 2005) or be accreted by growing protoplanets (Ormel & Klahr 2010; Lambrechts & Johansen 2012). The physical and chemical properties of dust particles affect their transport and collision outcomes and thereby the efficiency of dust coagulation (Birnstiel 2024). A comprehensive understanding of planet formation, therefore, requires studying how dust properties affect coagulation. This task is challenging, as resolving dust coagulation in detail is computationally demanding.

Two main approaches are commonly used to model dust coagulation in protoplanetary disks. The first class consists of Monte Carlo methods that explicitly track particle evolution. These methods were introduced to planet formation studies independently by Ormel et al. (2007) and Zsom & Dullemond (2008), with later developments by Ormel & Spaans (2008); Drążkowska et al. (2013) and Beutel & Dullemond (2023). The implementation described in Drążkowska et al. (2013) is available as the open-source code `mcdust` (Vaikundaraman et al. 2025b). The second widely used approach is the mass grid-based method (e.g., Weidenschilling 1980; Dullemond & Dominik 2005; Brauer et al. 2008; Birnstiel et al. 2010), in which the dust mass distribution is discretized into bins and interactions between bins are calculated. This approach is well established in the community, with open-source implementations such as the one-dimensional code `dustpy` (Stammler & Birnstiel 2022)

and the two-dimensional code `cuDisc` (Robinson et al. 2024). A major challenge of the mass-grid method is the inclusion of additional particle properties, as they substantially increase the complexity and computational cost of the method. While extensions to account for properties such as porosity, chemical composition, and charge are possible (Ossenkopf 1993; Okuzumi et al. 2009; Stammler et al. 2017; Akimkin et al. 2023), they require simplifying assumptions – such as that dust particles of equal mass share the same properties. In contrast, Monte Carlo methods enable a more straightforward implementation of additional properties without such assumptions, although adequately sampling the larger parameter space still requires increased numerical resolution.

Monte Carlo methods widely used in planet formation studies are based on the stochastic description of coagulation introduced by Gillespie (1975). Because it is not feasible to track all dust particles in a protoplanetary disk, these methods rely on a limited number of representative particles, whose evolution is assumed to describe that of many physical particles. There are different ways to define and evolve such representative particles, each way leading to different strengths and limitations. In the approach presented by Ormel et al. (2007), collisions are computed between individual representative particles confined within a small volume that represents a larger volume. When two particles coagulate, a new representative particle is created by randomly duplicating an existing one in order to keep the total number of representative particles constant. To preserve the dust density, the volume enclosing the particles is increased accordingly. This method can achieve high accuracy, particularly during the earliest stages of growth. An improved and more efficient implementation was presented by Ormel & Spaans (2008), showing that several physical collisions can be grouped into a

* Corresponding author: gurrutxaga@mps.mpg.de

single collision event. However, the general approach by Ormel et al. (2007) is designed to resolve dust coagulation locally, which can make it challenging to study global dust evolution, as dust transport influences coagulation across the disk (Birnstiel et al. 2012). An alternative Monte Carlo method was introduced by Zsom & Dullemond (2008). In this approach, a small number of representative particles is tracked, each representing a large group of identical physical particles. To compute collisions, the representative particle i is assumed to collide with physical particles identical to those represented by particle j . However, only particle i is updated after the collision. It was shown that, when a sufficient number of representative particles is used, the expected coagulation behavior is well reproduced. This method has two key advantages: it allows for spatially resolved dust coagulation, and it requires a relatively small number of particles to capture the statistical evolution of the system (Drażkowska et al. 2013).

Monte Carlo methods facilitate the inclusion of additional particle properties with minimal modification to the algorithm and have already been employed to study porosity (Ormel et al. 2007; Zsom & Dullemond 2008; Krijt et al. 2015) and chemical composition (Krijt et al. 2016; Houge & Krijt 2023; Vaikundaraman et al. 2025a). However, Krijt et al. (2016) noted that although the approach by Zsom & Dullemond (2008) can conserve the total dust mass, it does not strictly conserve the mass of different components. This arises because of the asymmetry of only updating the representative particle i during a collision with a physical particle j , thereby introducing fluctuations in the composition. These fluctuations become particularly significant when size-dependent processes operate and colliding dust particles have largely different compositions. In the alternative method by Ormel et al. (2007), the global inventory of dust properties is conserved during pairwise collisions. However, since a particle is duplicated to keep the number of representative particles constant, fluctuations can also arise when dust particles have largely different compositions.

A Monte Carlo method that both strictly conserves quantities such as dust composition and enables spatial resolution is still lacking. This limitation can hinder the development of more advanced models. In this work, we present a Monte Carlo approach that is spatially resolvable and enforces strict conservation of the global inventory of dust properties. Our method is comparable to that presented by Shima et al. (2009) for cloud microphysics but is formulated here in the context of planet formation. In Sect. 2, we describe the algorithm. In Sect. 3, we validate the method through a series of tests by comparing our results with analytical solutions of the Smoluchowski equation for coagulation kernels, computing dust evolution in a two-dimensional protoplanetary disk, and following the collisional evolution of a multi-component dust population. We summarize our findings in Sect. 4.

2. The Monte Carlo algorithm

We assume that the total population of N_{tot} dust particles can be represented by N_r representative particles. A representative particle i represents the evolution of N_i identical particles of mass m_i that constitute the group $\{i\}$ with a total group mass of $M_i = N_i m_i$. With M_{tot} being the total population mass,

$$N_{\text{tot}} = \sum_{i=1}^{N_r} N_i, \quad (1)$$

$$M_{\text{tot}} = \sum_{i=1}^{N_r} M_i. \quad (2)$$

N_{tot} decreases over time as particles coagulate into larger particles. However, M_{tot} should be conserved over time.

In the model presented here, we group particles that undergo one-to-one collisions in parallel as independent Poisson processes. A similar strategy was adopted by other authors (Ormel & Spaans 2008; Zsom & Dullemond 2008; Shima et al. 2009), although the grouping procedure differs between them. In our approach, as in Zsom & Dullemond (2008) and Shima et al. (2009), the number of representative particles, N_r , equals the number of particle groups and remains constant throughout the simulation. In contrast, in Ormel & Spaans (2008) the number of representative particles (or species) and groups are not equal, and either or both can vary depending on the specific method. When two groups $\{i\}$ and $\{j\}$ collide, we update particles involved in the collision from both groups as in Ormel & Spaans (2008) and Shima et al. (2009). By contrast, in the method of Zsom & Dullemond (2008), only one group is updated per collision event. First, in Sect. 2.1 we explain how to calculate the collision rate and choose collision events between pairs of particle groups. In Sect. 2.2, we describe how sticking collisions between two groups of particles are modeled. In Sects. 2.3 and 2.4, we discuss details of the algorithm that improve accuracy and computational efficiency. Figure 1 summarizes the algorithm.

