

An improved Representative Particle Monte Carlo method for the simulation of particle growth

M. Beutel[✉] and C. P. Dullemond[✉]

Institute of Theoretical Astrophysics (ITA), Center for Astronomy (ZAH), Ruprecht-Karls-Universität Heidelberg, Albert-Ueberle-Str. 2, 69120 Heidelberg, Germany
e-mail: moritz.beutel@ziti.uni-heidelberg.de

Received 12 September 2022 / Accepted 21 November 2022

ABSTRACT

Context. A rocky planet is formed out of the agglomeration of around 10^{40} cosmic dust particles. As dust aggregates grow by coagulation, their number decreases. But until they have grown to hundreds of kilometres, their number still remains well above the number of particles a computer model can handle directly. The growth from micrometres to planetesimal-sized objects therefore has to be modelled using statistical methods, often using size distribution functions or Monte Carlo methods. However, when the particles reach planetary masses, they must be treated individually. This can be done by defining two classes of objects: a class of many small bodies or dust particles treated in a statistical way, and a class of individual bodies such as one or more planets. This introduces a separation between small and big objects, but it leaves open how to transition from small to big objects, and how to treat objects of intermediate sizes.

Aims. We aim to improve the Representative Particle Monte Carlo (RPMC) method, which is often used for the study of dust coagulation, to be able to smoothly transition from the many-particle limit into the single-particle limit.

Results. Our new version of the RPMC method allows for variable swarm masses, making it possible to refine the mass resolution where needed. It allows swarms to consist of few numbers of particles, and it includes a treatment of the transition from swarm to individual particles. The correctness of the method for a simplified two-component test case is validated with an analytical argument. The method is found to retain statistical balance and to accurately describe runaway growth, as is confirmed with the standard constant kernel, linear kernel, and product kernel tests as well as by comparison with a fiducial non-representative Monte Carlo simulation.

Key words. planets and satellites: formation – protoplanetary disks – methods: numerical – methods: statistical – accretion, accretion disks

1. Introduction

How planets are formed is a difficult question to answer. We know that the process starts with micrometre-sized cosmic dust particles in a protoplanetary disk, which coagulate and form ever larger dust aggregates (Birnstiel et al. 2016). As these aggregates grow to a pebble size (of the order of millimetres to centimetres, henceforth conveniently called pebbles), they may, under certain conditions, form large overdensities in the disk that gravitationally collapse to form multi-kilometre sized planetesimals (Johansen et al. 2009; Wahlberg Jansson & Johansen 2014). From there onwards, these planetesimals agglomerate gravitationally to form planetary embryos (Wetherill & Stewart 1993; Kokubo & Ida 1998), which, in turn, grow further via accretion of planetesimals and any remaining pebbles (Safronov 1964; Ormel & Klahr 2010; Lambrechts & Johansen 2012; Bitsch et al. 2015).

One of the fundamental limiting factors of any numerical treatment of this problem is the vast number of particles involved. To form a single Earth-like planet, we need $\sim 10^{40}$ dust particles, $\sim 10^{30}$ pebbles, or $\sim 10^{10-20}$ planetesimals (dependent on their mass). This would not necessarily be a problem if we could treat these as an equilibrium fluid, just in the way we treat a gas. Unfortunately, solid particles behave very differently than a gas, and thus require a more sophisticated treatment.

For small particles, the problem is often approached with a particle size distribution function $n(m)dm$, giving the number of

particles per unit volume for a particle mass interval dm . This function is then sampled numerically on a grid in m , and the growth and fragmentation are then modelled using a version of the Smoluchowski coagulation equation (Smoluchowski 1916). This method (which we call the continuum approach) has been used by numerous teams (e.g. Weidenschilling 1980; Tanaka et al. 2005; Dullemond & Dominik 2005) and is reasonably fast and efficient. Furthermore, it can be extended in space by adding the spatial dimensions to the coordinates (Birnstiel et al. 2010; Okuzumi et al. 2012; Drażkowska et al. 2019). It works very well for small particles because they are well coupled to the gas. These particles do not have independent orbital elements, but instead largely move along with the gas, except for a slow drift. However, with this method, it is hard to add independent particle properties. By the Eulerian nature of such a grid-based approach, all particles have the same properties for each location and mass. Approximate methods exist to overcome this limitation (Okuzumi et al. 2009; Stammler et al. 2017), but a full solution would require an extension of the dimensionality of the problem, creating a huge computational overhead. For instance, by adding only a porosity parameter given by the value p , the particle distribution function now becomes $n(m, p) dm dp$. This quickly pushes the problem beyond the limit of computational feasibility.

For large particles (planetesimals and upwards), the problem becomes even more severe, because now the particles are large enough to acquire their own independent orbital elements.

A distribution function treatment as described above becomes unfeasible. Instead, the N -body method is the method of choice.

As a result of these (and other) complications, it is hard to develop a model of planet formation that starts with dust and ends with full-grown planets. The few complete dust-to-planets models available today involve a patchwork of different methods covering different size regimes (e.g. Ormel et al. 2017; Schneider & Bitsch 2021; Emsenhuber et al. 2021).

If we want to develop a single technique that can, at least in principle, handle large amounts of dust particles as well as individual planet-like bodies, and everything in between, the method must be based on a particle sampling approach. This could be a Monte Carlo method (stochastic), an N -body method (deterministic), or a blend of both. In this approach, it is easy to add any number of particle properties to the problem, because we simply store these properties for each computational particle. Even if all particles are in the same location and have the same mass, we can then still handle a heterogeneous set of supplementary properties (porosity, charge, chemical composition, orbital elements, etc.), which is not possible for continuum methods.