2.1. Collisions between groups of particles

The collision rate $C_{i,j}$ between particle i and j in a volume V that encloses N_i and N_j particles of each group is given by

$$C_{i,j} = N_i N_j \frac{K_{i,j}}{V}, \quad (3)$$

where $K_{i,j}$ is a coagulation kernel. In protoplanetary disks, the kernel is typically defined as

$$K_{i,j} = \Delta v_{i,j} \sigma_{i,j}, \quad (4)$$

where $\Delta v_{i,j}$ is the relative velocity between particle i and j and $\sigma_{i,j} = \pi(a_i + a_j)^2$ is the geometrical cross section of their collision, with a_i and a_j denoting the particle radii.

The number of collisions required for all particles from group $\{i\}$ or $\{j\}$ to collide is given by $N_{\text{col}} = \min(N_i, N_j)$. When one-to-one collisions are grouped, the effective collision rate between representative particles i and j is

$$\tilde{C}_{i,j} = \frac{C_{i,j}}{N_{\text{col}}} = \frac{N_i N_j \frac{K_{i,j}}{V}}{\min(N_i, N_j)}. \quad (5)$$

If $N_i < N_j$, a grouped collision event takes place when every particle in group $\{i\}$ collides with a particle in group $\{j\}$. In that case, the collision rate simplifies to

$$\tilde{C}_{i,j} = N_j \frac{K_{i,j}}{V}, \quad (6)$$

which corresponds to the collision rate between a defined grouped $\{i\}$ to collide with N_j particles (Liffman 1992, their Eq. (1)). The grouped collision rate in Eq. (5) is symmetric, such that $\tilde{C}_{i,j} = \tilde{C}_{j,i}$. This differs from the method described by Zsom & Dullemond (2008).

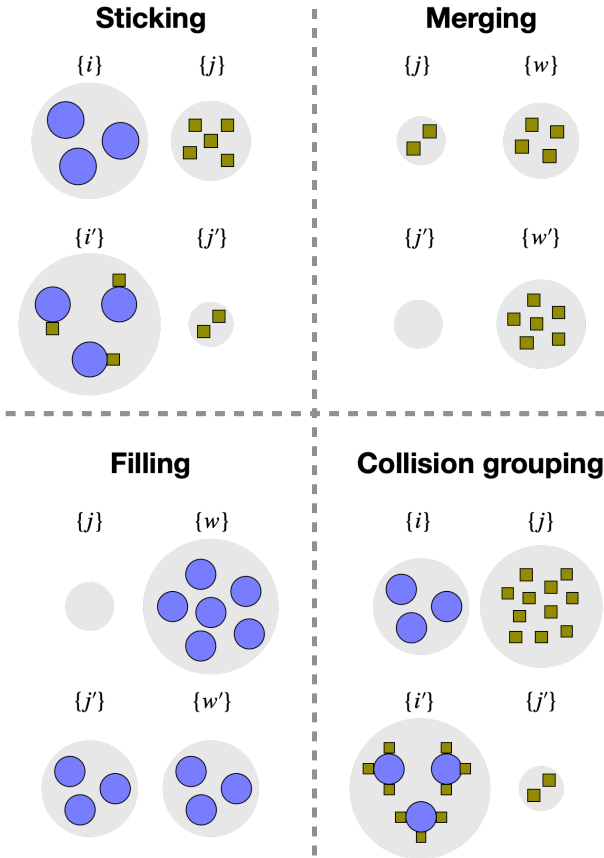


Fig. 1. Summary of the Monte Carlo algorithm presented in this paper. We group individual identical particles. *Sticking*: when group $\{i\}$ collides with group $\{j\}$, as $N_i < N_j$, every particle in group $\{i\}$ sticks to one particle that belongs to group $\{j\}$. *Merging*: if group $\{j\}$ contains negligible mass, it merges with group $\{w\}$, with j and w being particles with similar characteristics. *Filling*: when group $\{j\}$ is emptied, it is filled with half of the particles from the largest group $\{w\}$. *Collision grouping*: if group $\{i\}$ collides with $\{j\}$, with particle j being very small compared to particle i , we group collisional events, meaning that in a single collision we stick several j particles into a single particle i .

Once the collision rate between groups is determined, we compute the collision probabilities using the full-conditioning method of Gillespie (1975). The collision probability density function $P(\tau, i, j)$ can be expressed as the product of three one-variable probability functions:

$$P(\tau, i, j) = P_1(\tau) \cdot P_2(i | \tau) \cdot P_3(j | i). \quad (7)$$

Here, $P_1(\tau)$ is the probability that the next collision occurs within the time interval $[t, t + \tau]$, $P_2(i | \tau)$ is the probability that group $\{i\}$ participates in the given collision, and $P_3(j | i)$ is the probability that group $\{j\}$ participates in the collision, knowing that group $\{i\}$ is involved. Defining the total collision rate as C_{tot} and the collision rate of group $\{i\}$ as \tilde{C}_i ,

$$C_{\text{tot}} = \sum_{i=1}^{N_r} \tilde{C}_i, \quad (8)$$

$$\tilde{C}_i = \sum_{j=i}^{N_r} \tilde{C}_{i,j}. \quad (9)$$

We include the possibility that group $\{i\}$ collides with itself. In our method, this can be allowed because half of the particles in

$\{i\}$ can collide with the other half. When computing the collision rates in Eq. (5), we account for this by replacing N_i with $N_i/2$. Consequently, there are $N_r(N_r + 1)/2$ distinct collision pairs.

Each probability in Eq. (7) is defined as

$$P_1(\tau) = C_{\text{tot}} \exp(-C_{\text{tot}}\tau), \quad 0 \leq \tau < \infty, \quad (10)$$

$$P_2(i | \tau) = \frac{\tilde{C}_i}{C_{\text{tot}}}, \quad (11)$$

$$P_3(j | i) = \frac{\tilde{C}_{i,j}}{\tilde{C}_i}. \quad (12)$$

By generating three random numbers from a uniform distribution, we determine both the time of the next collision and the pair of groups involved.

2.2. Sticking

When two groups collide, the properties of groups $\{i\}$ and $\{j\}$ are modified according to the collision outcome, which depends on their relative velocity Δv_{ij} . Above specific velocity thresholds, collisions result in bouncing, fragmentation, or erosion, while below these thresholds, particles stick (e.g., Blum & Münch 1993; Güttler et al. 2010; Zsom et al. 2010; Schräpler & Blum 2011; Kelling et al. 2014). Here we describe how sticking events are computed. For clarity, we assume $N_i \leq N_j$ in this section; the opposite case is handled by exchanging the indices.

When particles in groups $\{i\}$ and $\{j\}$ collide at velocities that enable sticking, every particle in group $\{i\}$ coagulates with a particle in group $\{j\}$ (see Fig. 1). In a single simulated collision event, the $N_i \times m_j$ mass is transferred from group $\{j\}$ to group $\{i\}$. We update the mass of representative particles i and j , as well as the number of particles N_i and N_j constituting groups $\{i\}$ and $\{j\}$ as

$$m'_i = m_i + m_j, \quad (13)$$

$$m'_j = m_j, \quad (14)$$

$$N'_i = N_i, \quad (15)$$

$$N'_j = N_j - N_i. \quad (16)$$

After each sticking event, the collision rates in Eq. (6) for every pair involving groups $\{i\}$ and $\{j\}$ are recalculated, and Eqs. (8) and (9) are updated accordingly.