Ormel et al. (2007) developed a Monte Carlo solver for dust coagulation in which particles can have not only a mass but also other properties such as porosity or ice content. This method stands at the basis of further works (e.g. Krijt et al. 2015) and overcomes the limitations of the continuum methods. While the method was developed for small particles, it has also been used for modelling runaway growth of planetesimal size bodies (Ormel et al. 2010).

An alternative Monte Carlo method was developed by Zsom & Dullemond (2008). Their key difference with the method of Ormel et al. (2007) is that it is rigorously based on the concept of ‘representative particles’. We therefore call this method the ‘Representative Particle Monte Carlo’ (RPMC) method in this paper, and provide a brief review of the method in Sect. 2. This method has several advantages over other methods, one of them being its robustness and simplicity. But it has, so far, only been formulated in a somewhat restrictive sense: (1) All swarms must have the same mass, and (2) all swarms must contain a very large number of actual particles. As a result, this method has so far only been applied to problems of dust coagulation and fragmentation, where the number of actual particles is so large that condition (2) is always fulfilled. In this context the method has been successfully used for modelling dust coagulation, compactification, and fragmentation based on the full complexity of laboratory measurements (Zsom et al. 2010, 2011a). It has also been used for exploring the effect of external disturbances on the protoplanetary disk (Zsom et al. 2011b), the effect of particle collisions during the gravitational collapse of a pebble cloud (Wahlberg Jansson et al. 2017), and for various other applications (e.g. Windmark et al. 2012; Drażkowska et al. 2014; Krijt & Ciesla 2016; Krijt et al. 2016).

As the RPMC method stands, however, it cannot be used for problems where the growth proceeds to the point when swarms become individual particles (e.g. planets). In fact, the validity of the method, in the form presented by Zsom & Dullemond (2008), already formally breaks down well before that. And although the equal-mass swarms provide a certain robustness, and focus computational effort where the mass is, this condition can also severely limit the applications of the method. There are well-known particle growth problems where initially insignificant particles can grow to dominate the process, for example the planetesimal runaway growth regime (Greenberg et al. 1978) or the ‘lucky particle’ scenario (Windmark et al. 2012). To model these with a Monte Carlo approach either requires a very large

number of representative particles, or the clever splitting of swarms into smaller swarms to resolve particles of interest. This is only possible if the method allows swarms of different masses.

The goal of this paper is to improve the RPMC method such that the drawbacks (1) and (2) are remedied. The method will then still have the advantages of robustness of the original RPMC method, but will gain in flexibility and applicability so that it can be used to model planet formation over a wide range of particle masses. In a companion paper (Beutel et al., in prep.), we address another problem of the RPMC method: the n^2 scaling of all possible pair-wise interactions.

2. Conceptual summary of the original RPMC method

Let us consider the problem of coagulation or fragmentation of, say, $N = 10^{30}$ particles. Obviously we cannot model all of these particles individually. With the RPMC method we consider, of these 10^{30} particles, only n particles which together form a representative sample of the actual particles. The number n must be large enough that this sample provides a good enough representation of the true distribution. At the same time, it must be small enough to keep the computational cost manageable.

The sampling is mass-weighted. That means that a boulder with a mass of 1000 kg will have a million times higher chance to be one of the representative particles than a pebble of 1 g. However, if there are a million times more pebbles than boulders (meaning that the total mass in pebbles equals the total mass in boulders), then the chance that representative particle i happens to be a pebble is the same as that it happens to be a boulder.

Now let us assume that representative particle i is a boulder of $m_B = 1\,000$ kg. We let it undergo an event that causes it to fragment into a mass distribution $n_{\text{fr}}(m)$. The total mass of the mass distribution is still 1000 kg:

$$\int_0^{m_B} m n_{\text{fr}}(m) dm = m_B. \quad (1)$$

In the RPMC method, rather than splitting the numerical particle up into a large number of new numerical particles, we randomly choose a new mass for the representative particle using the probability distribution function

$$p(m) dm = \frac{m n_{\text{fr}}(m)}{m_B} dm. \quad (2)$$

In other words, we use mass-weighting to choose the new property of the representative particle, so the sampling remains mass-weighted. All the other fragments are ‘forgotten’. If only a single such fragmentation event were to happen, such forgetfulness would be problematic, since a single randomly chosen particle does not represent an entire distribution of fragments. But since we are dealing with very large numbers of boulders, such fragmentation events will occur many times (each time for a different representative particle $j \neq i$). Statistically, the outcomes of many representative particles follow the fragment mass distribution $n_{\text{fr}}(m)$.

The representative particles form a representation of the real (full) collection of particles at a given time. They contain all the information we have of the system at that particular time. In order to reconstruct the full collection of particles from this limited information, the RPMC method follows the principle of Ockham’s Razor: if we do not know more than the information

we currently have of the representative particles, then the simplest assumption possible is that the other (unknown) particles are the same as the representative ones. Given that the representative particles follow a mass-weighted sampling, each of them represents a fraction $1/n$ of the total mass of solids. And so we divide the total mass up into n equal-mass parts, and assume each part to consist of particles with identical properties as their representative particle.

Although it might be possible to conceive different methods of extrapolating the full distribution of particles from the set of representative particles, our simple method has the advantage that it trivially guarantees the idempotency of the selection process. Regardless of the how representative particles are chosen from swarms (random pick, mass median, nearest to average, etc.), we would end up with an exactly equivalent set of representative particles after extrapolating a full particle ensemble and selecting a new set of representative particles.