If $N_i = N_j$, group $\{j\}$ is emptied with $N'_j = 0$ when sticking. The empty group $\{j\}$ can then be reused to split another group $\{w\}$, thereby better resolving the evolution of particles of type w . Our criterion for splitting groups is described in Sect. 2.3.

In this method, we advance the system over multiple independent collisions in a single step. The approximation is valid provided that the system properties do not vary significantly over this sequence. This condition is most easily satisfied during sticking events involving particles with different mass, since coagulation between similar particles produces the largest relative changes in the system. The approximation is also more accurate for collisions involving the largest particles, since a group of large particles typically contains fewer particles. In Sect. 3, we show that grouping one-to-one collisions provides reliable estimates for typical dust evolution problems.

2.3. Merging and filling the empty group

When a group $\{j\}$ becomes empty, the freed memory can be used to split an existing group $\{w\}$ into two (see Fig. 1). Unlike the

approaches in Ormel et al. (2007) and Ormel & Spaans (2008), we did not duplicate particles. Instead, we began from large particle groups and increased the resolution by splitting existing groups at later times, as similarly done by Shima et al. (2009).

A convenient choice of which group $\{w\}$ to split improves the resolution, whereas a poor choice generates particles that occupy space but are unimportant to the system's evolution, as they represent only a tiny fraction of the total mass. In a coagulation event between group $\{i\}$ and $\{j\}$, with $N_i \leq N_j$, group $\{i\}$ gains mass, while group $\{j\}$ loses mass. If the collision probability increases with particle mass, as in protoplanetary disks, group $\{i\}$ is expected to play an increasingly important role in the subsequent collisional evolution. It is therefore more likely to undergo further collisions and continue growing. When a single group $\{i\}$ dominates the mass distribution while still comprising many particles, the grouped description becomes inaccurate, because it artificially reduces the particles' growth rate compared to an ungrouped treatment. A practical way to mitigate this effect is to preferentially split the group $\{i\}$ that contains the largest group mass $M_i = N_i m_i$ (see Fig. 1). Splitting the groups with the largest mass increases the resolution in the high-mass regime. Alternative splitting criteria can be employed to resolve the distribution of smaller particles better. For instance, one may adopt an approach similar to Ormel & Spaans (2008), dividing the size distribution into exponentially spaced mass bins while aiming to maintain a similar number of representative particles per bin.

Even with an appropriate choice of how groups are filled, some groups may still contribute negligibly to the collisional evolution of the system. This occurs because group $\{j\}$ typically requires several coagulation steps to lose all its mass. To accelerate this process, we introduced a group-mass threshold M_{\min} , and groups with masses below this value are considered negligible and are emptied. We define

$$M_{\min} = X \cdot \frac{M_{\text{tot}}}{N_r}, \quad (17)$$

where M_{tot}/N_r is the average mass of a group (see Eq. (2)), and X is a user-defined merging parameter ($0 \leq X < 1$). A lower value of X leads to fewer merging events and, consequently, fewer large groups are split. Therefore, achieving higher resolution among the largest particles requires choosing a higher value of X . Conversely, if the focus is on the small-particle population, X can be reduced (see further discussion in Sect. 3.2).

When group $\{j\}$ is forced to empty, it can be merged with its most similar group $\{w\}$ to minimize the noise added to the simulation (Ormel & Spaans 2008). We quantify similarity between groups by calculating the distances in the property space. In Sects. 3.1 and 3.2, we consider only the particle mass m_i as a relevant property for merging, and therefore the distance between groups $\{i\}$ and $\{j\}$ is defined as

$$d_{ij} = |m_i - m_j|. \quad (18)$$

We compute d_{ij} for all $i \neq j$ and find the group $\{w\}$, which has the shortest distance d_{wj} . In Sect. 3.3, we track water and silicate masses, $m_{\text{ice},i}$ and $m_{\text{si},i}$, and thus define distance as

$$d_{ij} = \sqrt{(m_{\text{ice},i} - m_{\text{ice},j})^2 + (m_{\text{si},i} - m_{\text{si},j})^2}. \quad (19)$$

Equation (19) is especially suitable for the multicomponent simulation discussed in Sect. 3.3 because we can express properties in terms of mass. However, there are instances where this is not possible, and one property may have a significantly broader

range of values than others. In such cases, alternative merging criteria may be more suitable (e.g., Krijt et al. 2015, for particle mass and porosity).

We treat merging as a process distinct from coagulation (see Fig. 1): when group $\{j\}$ merges with group $\{w\}$, their total group mass is combined, while the new mass of the representative particle w is computed as an average such that

$$m'_w = \frac{N_w m_w + N_j m_j}{N_w + N_j}, \quad (20)$$

$$m'_j = 0, \quad (21)$$

$$N'_w = N_w + N_j, \quad (22)$$

$$N'_j = 0, \quad (23)$$

where $\{w\}$ is the group whose particle properties are most similar to those in group $\{j\}$. After group $\{j\}$ is emptied, it is refilled with half of the particles in the group $\{i\}$ that contains the largest group mass M_i . After each merging and filling event, the collision rates in Eq. (6) of pairs involving any modified groups are recalculated, and Eqs. (8) and (9) are updated accordingly.

As particles merge, their properties average out, losing information about the dispersion of those properties. To test the validity of the results against this artificial smoothing, we can vary the merging parameter X . If in simulations with little or no merging (i.e., low X), particle properties are more spread than in simulations with frequent merging events, this suggests that merging artificially smooths the distribution of particle properties. In such cases, it may be necessary to increase the number of representative particles N_r , reduce X , or modify the criteria for selecting the most similar particle.

2.4. Physical collision grouping

When the particle size distribution is broad, collisions between particles i and j with very different masses can be frequent. These collisions, however, change the mass of the largest particle only marginally in a single collisional event. It is therefore possible to combine several physical collisions into a single effective collision.

Zsom & Dullemond (2008) and Ormel & Spaans (2008) introduced numerical optimizations to speed up the calculation of this type of collision, which is also applicable to our method. In their approach, collisions between individual particles are grouped such that a single representative particle undergoes many physical collisions. It is important to note that this differs from the particle grouping described so far. Until now, we have considered the grouping of N_i events across N_i different particles, whereas in this section we focus on each particle experiencing multiple collisions (see Fig. 1).