We note that, so far, we have not used the concept of ‘swarms’. The RPMC method does not require this concept. But to make it easier to talk about the method, and to compare it to the ‘superparticles’ often used in computational astrophysics (e.g. Johansen et al. 2007), it is convenient to define a swarm in the context of RPMC. The key lies in the mass-weighted sampling used by RPMC. If the total mass of solids is M , and we used n representative particles which are randomly sampling the solids in a mass-weighted fashion, then we can say that each representative particle represents a fraction $1/n$ of the total mass of solids. We then assign to each representative particle i a swarm of N_i particles with identical properties as the representative particle. The total mass of the swarm is $M_i = M/n$, so that $N_i = M_i/m_i$, where m_i is the mass of the individual representative particle. If the properties of the representative particle change, so will the properties of the other particles in the swarm. This also means that if the representative particle becomes more massive (i.e. m_i increases) due to, for instance, a collide-and-merge event with another particle, the other swarm particles also become more massive. As the total mass of the swarm stays constant, the number of particles in the swarm N_i will then drop. This is a purely statistical effect, not a true sudden disappearance of particles.

To better understand how the representative particles work, it is helpful to pretend that all particles are composites of the same elementary substance, and that a representative particle is identified with a ‘tracer atom’ of this substance. If tracer atoms were initially chosen at random from the entirety of atoms, then at any time the particle mass distribution of the system must be approximated by the mass distribution of representative particles, assuming that the number of particles is so much larger than the number of tracer atoms that the possibility of two tracer atoms ending up in the same particle can be neglected. In this picture, swarms just embody the simplest possible way of extrapolating an approximate global mass distribution from the mass distribution of representative particles.

We have not yet introduced any collisions between particles. The fragmentation event is, so far, a purely hypothetical spontaneous event affecting only the representative particle itself. The RPMC method can, however, easily handle collisions. Since each swarm consists of truly many particles, when a representative particle of swarm j collides with a particle from swarm k , the probability that it hits the representative particle of swarm k is extremely low. Instead, it will most likely (with a probability close to 1) hit another particle of that swarm, leaving representative particle k alone. A collision nearly always affects both colliding particles. For instance, upon high-velocity impact,

both particles can fragment. Or upon low-velocity impact, the particles can merge. But since with almost certainty the representative particle j will collide with a non-representative particle from swarm k , only representative particle j will fragment or merge, while representative particle k will be entirely unaffected as it is not involved in the collision. This means that only the properties of representative particle j , and by extrapolation those of swarm j , are changed, while swarm k stays unaffected. This asymmetric interaction between swarms is a key feature of the RPMC method. It means that on individual collision basis the results can be skewed, but statistically over many collisions the results are correct.

In Zsom & Dullemond (2008), the method is described in much more detail. It is explained how to randomly sample collision events, how to modify the representative particle involved, how to update the collision matrix, how to deal with pathological cases, etc. We discuss these issues in Sect. 4.3, where we define our improved version of the RPMC method.

An advantage of the RPMC method is that it naturally keeps the number of representative particles unchanged, even as particles merge or shatter. While the actual number of particles may decline (as particles coalesce) or increase (as particles shatter), the number of representative particles stays, by design, the same. This avoids the need of re-introducing computational particles as the number of particles declines (to keep the resolution optimal), or forcefully merging particles as the number of particles increases (to prevent runaway computational cost). Also, it naturally conserves mass. These properties make the RPMC method exceptionally stable and robust. But this robustness comes at a cost. First, the interactions between particles become asymmetric, which is not problematic but can be confusing. Furthermore, the method has the basic assumption of mass-weighted sampling, which automatically means that all swarms have the same total mass, namely M/n . And the method assumes that two representative particles have a negligible chance of colliding with each other, which requires that the swarm mass always is huge compared to the particle mass (many particles per swarm). Together, these disadvantages limit the applicability of the RPMC method. Addressing these shortcomings is the purpose of this paper.

3. Mathematical formalisation of the RPMC method

3.1. Fundamentals

We consider an ensemble of N physical particles which interact through collisions. The number N will typically be very large (say, $N \sim 10^{30}$). Here and in the following, we refer to a ‘physical particle’ as any particle of the system we study, while only a (tiny) subset of these particles, the ‘representative particles’, are stored on the computer. Each physical particle k is associated with a particle mass m_k and possibly other properties. These particle properties, in particular the mass, may span many orders of magnitude. We denote the full set of particle properties of particle k with the symbol $\mathbf{q}_k \in \mathbb{Q}$, where \mathbb{Q} is the space of intrinsic properties (e.g. mass, charge, porosity) and statistical properties (e.g. velocity dispersion).

For every pair of particles j and k , we define the *raw collision rate* $\lambda(\mathbf{q}_j, \mathbf{q}_k)$ as the rate (in units of s^{-1}) at which both particles collide. It is necessarily commutative, $\lambda(\mathbf{q}_j, \mathbf{q}_k) = \lambda(\mathbf{q}_k, \mathbf{q}_j)$. If the particles are distributed homogeneously and isotropically in a volume V , the collision rate $\lambda(\mathbf{q}_j, \mathbf{q}_k)$ can be written as

$$\lambda(\mathbf{q}_j, \mathbf{q}_k) = V^{-1} \sigma(\mathbf{q}_j, \mathbf{q}_k) |\Delta \mathbf{v}(\mathbf{q}_j, \mathbf{q}_k)|, \quad (3)$$

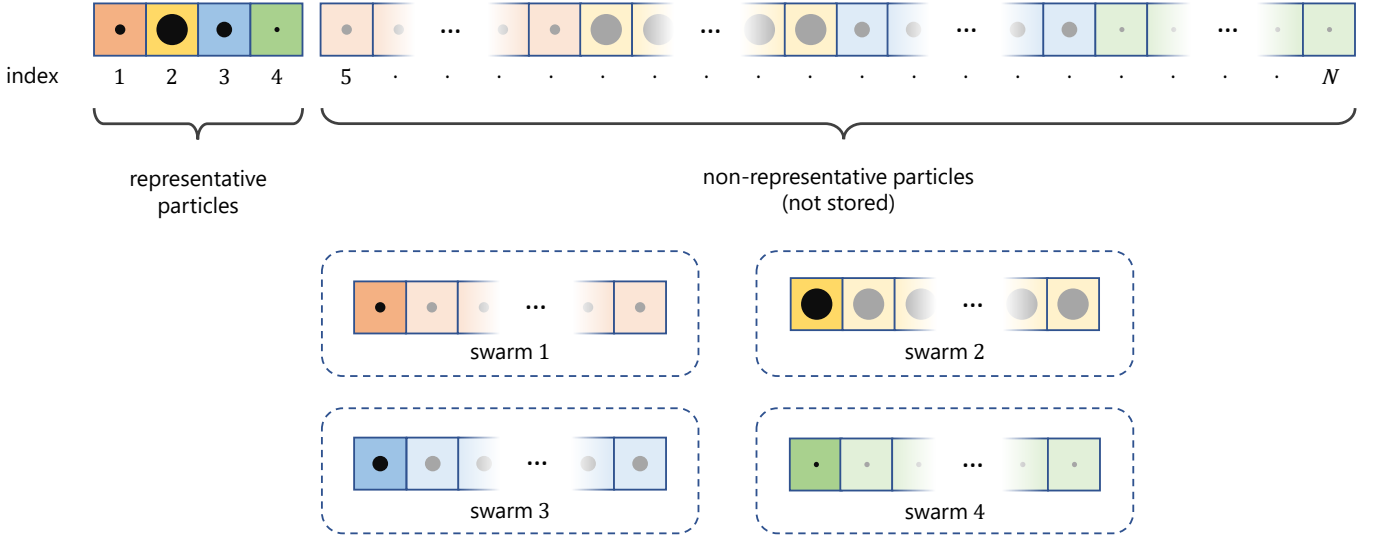


Fig. 1. Indexing of the full ensemble of physical particles, with the representative particles ordered at the beginning of the sequence. The entirety of particles in a given ensemble is indexed from 1 to N . The first n indices refer to RPs, and the remaining indices belong to non-RPs. Because N may be enormously large, tracking the entirety of particles is not feasible. Only the n RPs are actually modelled on the computer, while the properties of the $(N - n)$ non-RPs are estimated from the known properties of the n representative ones.

where $\sigma(\mathbf{q}_j, \mathbf{q}_k)$ is the collision cross-section and $|\Delta\mathbf{v}(\mathbf{q}_j, \mathbf{q}_k)|$ the average relative velocity between the particles. Because particles cannot collide with themselves, we define a true collision rate λ_{jk} which explicitly suppresses self-collision:

$$\lambda_{jk} := (1 - \delta_{jk})\lambda(\mathbf{q}_j, \mathbf{q}_k). \quad (4)$$

The collision rate λ_j of particle j with any other particle in the ensemble is given by

$$\lambda_j := \sum_{k=1}^N \lambda_{jk}. \quad (5)$$

The total rate of collisions is

$$\lambda := \sum_{j=1}^N \sum_{k=j+1}^N \lambda_{jk} = \frac{1}{2} \sum_{j=1}^N \lambda_j. \quad (6)$$

Supposing, for the moment, that we had a small enough number of physical particles N that we can store them all on the computer (i.e. they would all be representative, though only of themselves), we would then proceed as follows to simulate the particle collisions. Following the method of Gillespie (1975), we choose a random time increment

$$\Delta t = -\lambda^{-1} \log(1 - \xi), \quad (7)$$

where ξ is a uniform random number drawn from the interval $[0, 1)$. We then draw random indices j and k from the discrete distribution defined by the joint probability

$$P(j)P(k|j) = \frac{\lambda_{jk}}{2\lambda}, \quad (8)$$

where $P(j) = \frac{1}{2}\lambda_j/\lambda$ is the chance that particle j is involved in the collision event and $P(k|j) = \lambda_{jk}/\lambda_j$ is the chance that particle k is the collision partner of particle j given that particle j will suffer a collision. We then advance the system by the time increment Δt and let particles j and k collide. This collision will generally

lead to a modification of the properties of the particles j and k involved. It can also lead to the merging of particles (in the case of coagulation) or to the creation of additional particles (by means of fragmentation). Therefore, the collision rate λ has to be recomputed after a collision. It also needs to be recomputed if the properties of the particles change by means of other processes.

3.2. Representative particles

Simulating a full ensemble of $N \sim 10^{30}$ particles is computationally unfeasible. We therefore pick a representative subset of n representative particles with $n \ll N$ and simulate only them while still allowing them to interact with all the other physical particles. For convenience we abbreviate ‘representative particle’ as RP henceforth.

Also for notational convenience, we assume that all physical particles are indexed from 1 to N , but with the RPs arranged at the beginning of the ordering. In other words, the particles with indices 1 to n are the RPs, all others are not, as illustrated in Fig. 1.

Let us now define the total collision rate of the RPs, or more precisely: the total collision rate of all RPs (indices 1 to n) with all physical particles (indices 1 to N):

$$\bar{\lambda} := \sum_{j=1}^n \sum_{k=j+1}^N \lambda_{jk}. \quad (9)$$

We adapt Gillespie’s method for the simulation of RPs by substituting $\bar{\lambda}$ for λ in Eq. (7). However, $\bar{\lambda}$ is not a computable quantity because the properties of the $(N - n)$ non-RPs are not known. To obtain a computable estimate of $\bar{\lambda}$, we first need to clarify how RPs actually represent other particles.