Zsom & Dullemond (2008) suggested grouping collisions based on the particle mass ratio, using a grouping minimum mass dm_{\max} , such that collisions are grouped when $m_j/m_i \leq dm_{\max}$. To implement this in our method, a sufficient number of physical particles must be available for collision grouping. If $m_j/m_i \leq dm_{\max}$, we modify the collision rate in Eq. (6) as

$$\tilde{c}'_{i,j} = \frac{\tilde{c}_{i,j}}{n_{\text{group}}}, \quad (24)$$

where n_{group} is the number of physical collisions that every particle i undergoes in one simulated event. The value of n_{group} depends on the mass ratio of particles and the number of physical

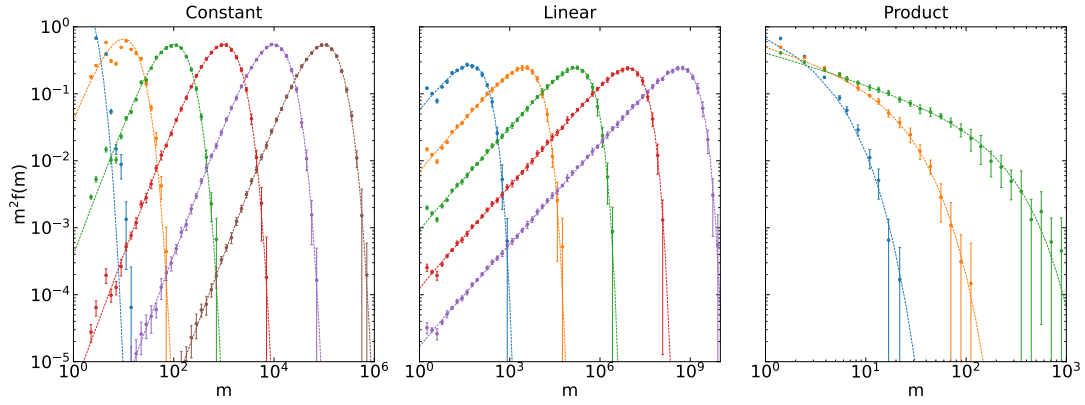


Fig. 2. Particle mass distribution for the high-resolution tests against the analytical solutions of the Smoluchowski equation (dotted lines) evaluated at different dimensionless times. We used 10^4 representative particles and the merging parameter $X = 0.01$. We repeated each simulation ten times. Left: constant kernel $K_{ij} = 1$, evaluated at times 1, 10, 10^2 , 10^3 , 10^4 , and 10^5 . Center: linear kernel $K_{ij} = \frac{1}{2}(m_i + m_j)$, evaluated at times 4, 8, 12, 16, and 20. Right: product kernel $K_{ij} = m_i \times m_j$, evaluated at times 0.4, 0.7, and 0.9.

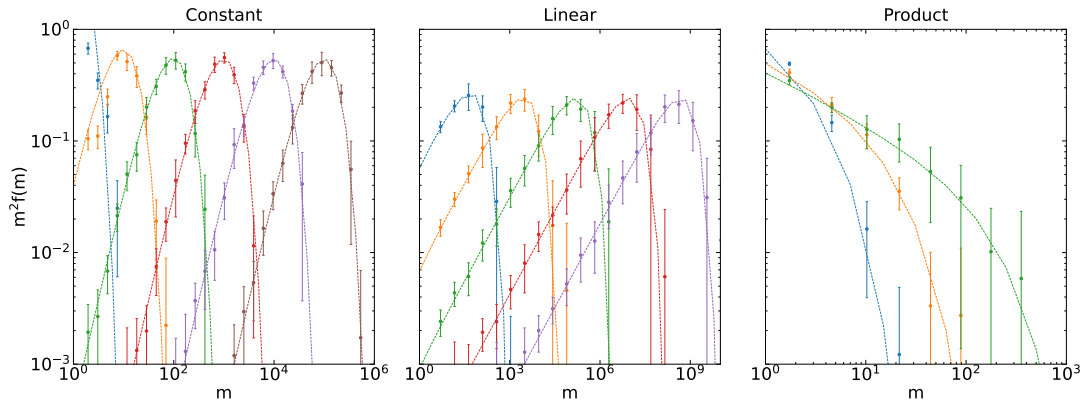


Fig. 3. Particle mass distribution for the low-resolution tests against the analytical solutions of the Smoluchowski equation (dotted lines) evaluated at dimensionless times. We used 200 representative particles and the merging parameter $X = 0.1$. We repeated each simulation ten times and evaluated the same time instances, as in Fig. 2.

particles: if $N_i / N_j \leq m_j / m_i$, $n_{\text{group}} = (m_i / m_j) dm_{\text{max}}$; otherwise, $n_{\text{group}} = (N_j / N_i) dm_{\text{max}}$.

We set $dm_{\text{max}} = 10^{-2}$ for all tests in Sect. 3. This value results in a 1% error in particle mass, because when particle i sticks to, for example, $n_{\text{group}} = (m_i / m_j) dm_{\text{max}}$ j -like particles, the particle mass i is updated as

$$m'_i = m_i + n_{\text{group}} m_j = m_i + \left(\frac{m_i}{m_j} dm_{\text{max}} \right) m_j = (1 + dm_{\text{max}}) m_i. \quad (25)$$

3. Tests

3.1. Analytical solutions of the Smoluchowski equation

The Smoluchowski equation, commonly used to describe dust coagulation in astrophysics (Smoluchowski 1916), has analytical solutions for three specific forms of the coagulation kernel: the constant kernel $K_{ij} = 1$, the linear kernel $K_{ij} = \frac{1}{2}(m_i + m_j)$, and the product kernel $K_{ij} = m_i \times m_j$. Our coagulation model is required to reproduce these analytical solutions. We performed the analytical kernel tests as in Zsom & Dullemond (2008) and Drążkowska et al. (2013). We initialized all particles with a dimensionless mass $m_0 = 1$. We performed high- and low-resolution tests (see Figs. 2 and 3) and compared them against the analytical solutions given in Eqs. (2.1)–(2.3) of Tanaka & Nakazawa (1994).

For the high-resolution test, we used 10^4 representative particles, each initially representing the same number of real particles. We adopted the merging parameter $X = 0.01$ from Sect. 2.3 and repeated each simulation ten times. Figure 2 shows that in the high-resolution simulations, our algorithm accurately describes the evolution of the particle mass distribution over five orders of magnitude for the constant and linear kernels, and over four orders of magnitude for the product kernel. A notable feature seen in the constant kernel case is that the first two time steps are less accurate than the rest. This behavior arises because we assume that the system remains nearly unchanged over the time interval during which one-to-one collisions are grouped (see Sect. 2.2). Collisions between similar-sized particles produce the largest relative changes in the system, and the initial stages correspond to the largest number of real particles (see Eq. (1)). As the mass distribution shifts toward larger masses, each group contains fewer particles, and fewer one-to-one collisions are grouped, which leads to improved accuracy. We conclude that this assumption has only a minor impact on the overall evolution of the system.

We then reduced the resolution to 200 particles and increased the merging parameter to $X = 0.1$ to assess the performance of the algorithm at low particle numbers. As in the previous test, the simulations were repeated ten times, and the results are presented in Fig. 3. As expected, the results are noisier than in the high-resolution simulation. However, we see that the algorithm can

still describe the overall evolution for the three kernels. We find that below 200 particles the linear and product kernels exhibit deviations from the analytical solutions. Therefore, we conclude that our algorithm requires at least 200 representative particles, consistent with the findings of Drążkowska et al. (2013). We further note that reducing the merging parameter X below 0.01 requires a larger number of particles to resolve the linear kernel accurately.

3.2. Dust evolution in global protoplanetary disks

In this section, we describe the ability of our method to capture dust evolution in a two-dimensional protoplanetary disk, including dust transport, growth, and fragmentation. To compute collisions between nearby particles, we used the adaptive radial–vertical grid described in Drążkowska et al. (2013). We compared our algorithm with the publicly available code `mcdust` (Vaikundaraman et al. 2025b), using the same prescriptions for dust transport and grid construction. The coagulation algorithm currently implemented in `mcdust` is based on Zsom & Dullemond (2008) approach.

Overall, the `mcdust` algorithm works as follows. Dust particles are initialized and an initial grid is constructed with an equal number of representative particles per cell. Particle velocities are then computed, including radial drift, vertical settling, and radial and vertical turbulent diffusion (see Eqs. (7)–(18) in Drążkowska et al. 2013). The global time step Δt_{global} is set by the Courant condition, $\Delta t < \Delta x/v_{\text{max}}$, ensuring that particles do not cross grid-cell boundaries within a single time step (see Sect. 2.5 of Drążkowska et al. 2013). The global timescale is further constrained by the collisional timescale by limiting the average number of collisions per particle, scaling as $\sim \Delta t'_{\text{global}}(n_{\text{cell}}/n_{\text{coll}})$, where n_{cell} and n_{coll} denote the number of particles and collisions per cell during the previous time step. Once Δt_{global} is determined, particles are transported over this time step. A new grid is then constructed from the updated positions, and collisions are computed within each cell. The time between consecutive collisions, τ , is calculated via random sampling (see Eq. (10)) and accumulated until Δt_{global} is reached. The global time step is then updated and the procedure repeated.

For the algorithm presented here, we tested merging parameters of $X = 0.1, 0.01, \text{ and } 0.001$ in independent simulations, and Eqs. (17) and (18) were evaluated for each bin of the grid. Since each bin represents a single location in the disk, group positions remained unchanged during coagulation: when $\{j\}$ sticks to $\{i\}$, merges with $\{w\}$, or is filled with half of particles of $\{w\}$, the resulting groups $\{i'\}$, $\{w'\}$, and $\{j'\}$ remain at the original locations of $\{i\}$, $\{w\}$, and $\{j\}$, respectively.

In the current `mcdust`, when two representative particles collide at velocities above the fragmentation threshold, only one representative particle undergoes fragmentation. In reality, collision outcomes also depend on the mass ratio of particles (e.g., Güttler et al. 2010; Hasegawa et al. 2021, 2023), but here we neglect it for simplicity. The new mass of the representative particle is chosen by randomly sampling from a fragment mass distribution $n(m) \propto m^{-11/6}$ (Birnstiel et al. 2011; Drążkowska & Dullemond 2014), and fragments can be composed of multiple monomers or continuous sections rather than being limited to discrete monomers. To construct a comparable model to `mcdust`, we assume that when groups $\{i\}$ and $\{j\}$ collide (with $N_i \leq N_j$), all particles in group $\{i\}$ fragment. Since there are N_i real collision events, not all particles in group $\{j\}$ fragment in reality. In our method, however, all particles belonging to the same group evolve identically, so we assign a probability of N_i/N_j that all

particles in group $\{j\}$ fragment; otherwise, they remain intact. After a group fragments, the collision rates in Eq. (6) of pairs involving that group are recalculated, and Eqs. (8) and (9) are updated accordingly.

To compare the global dust evolution using the two algorithms, we ran independent simulations with the same setup and resolution. We initialized a disk of 100 AU in size with a static gas surface density and temperature given by the power laws

$$\Sigma_g = 800 \left(\frac{r}{\text{AU}} \right)^{-1} \text{ g cm}^{-2}, \quad (26)$$

$$T = 280 \left(\frac{r}{\text{AU}} \right)^{-1/2} \text{ K}, \quad (27)$$

where r is the radial distance from a Sun-like star. The turbulence parameter was set to $\alpha = 10^{-3}$. For the vertical structure, we assumed a Gaussian gas density profile and an isothermal temperature structure. We employed 256 representative particles in each bin in the collision grid, with a total of 200 radial and 20 vertical bins. We began the simulation with a global vertically integrated dust-to-gas ratio of 0.01, with monomers of $1 \mu\text{m}$ in radius and an internal density of 1 g cm^{-3} . We assumed that particles fragment when they collide at velocities above 10 m s^{-1} . Studies on water-ice aggregates support this threshold value (e.g., Wada et al. 2013; Gundlach & Blum 2015), although its temperature dependency remains under debate (e.g., Gundlach et al. 2018; Musiolik & Wurm 2019). While silicates have a higher internal density than water ice and are expected to fragment at lower velocities (Güttler et al. 2010), we simplified our approach by assuming a single component throughout the disk.

Figure 4 compares the two methods at a snapshot of 10^4 yr. At this point, particles in the inner regions of the disk had already grown to sizes at which fragmentation occurs. In the outer regions, where collisional timescales are longer, dust had not yet reached any growth barrier. Since particles initially grow by adding one monomer at a time, there are no particles with sizes between 1 and $2 \mu\text{m}$ in the outer regions (beyond 20–30 AU). In contrast, in the inner regions fragmentation replenishes the population of 1 – $2 \mu\text{m}$ particles, because, as previously stated, fragmentation is not limited to discrete monomers in this work.

Overall, the two methods yield very similar results, and many of the small differences between the snapshots in Fig. 4 are attributable to the stochastic nature of the Monte Carlo simulations. The new method achieves modest improvement in the resolution of small particles. In the approach of Zsom & Dullemond (2008), each representative particle carries the same total mass, which results in small grains being represented by only a few representative particles. In contrast, our method allows the total mass of each group to vary, enabling a larger number of representative particles to sample the small-size regime. The trade-off is a reduced number of representative particles at the largest sizes. The merging parameter X (see Sect. 2.3) plays a central role in this balance: low values of X enable larger variation in group mass, resulting in fewer merging events and consequently splitting large groups less often. For applications such as the synthetic observations of protoplanetary disks, where the opacity may be dominated by small dust grains, lower values of X are therefore particularly suitable, although $X < 0.01$ requires a particle resolution of more than 200 particles per grid cell (see Sect. 3.1). In this respect, the method presented here enables greater flexibility.

All simulations in Fig. 4 were evolved up to 10^4 yr on identical computational resources (128 CPU cores), each requiring