The basic assumption underlying the RPMC method is that a RP k can be regarded as a primus inter pares (first among equals), chosen from a larger *swarm* \mathcal{S}_k of particles whose properties are similar to \mathbf{q}_k . Here, \mathcal{S}_k is the index set of the N_k physical particles in the swarm k belonging to RP k . In other words: the n RPs divide the total ensemble of N physical particles up into n

swarms, each containing N_k physical particles, such that

$$N = \sum_{j=k}^n N_k. \quad (10)$$

Given that, for each swarm k , we only have information about the RP k , the simplest possible assumption about the properties of the other particles in that swarm is to say that they are not only similar in some unspecified way, but in fact identical:

$$\mathbf{q}_i \stackrel{!}{=} \mathbf{q}_k \quad \forall i \in \mathcal{S}_k. \quad (11)$$

What still needs to be defined is how large each swarm is. This is best defined through a conserved quantity. The only practical quantity that is strictly conserved, irrespective of the complexity of the model, is mass. And so we divide the total mass M up into swarms of masses M_k such that

$$M = \sum_k M_k. \quad (12)$$

In other words: each swarm k has a mass M_k , which is conserved throughout the simulation, unless a swarm is deliberately split or merged. The mass of a swarm and the number of particles in the swarm are directly related via

$$M_k = N_k m_k, \quad (13)$$

where m_k is the mass of each of the particles in the swarm k . In the original RPMC method of [Zsom & Dullemond \(2008\)](#) all swarms by design had the same mass, so that

$$M_k = \frac{M}{n}, \quad (14)$$

where M is the total mass in the system and n is the number of RPs, and thus also the number of swarms. We later show that this equal-mass partitioning of swarms is not necessary for the method to work.

3.3. Simulation of representative particles

To approximate the total collision rate of RPs in Eq. (9), we first split it in two terms,

$$\bar{\lambda} = \sum_{j=1}^n \left(\sum_{k=j+1}^n + \sum_{k=n+1}^N \right) \lambda_{jk}, \quad (15)$$

where the first term describes the collisions among RPs and the second term describes collisions between RPs and non-representative swarm particles.

By invoking the assumption that all particles in a swarm have identical properties, as stated by Eq. (11), we can replace the sum of RP–non-RP collision rates with the sum of RP–RP collision rates multiplied with the number of non-RP particles in the swarm,

$$\sum_{k=n+1}^N \lambda_{jk} = \sum_{i=1}^n (N_i - 1) \lambda(\mathbf{q}_j, \mathbf{q}_i) \quad \text{for } j \in \{1, \dots, n\}. \quad (16)$$

We note that the raw collision rate $\lambda(\mathbf{q}_j, \mathbf{q}_i)$ had to be used here instead of the true collision rate λ_{ji} . The reason is that, while a RP j cannot collide with itself, it may collide with another

particle from its own swarm. In the true collision rate, λ_{jj} had been set to 0 to avoid self-collisions, so its use in Eq. (16) would unphysically avoid collisions between particles from the same swarm. Hence the use of the raw collision rate instead.

Rewriting the first term of Eq. (15) as

$$\sum_{j=1}^n \sum_{k=j+1}^n \lambda_{jk} = \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \lambda_{jk}, \quad (17)$$

and using Eqs. (4), (16), we find

$$\bar{\lambda} = \sum_{j=1}^n \sum_{k=1}^n \left(N_k - \frac{1 + \delta_{jk}}{2} \right) \lambda(\mathbf{q}_j, \mathbf{q}_k). \quad (18)$$

From this we can define the RP–swarm collision rate $\bar{\lambda}_{jk}$,

$$\bar{\lambda}_{jk} = \left(N_k - \frac{1 + \delta_{jk}}{2} \right) \lambda(\mathbf{q}_j, \mathbf{q}_k), \quad (19)$$

and the cumulative collision rate of RPs $\bar{\lambda}_j$,

$$\bar{\lambda}_j = \sum_{k=1}^n \bar{\lambda}_{jk}. \quad (20)$$

Although λ_{jk} and $\lambda(\mathbf{q}_j, \mathbf{q}_k)$ are commutative, $\bar{\lambda}_{jk}$ is not.

The term $-(1 + \delta_{jk})/2$ in the above expressions accounts for the possibility that RP j collides with a particle from swarm k that is itself a RP. The probability that this happens is

$$\bar{P}_{jk}^{\text{RP}} = \frac{1 - \delta_{jk}}{N_k}. \quad (21)$$

For such (usually rare) events, we have to avoid double-counting, because this potential collision is also accounted for by RP k . So this collision has to be counted only half, meaning that we have to subtract 1/2 from the number of particles N_k in the swarm k , as done in Eq. (18). If $j = k$, that is, if we consider collisions of RP j with members of its own swarm, then instead of subtracting just 1/2 we have to subtract 1, because apart from the RP itself, there are only $(N_j - 1)$ particles in the swarm. Hence the term $-(1 + \delta_{jk})/2$ in Eq. (18).