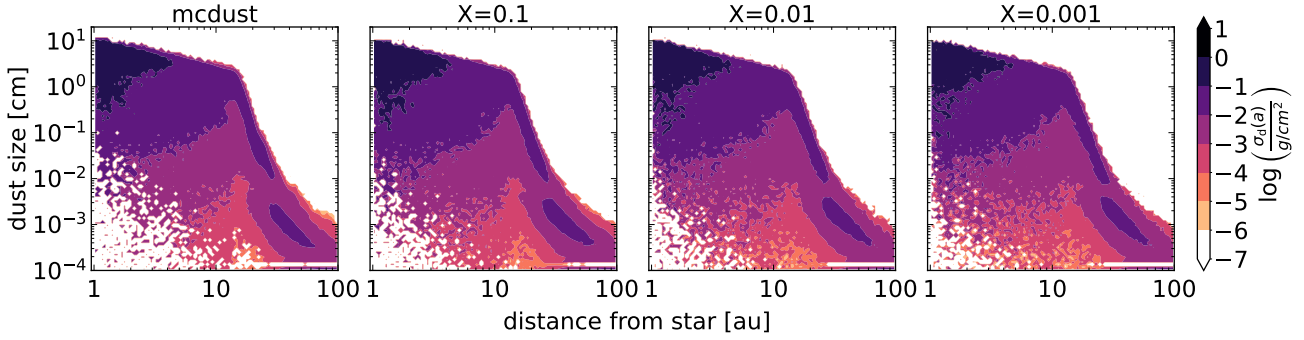


Fig. 4. Dust size distribution in two-dimensional protoplanetary disks. The color map shows the logarithmic, size-dependent dust surface density $\sigma_d(a)$, computed with 100 radial and dust-size bins. Left panel: reference simulation using the open-source code `mcdust`. Remaining panels: results obtained with the algorithm presented in this work, adopting the merging parameters $X=0.1$, 0.01 , and 0.001 (see Sect. 2.3). For $X=0.01$ and 0.001 , the small-size range is more accurately resolved, whereas larger X values reduce the resolution in this regime. The difference between $X=0.01$ and 0.001 is minor, since a larger number of particles is required to fully benefit from values of $X < 0.01$.

approximately 3 h of wall-clock time. For the setups considered in this study, the two algorithms therefore exhibit comparable computational cost.

3.3. Conservation of the global budget: water and silicates

To demonstrate that our algorithm conserves the global budget of particle properties, we built the following toy model. We considered a zero-dimensional simulation with 10^3 representative particles composed of silicates and water ice at a radial distance of 5 AU. For the same disk parameters as in Sect. 3.2, this location lies beyond the water ice line. We triggered an outburst-like event by increasing the disk temperature by a factor of 3. The increase in temperature shifts the water-ice line beyond 5 AU, leading to ice sublimation in the simulated region. We assumed that the silicate aggregates survive sublimation without fragmenting back into monomers (Houge & Krijt 2023; Houge et al. 2024). After the outburst, the water vapor recondenses back onto the silicate grains.

The recondensation of water onto dust grains within a volume V occurs when the vapor pressure in that volume exceeds the saturation vapor pressure. The volume density for saturation is (Supulver & Lin 2000),

$$\rho_{\text{sat}} = \frac{m_{\text{H}_2\text{O}}}{k_B T} \cdot 1.013 \times 10^6 \exp\left[15.6 - \frac{5940 \text{ K}}{T}\right] \text{ dyn/cm}^2, \quad (28)$$

where $m_{\text{H}_2\text{O}}$ is the mass of water molecules. Vapor deposition onto an existing icy surface is energetically more favorable than the formation of an initial ice monolayer on a silicate grain (Ros et al. 2019). Although the distinction between icy and non-icy particles during recondensation can have strong implications on the dust size distribution after an outburst (Ros & Johansen 2024), for this test case, we assumed that water condensation is independent of the grain's surface composition. We included size-dependent condensation and evaporation of water on dust particles, following the approach of Krijt et al. (2016). Let $\rho_{\text{H}_2\text{O}}$ be the volume density of water vapor. When $\rho_{\text{H}_2\text{O}} > \rho_{\text{sat}}$, every particle gains water ice, $m_{\text{ice},i}$, over the global time step Δt_{global} ,

$$\Delta m_{\text{ice},i} = (1 - e^{-\Delta t_{\text{global}}/\tau_{\text{con}}}) (\rho_{\text{H}_2\text{O}} - \rho_{\text{sat}}) \frac{a_i^2}{\sum_{i=1}^{N_r} N_i a_i^2} V, \quad (29)$$

where τ_{con} is the condensation timescale, a_i is the particle radius, and N_i is the number of physical particles in group $\{i\}$. τ_{con} is

defined as

$$\tau_{\text{con}} = \frac{V}{\pi v_{\text{th}} \sum_{i=1}^{N_r} N_i a_i^2}, \quad (30)$$

where $v_{\text{th}} = (8k_B T / \pi m_{\text{H}_2\text{O}})^{1/2}$ is the thermal velocity of water molecules. When $\rho_{\text{H}_2\text{O}} < \rho_{\text{sat}}$, every particle loses water ice over the time step Δt_{global} as

$$\Delta m_{\text{ice},i} = -\min\left(\pi a_i^2 v_{\text{th}} (\rho_{\text{sat}} - \rho_{\text{H}_2\text{O}}) \Delta t_{\text{global}}, m_{\text{ice},i}\right). \quad (31)$$

Consequently, the total particle mass and radius, m_i and a_i , increase or decrease during condensation or sublimation. Since in the method by Zsom & Dullemond (2008) the total mass of every representative particle, $M_i = N_i m_i$, is equal, if mass changes, the number of physical particles N_i must change accordingly. However, during sublimation and recondensation the particle number should remain constant. Therefore, a more appropriate assumption is to keep the total silicate mass of every particle, $M_{\text{si},i} = N_i m_{\text{si},i}$, constant, since $m_{\text{si},i}$ remains constant during condensation or sublimation. In that case, the total silicate mass is conserved, whereas the total mass of water may fluctuate during collisions. In addition, maintaining $M_{\text{si},i}$ constant does not allow for tracking particles that contain only ice. On the other hand, the method presented in this work does not have this limitation, as M_i is allowed to vary.

To test the distribution of water during collisions after an outburst, we ran two independent simulations: one using the coagulation algorithm presented in this work, and another using the algorithm of Zsom & Dullemond (2008). We considered sticking and fragmentation during collisions, as described in Sect. 3.2, with the outcome determined by the fragmentation velocity of 10 m s^{-1} . In reality, silicate aggregates may fragment at velocities an order of magnitude lower (Güttler et al. 2010). However, we adopted a single fragmentation threshold to simplify the model.

Throughout the simulations, we tracked the total particle mass, m_i , along with the ice fraction of dust particles, defined as

$$f_i = \frac{M_{\text{ice},i}}{M_i} = \frac{m_{\text{ice},i}}{m_i}. \quad (32)$$

We then calculated the component masses as $m_{\text{ice},i} = f_i m_i$ and $m_{\text{si},i} = (1 - f_i) m_i$. In the algorithm presented in this paper, during sticking collisions between group $\{i\}$ and $\{j\}$, we update the

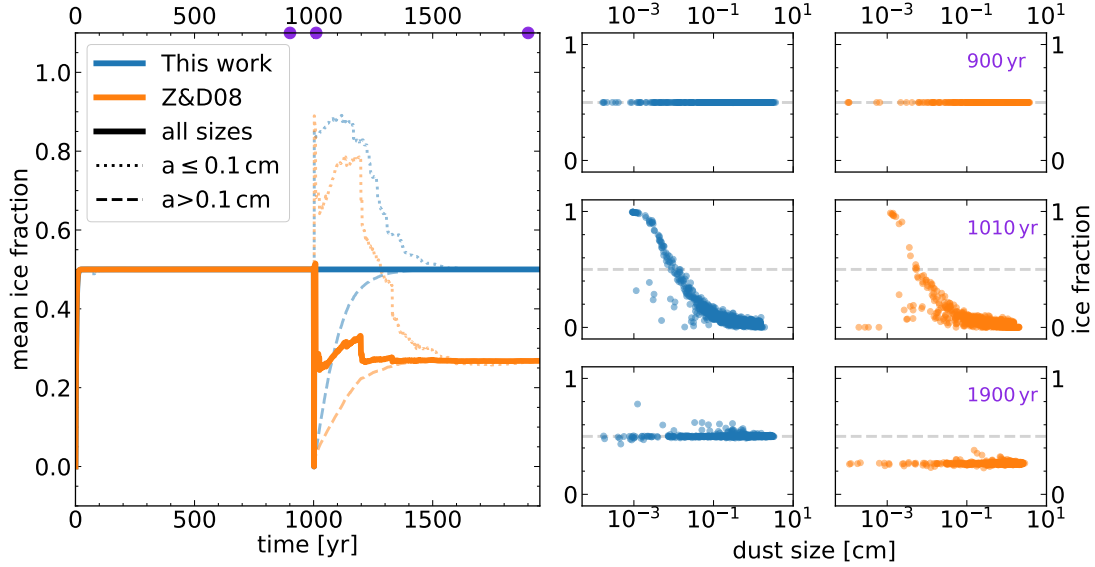


Fig. 5. Left: mean water ice fraction of the dust population before and after an outburst-like event triggered at 1000 yr. Two independent simulations are shown: the coagulation algorithm (blue line) presented in this work and the algorithm of *Zsom & Dullemond (2008)* (orange line). The solid line indicates the average ice fraction over all particle sizes, the dotted line the average for small particles, and the dashed line the average for large particles. Initially, water vapor condenses almost instantaneously into particles of equal size. Later, ice sublimates during the outburst event that lasts 1 yr. Upon recondensation onto silicate grains of varying sizes, the coagulation method presented here exactly conserves the mean ice fraction, while the method by *Zsom & Dullemond (2008)* does not. Right: water ice fraction and sizes of every representative particle at different times for the two simulations. Before the outburst (900 yr), water is equally distributed across particles of different sizes. Following the outburst (1010 yr), small particles gain more water in proportion. At later times (1900 yr), the water fraction is again nearly equally distributed across different sizes.

properties as

$$f'_i = \frac{M'_{ice,i}}{M'_i} = \frac{N_i m_i f_i + N_j m_j f_j}{N_i m_i + N_j m_j} = \frac{m_i f_i + m_j f_j}{m_i + m_j}, \quad (33)$$

$$f'_j = \frac{M'_{ice,j}}{M'_j} = \frac{N_j m_j f_j - N_i m_j f_j}{N_j m_j - N_i m_j} = f_j, \quad (34)$$

and thus the total ice mass budget $M_{ice,i} + M_{ice,j}$ in a sticking event is conserved (see Eqs. (13)–(16)). If group $\{j\}$ contains negligible mass after transferring mass to group $\{i\}$ (see Eq. (17)), it merges with its most similar group $\{w\}$ (see Eq. (19)), and the resulting ice fraction is

$$f'_w = \frac{M'_{ice,w}}{M'_w} = \frac{N_w m_w f_w + N_j m_j f_j}{N_w m_w + N_j m_j}. \quad (35)$$

We chose a merging parameter of $X = 0.01$ for this simulation.

Assuming that the aggregates are compact, spherical, and well mixed in composition, the internal density depends on the ice fraction as well as the ice and silicate densities as

$$\rho_i = \left(\frac{f_i}{\rho_{ice}} + \frac{1 - f_i}{\rho_{silicate}} \right)^{-1}. \quad (36)$$

We set $\rho_{ice} = 1 \text{ g cm}^{-3}$ and $\rho_{silicate} = 3 \text{ g cm}^{-3}$. In reality, aggregates may not be well mixed, and consequently, the composition of fragments can vary (*Krijt et al. 2024*). Although Monte Carlo methods can track variations in fragment composition (e.g., *Gurrutxaga et al. 2026*), further refinement is required to ensure conservation of each component's total mass to track heterogeneous fragmentation. In this study, we neglected this and assumed that the ice fraction, f_i , remains constant during fragmentation.

In this setup, particles are expected to coagulate to centimeter sizes before fragmenting. We therefore initialized the simulation with centimeter-sized silicate particles settled in the midplane, corresponding to an initial Stokes number of $St \approx 0.03$. For a vertically integrated dust-to-gas ratio of $Z_{si} = 0.005$ and turbulence of $\alpha = 10^{-3}$, the midplane dust-to-gas ratio of these dust population after vertical settling is (*Youdin & Lithwick 2007*)

$$\epsilon_{si} = \sqrt{\frac{\alpha + St}{\alpha}} Z_{si} \approx 0.028, \quad (37)$$

which was expected to remain roughly constant for our simulated test case. We thus assumed a constant volume V and a fixed number of representative particles. We also assumed that the initial vapor density at the midplane equals the dust volume density.

After initializing the simulation, we evolved the dust population for 1000 yr to reach a coagulation–fragmentation equilibrium. We then triggered a brief outburst of 1 yr and let the simulation stabilize again for another 1000 yr. We neglected dust transport, and therefore the global time step Δt_{global} was set solely by the collisional timescale (see Sect. 3.2). However, we imposed an additional constraint on the global timescale by setting a maximum of 1 yr to resolve the temperature change due to the outburst and the resulting sublimation and recondensation.

Figure 5 shows that for our disk parameters at 5 AU, water condenses almost instantaneously onto centimeter-sized silicate grains, since our initial condensation timescale is ~ 7 yr (see Eq. (30)). Because all particles initially have the same size, each silicate grain gains the same water content, and the total ice mass is conserved in both algorithms during coagulation (see Eq. (29)). Subsequently, an outburst-like event is triggered at 1000 yr. The rise in temperature increases the saturation vapor density (see Eq. (28)), and consequently, water ice sublimates. After the outburst, water vapor recondenses onto the silicate grains. At this stage, particle sizes are no longer equal, and

smaller grains acquire a higher ice mass fraction (see Eq. (29); and also Fig. 5, right panels). In both simulations considered here, the dust population requires approximately 500 yr to redistribute the ice uniformly after the outburst (see Fig. 5, dashed and dotted lines), which is faster than the time reported by Houge & Krijt (2023) for their resilient model. This difference in re-equilibration time is likely due to our choice of a higher midplane dust-to-gas ratio (see Eq. (37)).

During collisions between particles with different ice contents, the algorithm of Zsom & Dullemond (2008) updates only one of the colliding representative particles and therefore does not conserve the total water content, leading to numerical fluctuations (see Fig. 5, solid orange line). While these fluctuations would decrease with increasing particle number (Houge & Krijt 2023), simulations with such high numbers of representative particles per grid cell in global disks are currently computationally too expensive. By contrast, we demonstrate that the new algorithm conserves water and silicate mass independently, without introducing such fluctuations.

4. Conclusions

We presented a Monte Carlo method for computing dust coagulation in protoplanetary disks. Although derived independently, the method is conceptually equivalent to the Monte Carlo approach introduced by Shima et al. (2009) for cloud microphysics. A key strength of this approach is its ability to conserve the global inventory of dust properties while remaining applicable to global disk simulations.