3.4. The large-particle-number approximation of the original RPMC method

The original RPMC method now makes the additional assumption that the number of particles per swarm is much greater than unity,

$$N_k \gg 1 \quad \forall k \in \{1, \dots, n\}. \quad (22)$$

This allows neglecting the probability of RP–RP collisions ($\bar{P}_{jk}^{\text{RP}} \approx 0$). With this assumption, we can define the simplified RP–swarm collision rate as

$$\tilde{\lambda}_{jk} = N_k \lambda(\mathbf{q}_j, \mathbf{q}_k) \approx \bar{\lambda}_{jk} \quad (23)$$

and the simplified cumulative collision rates as

$$\tilde{\lambda}_j = \sum_{k=1}^n \tilde{\lambda}_{jk}, \quad \tilde{\lambda} = \sum_{j=1}^n \tilde{\lambda}_j. \quad (24)$$

Gillespie's method is then applied to simulate RP collisions by substituting λ with $\tilde{\lambda}$ in Eq. (7):

$$\Delta t = -\tilde{\lambda}^{-1} \log(1 - \xi). \quad (25)$$

A RP suffering a collision is then chosen by drawing an index j from the discrete distribution with probability

$$\tilde{P}(j) = \frac{\tilde{\lambda}_j}{\tilde{\lambda}}. \quad (26)$$

Then a collision partner from swarm k is chosen, where the index k is drawn from the discrete distribution with the probability

$$\tilde{P}(k|j) = \frac{\tilde{\lambda}_{jk}}{\tilde{\lambda}_j}. \quad (27)$$

3.5. Stochastic representation of collisional outcomes

With this method we can randomly determine the time of the next collision event, the RP j suffering a collision, and the swarm k which represents the collision partner. The collision can have a variety of results: coagulation, fragmentation, pulverisation, cratering, etc. (Güttler et al. 2010; Blum 2018; Wurm & Teiser 2021), and it can yield a wide variety of particle distributions ranging from a single merged particle to an almost-continuous spectrum of fragment masses. This raises the question of how the resulting particle distribution can be incorporated into the simulation model.

Of course, in case of fragmentation or pulverisation, adding every resulting particle to the simulation is infeasible. Instead, we use the sampling nature of the Monte Carlo process to let the resulting particle distribution realise itself stochastically. Let

$$n^{\text{coll}}(\mathbf{q}; \mathbf{q}_j, \mathbf{q}_k) =: n_{jk}^{\text{coll}}(\mathbf{q}) \quad (28)$$

denote the number density distribution of the resulting fragments from a collision of RP j with a particle from swarm k . It is normalised as follows:

$$\int m(\mathbf{q}) n_{jk}^{\text{coll}}(\mathbf{q}) d\mathbf{q} = m(\mathbf{q}_j) + m(\mathbf{q}_k), \quad (29)$$

where $m(\mathbf{q})$ is the mass of a particle with properties \mathbf{q} , and the integration is over all possible properties \mathbf{q} .

We now draw a single set of particle properties \mathbf{q}^{coll} from the weighted distribution

$$P_{jk}(\mathbf{q}) = \frac{m(\mathbf{q}) n_{jk}^{\text{coll}}(\mathbf{q})}{m(\mathbf{q}_j) + m(\mathbf{q}_k)}. \quad (30)$$

In other words, a collision in our simulation always consumes the representative particle and replaces it with a new representative particle sampled from the particle mass distribution of the collision outcome,

$$\mathbf{q}_j \rightarrow \mathbf{q}^{\text{coll}}. \quad (31)$$

The masses of both swarms do not change,

$$M_j \rightarrow M_j, \quad M_k \rightarrow M_k. \quad (32)$$

For a single collision, the method always chooses a single outgoing particle, so that the number of RPs remains constant. If the collision leads to merging, then the resulting particle will

have a mass $m(\mathbf{q}_j) + m(\mathbf{q}_k)$. If, instead, the collision leads to a distribution of fragments, then the resulting particle will be a single mass-weighted choice among those fragments. The fragment distribution will then emerge as large numbers of collisions lead to many samplings of similar fragment distributions.

As demonstrated in Zsom & Dullemond (2008), the RPMC method yields the expected result for the standard analytic coagulation tests, albeit with substantial noise owed to the Monte Carlo nature of the method. The method also has the desirable consequence that the number of RPs always remains constant, which allows for simpler and more efficient implementation of the method. Finally, the representativeness of the sampling remains optimal: every RP keeps representing the same total mass.

4. The improved RPMC method

As mentioned before, the RPMC method, as described in Sect. 3, has two major limitations: (1) all swarms have the same mass and (2) all swarms must contain a very large number of physical particles. Here we present an improved RPMC method that overcomes these limitations. We put these improvements to the test in later sections.

4.1. Unequal-mass swarms

The original RPMC method assumed that all swarms have equal mass $M_k = M/n$. The reason for this choice was that the representative particles were considered to be randomly chosen from the full set of particles, using mass as the weight factor. This fits well to the 'atomistic' picture mentioned above: we randomly pick n 'atoms', all having equal weight, and then consider, for each representative atom, the dust particle it is in to be the representative particle. This guarantees that the n representative particles together form an unbiased sampling of the full ensemble of particles.

However, as we saw in Sect. 3.2, the formalisation of the RPMC algorithm allows for unequal swarm masses $M_k \neq M/n$, as long as the sum of all these masses equals the total mass M . The way this enters into the algorithm is through the modified number of particles per swarm

$$N_k = \frac{M_k}{m_k} \quad (33)$$

which appears in the collision rate calculation in Eq. (18). If we define the 'weight' of a RP as

$$W_k = \frac{M_k}{M} \quad (34)$$

then in the original RPMC method all RPs had equal weight (i.e. they were all equally 'important'). But in the improved RPMC method the RPs are now no longer necessarily of equal weight.