We first validated the algorithm by benchmarking it against analytical solutions of the Smoluchowski equation. We demonstrated that the method successfully reproduces the analytical results and that it requires a minimum of 200 representative particles (see Figs. 2 and 3).

We then implemented the method in a two-dimensional protoplanetary disk model and compared it with the algorithm of Zsom & Dullemond (2008). We find that both approaches yield similar results when the same numerical resolution is employed (see Fig. 4). We introduced the merging parameter X in Sect. 2.3, which can be adjusted to concentrate resolution on the largest particles (e.g., $X \geq 0.1$) or to distribute it more evenly across the size distribution (e.g., $X < 0.1$). We note, however, that decreasing X requires a larger number of representative particles to resolve dust evolution, since collisions are typically dominated by the largest particles in a protoplanetary disk.

By tracking the composition of a dust population undergoing sublimation and recondensation events, we demonstrated that the method presented here conserves the global budget of particle properties (see Fig. 5). Overall, this method offers a promising tool for tracking dust properties during coagulation in global protoplanetary disk simulations.

Data and code availability

All data and code required to reproduce the results presented in this work are publicly available. The dataset can be accessed via Zenodo at <https://doi.org/10.5281/zenodo.19390275>. The code used to perform the simulations and generate all figures is available in the GitHub repository at <https://github.com/nereagurru/A-Monte-Carlo-method-for-tracking-dust-properties-during-coagulation-in-protoplanetary-disks>. The original

mcdust codebase is available at <https://github.com/vicky1997/mcdust>.

Acknowledgements. The authors thank the anonymous referee for constructive comments that helped to improve the manuscript. J.D. and V.V. are funded by the European Union under the European Union's Horizon Europe Research & Innovation Programme 101040037 (PLANETOIDS).

References

- Akimkin, V., Ivlev, A. V., Caselli, P., Gong, M., & Silsbee, K. 2023, *ApJ*, **953**, 72
- Beutel, M., & Dullemond, C. P. 2023, *A&A*, **670**, A134
- Birnstiel, T. 2024, *ARA&A*, **62**, 157
- Birnstiel, T., Dullemond, C. P., & Brauer, F. 2010, *A&A*, **513**, A79
- Birnstiel, T., Ormel, C. W., & Dullemond, C. P. 2011, *A&A*, **525**, A11
- Birnstiel, T., Klahr, H., & Ercolano, B. 2012, *A&A*, **539**, A148
- Blum, J., & Münch, M. 1993, *Icarus*, **106**, 151
- Brauer, F., Dullemond, C. P., & Henning, T. 2008, *A&A*, **480**, 859
- Drążkowska, J., & Dullemond, C. P. 2014, *A&A*, **572**, A78
- Drążkowska, J., Windmark, F., & Dullemond, C. P. 2013, *A&A*, **556**, A37
- Dullemond, C. P., & Dominik, C. 2005, *A&A*, **434**, 971
- Gillespie, D. T. 1975, *J. Atmos. Sci.*, **32**, 1977
- Gundlach, B., & Blum, J. 2015, *ApJ*, **798**, 34
- Gundlach, B., Schmidt, K. P., Kreuzig, C., et al. 2018, *MNRAS*, **479**, 1273
- Gurrutxaga, N., Drążkowska, J., Vaikundaraman, V., et al. 2026, arXiv e-prints [arXiv:2604.16604]
- Güttler, C., Blum, J., Zsom, A., Ormel, C. W., & Dullemond, C. P. 2010, *A&A*, **513**, A56
- Hasegawa, Y., Suzuki, T. K., Tanaka, H., Kobayashi, H., & Wada, K. 2021, *ApJ*, **915**, 22
- Hasegawa, Y., Suzuki, T. K., Tanaka, H., Kobayashi, H., & Wada, K. 2023, *ApJ*, **944**, 38
- Houge, A., & Krijt, S. 2023, *MNRAS*, **521**, 5826
- Houge, A., Macías, E., & Krijt, S. 2024, *MNRAS*, **527**, 9668
- Kelling, T., Wurm, G., & Köster, M. 2014, *ApJ*, **783**, 111
- Krijt, S., Ormel, C. W., Dominik, C., & Tielens, A. G. G. M. 2015, *A&A*, **574**, A83
- Krijt, S., Ciesla, F. J., & Bergin, E. A. 2016, *ApJ*, **833**, 285
- Krijt, S., Arakawa, S., Oosterloo, M., & Tanaka, H. 2024, *MNRAS*, **534**, 2125
- Lambrechts, M., & Johansen, A. 2012, *A&A*, **544**, A32
- Liffman, K. 1992, *J. Comput. Phys.*, **100**, 116
- Musioli, G., & Wurm, G. 2019, *ApJ*, **873**, 58
- Okuzumi, S., Tanaka, H., & Sakagami, M.-a. 2009, *ApJ*, **707**, 1247
- Ormel, C. W., & Klahr, H. H. 2010, *A&A*, **520**, A43
- Ormel, C. W., & Spaans, M. 2008, *ApJ*, **684**, 1291
- Ormel, C. W., Spaans, M., & Tielens, A. G. G. M. 2007, *A&A*, **461**, 215
- Ossenkopf, V. 1993, *A&A*, **280**, 617
- Robinson, A., Booth, R. A., & Owen, J. E. 2024, *MNRAS*, **529**, 1524
- Ros, K., & Johansen, A. 2024, *A&A*, **686**, A237
- Ros, K., Johansen, A., Riipinen, I., & Schlesinger, D. 2019, *A&A*, **629**, A65
- Schräpler, R., & Blum, J. 2011, *ApJ*, **734**, 108
- Shima, S., Kusano, K., Kawano, A., Sugiyama, T., & Kawahara, S. 2009, *Quarterly J. R. Meteor. Soc.*, **135**, 1307
- Smoluchowski, M. V. 1916, *Z. Phys.*, **17**, 557
- Stammler, S. M., & Birnstiel, T. 2022, *ApJ*, **935**, 35
- Stammler, S. M., Birnstiel, T., Panić, O., Dullemond, C. P., & Dominik, C. 2017, *A&A*, **600**, A140
- Supulver, K. D., & Lin, D. N. C. 2000, *Icarus*, **146**, 525
- Tanaka, H., & Nakazawa, K. 1994, *Icarus*, **107**, 404
- Vaikundaraman, V., Drążkowska, J., Binkert, F., Birnstiel, T., & Miotello, A. 2025a, *A&A*, **696**, A215
- Vaikundaraman, V., Gurrutxaga, N., & Drążkowska, J. 2025b, arXiv e-prints [arXiv:2507.21239]
- Wada, K., Tanaka, H., Okuzumi, S., et al. 2013, *A&A*, **559**, A62
- Weidenschilling, S. J. 1980, *Icarus*, **44**, 172
- Youdin, A. N., & Goodman, J. 2005, *ApJ*, **620**, 459
- Youdin, A. N., & Lithwick, Y. 2007, *Icarus*, **192**, 588
- Zsom, A., & Dullemond, C. P. 2008, *A&A*, **489**, 931
- Zsom, A., Ormel, C. W., Güttler, C., Blum, J., & Dullemond, C. P. 2010, *A&A*, **513**, A57