We can split any swarm up into a number of smaller swarms. For example, we might replace a RP k with weight W_k with two RPs l, m with weights $W_l + W_m = W_k$. Or to put it in terms of their masses: we split the swarm of mass M_k up into two swarms of masses $M_l + M_m = M_k$. Because the particles in both swarms have identical mass, $m_l = m_m = m_k$, the total number of particles is not changed, $N_l + N_m = N_k$.

4.2. The limits to growth

A much more difficult problem is to enable the RPMC method to deal with swarms that contain only a few or even just a single particle.

The original RPMC method was constructed to simulate a growth process over many orders of magnitude with a constant number of RPs while keeping resolution optimal. The method crucially relies on the assumption stated in Eq. (22) that the number of particles in each swarm is much greater than unity, $N_k \gg 1 \forall k \in \{1, \dots, n\}$. Equivalently, the particle mass is assumed to be much smaller than the swarm mass, $m_k \ll M_k \forall k \in \{1, \dots, n\}$. This assumption allowed the use of the simplified collision rate $\tilde{\lambda}_{jk}$ defined in Eq. (23) instead of $\bar{\lambda}_{jk}$ in Eq. (18), and it allowed us to keep the number of RPs n constant because we could neglect RP–RP collisions.

When the particle mass m_k is no longer negligibly small compared to the swarm mass M_k , several problems appear with the original RPMC method. Firstly, the number of particles in a swarm is computed through $N_k = M_k/m_k$. Like all RP and swarm properties, N_k is an expectation value. Therefore, a fractional swarm particle count such as $N_k = 1.3$ is not per se problematic. But if we want to treat single-particle swarms as non-stochastic objects—for example, we might want to identify singular particles with N -body objects in a hybrid MC/ N -body simulation—we must decide for an integer number of particles. However, by rounding N_k to an integer, we no longer conserve the total mass of the system. Secondly, by neglecting the possibility of RP–RP collisions, we were able to adopt the simplified RP–swarm collision rate of Eq. (23). $\tilde{\lambda}_{jk} = N_k \lambda(\mathbf{q}_j, \mathbf{q}_k)$. But for $N_k \sim 1$, this collision rate significantly overestimates the actual RP–swarm collision rate given in Eq. (19). In the extreme case of $N_k = 1$, we have

$$\tilde{\lambda}_{jk} = \lambda(\mathbf{q}_j, \mathbf{q}_k); \quad \bar{\lambda}_{jk} = \frac{1 - \delta_{jk}}{2} \lambda(\mathbf{q}_j, \mathbf{q}_k). \quad (35)$$

(To solve this problem, one might be tempted to simply use the actual RP–swarm collision rate $\bar{\lambda}_{jk}$ instead of approximating it as $\tilde{\lambda}_{jk}$. However, it turns out that the approximation $\bar{\lambda}_{jk} \approx \tilde{\lambda}_{jk}$ must be made, otherwise the RPMC method is not statistically balanced, as demonstrated in Appendix C). Thirdly, a particle cannot grow in mass beyond the swarm mass: $m_k \leq M_k$, whereas in reality it might do so. The original RPMC method strictly keeps M_k constant in time, meaning that this creates an artificial upper bound to the mass a physical particle can acquire. When increasing the number of swarms n , we can decrease the number of particles N_k in swarms $k \in \{1, \dots, n\}$ to improve the resolution, but as a result, we also decrease the largest individual mass $m_k \leq M_k$ which the simulation can represent. Finally, if N_k is not large, the probability that RP j will collide with the RP of swarm $k \neq j$ is not negligibly small.

4.3. Extending the method

To overcome the problems outlined in the previous section, we propose some modifications to the RPMC method. As the basis of our extension, we first classify all swarms $k \in \{1, \dots, n\}$ in two categories: many-particles swarms with $N_k > N_{\text{th}}$, and few-particles swarms with $N_k \leq N_{\text{th}}$, where the simulation parameter N_{th} is the particle number threshold.

Every RP is associated with a swarm. When a RP j interacts with a particle from swarm k , we can use the above classification to define interaction regimes: the interaction operates in the many-particles regime if both swarms j, k are many-particles

swarms, and in the few-particles regime if at least one swarm of j, k is a few-particles swarm.

We then use the following adaption of the effective RP–swarm collision rate:

$$\tilde{\lambda}_{jk} = N_{jk} \lambda(\mathbf{q}_j, \mathbf{q}_k) \quad (36)$$

where we define the swarm multiplicity factor N_{jk} :

$$N_{jk} := \begin{cases} N_k & \text{in the many-particles regime} \\ N_k - \frac{1 + \delta_{jk}}{2} & \text{in the few-particles regime.} \end{cases} \quad (37)$$

Interactions in the many-particles regime are handled with the conventional RPMC method: we assume that a RP j always interacts with a non-RP from swarm k . A collision then alters only the properties of RP j .

If the interaction operates in the few-particles regime, we make the simplest possible choice and impose that, in the swarm with fewer particles, all particles operate in ‘lock-step’, that is, they all do as the RP does, at the same time. Therefore, instead of RP–swarm interactions, we now consider swarm–swarm interactions. Imposing $N_j \leq N_k$ without loss of generality, the result of the collision is still determined following the procedure described in Sect. 3.5, $\mathbf{q}_j \rightarrow \mathbf{q}^{\text{coll}}$, but the swarm masses M_j and M_k now grow and shrink as mass is transferred to the swarm with fewer particles:

$$M_j \rightarrow M_j + N_j m_k, \quad M_k \rightarrow M_k - N_j m_k. \quad (38)$$

By assumption of $N_j \leq N_k$, the swarm mass M_k cannot become negative.

Interactions in the few-particles regime will therefore upset the sampling balance: as swarm weights change, different RPs represent different fractions of the total mass. As was argued in Sect. 4.1, this does not impair the validity of the simulation, but it may lead to inefficient use of the available RPs. The particle number threshold N_{th} should therefore be chosen as small as possible so that the sampling is not unnecessarily disequibrated. At the same time, N_{th} should be large enough such that, for $N_k \geq N_{\text{th}}$, the simplified RP–swarm collision rate Eq. (23), $\tilde{\lambda}_{jk} = N_k \lambda(\mathbf{q}_j, \mathbf{q}_k)$, only negligibly overestimates the RP–swarm collision rate $\bar{\lambda}_{jk}$. In our tests we found satisfactory results with $N_{\text{th}} \sim 10$.

This simple prescription allows unconstrained growth in the RPMC method while conserving mass and keeping the RP count unchanged. However, it has some implications that are detrimental to our goal of modelling runaway growth processes with high accuracy. First, unless N_j happens to be a power of 2, swarm self-interaction will still result in a fractional particle number. More gravely, the method usually does not resolve the runaway body individually. Let us consider a swarm j representing the heaviest bodies. By sweeping up smaller particles, the swarm has just become a few-particles swarm, $N_j = N_{\text{th}}$. Let us assume $N_{\text{th}} = 10$. The mass of RP j will grow further by coagulation of smaller bodies; but because of the mass transfer precept given in Eq. (38), the swarm body count will stay the same, $N_j = 10$. In other words, we have 10 bodies simultaneously experiencing runaway growth. This is at odds with the mass separation characteristic of runaway growth, sometimes paraphrased as ‘the winner takes it all’. Runaway growth happens when the accretion rate has a superlinear proportionality with mass. Then, the body which first enters the runaway growth regime has the largest as well as the fastest-growing accretion rate; it will separate from the particle mass distribution and ‘starve out’ its competitors.

Even worse, if bodies in the runaway regime cannot accrete each other, and their swarms will therefore not self-interact, then the swarm particle count cannot drop below N_{th} at all, and runaway bodies will never be resolved individually.

We therefore relinquish a core property of the RPMC method: we allow the number of RPs to grow. We impose that a swarm j that reaches the particle number threshold, $N_j \succ N_{\text{th}}$, is split into N_{th} single-particle swarms. The RP of a single-particle swarm i with $N_i = 1$ represents only itself.

To accurately model a runaway process, the initial number of RPs n^0 and the particle number threshold N_{th} must then be chosen large enough that swarms whose RPs surpass the critical mass are already resolved individually:

$$N_{\text{th}} n^0 > \frac{M}{m_{\text{crit}}}. \quad (39)$$

There is still the question of how to handle fractional particle counts. A many-particles swarm k becomes a few-particles swarm as the swarm particle count falls to or below the particle number threshold, $N_k \leq N_{\text{th}}$. This can happen in two different ways: (1) RP k may grow by accumulating mass through a collision in the many-particles regime where the swarm mass M_k is conserved, thereby decreasing the swarm particle count $N_k = M_k/m_k$; or (2) a few-particles swarm may subtract mass from swarm k through a few-particles interaction as per Eq. (38). In both cases, N_k may end up non-integral. Some kind of compromise then has to be made when splitting up swarm k into individual RPs. Assuming non-integral N_k , some possibilities are: (i) slightly adjusting the mass m_k before the splitting such that N_k becomes an integral number; (ii) splitting up the swarm to $\lceil N_k \rceil$ particles, one of which is assigned a slightly lower mass $(N_k - \lfloor N_k \rfloor)m_k$. In our implementation we chose the latter option. We argue that, if N_{th} is chosen large enough, this adjustment is of negligible significance. Furthermore, the boosting technique described in the next section will actually make the swarm particle count N_k an integral number in most situations where a regime transition would occur.

Fragmentation can be another cause of fractional particle counts. Consider a few-particles swarm j , $N_j \leq N_{\text{th}}$ whose representative particle collides with a particle from some other swarm k , resulting in a distribution of fragment masses. As per Sect. 3.5, a new mass m'_j is sampled from the number density distribution of the resulting fragments, while swarm j absorbs the mass of the impacting particles from swarm k as per Eq. (38). The new particle number can then become non-integral, for example if a cratering event leaves the new particle mass m'_j only slightly smaller than the combined mass, $m'_j = (1 - \epsilon)(m_j + m_k)$ with $\epsilon \ll 1$:

$$N'_j = \frac{M'_j}{m'_j} = \frac{N_j}{1 - \epsilon} \approx N_j(1 + \epsilon). \quad (40)$$

This incongruity points at a more severe problem: statistical treatment is not appropriate for fragmentation of individual bodies. As an example, consider the collision of proto-Earth and a protoplanet called Theia, an event which is hypothesised to have formed the Moon by cratering. We assume that proto-Earth is represented by RP j with $N_j = 1$. With the collision method discussed above, the mass of proto-Earth and Theia would be combined into a new unified swarm, and the new mass of its representative body would be chosen from the distribution of the fragments as either cratered proto-Earth (most likely), the Moon (with a chance of 1 : 82), or debris (unlikely). The

collision would thus likely result in a swarm of 1.0123 Earth-mass planet(s) or, on rare occasions, 82 Moons. This is clearly meaningless and wrong.

To ameliorate this we suggest, for non-coagulating collisions involving self-representing RPs, to decree that every fragment whose mass exceeds a certain threshold m_{th} be represented individually, and therefore to add new RPs to the simulation as required. From the sub- m_{th} end of the fragment distribution one can then sample one or more statistically representative fragments as RPs. With this approach, the collision of proto-Earth and Theia would result in two self-representing RPs of Earth and Moon mass and one or more RPs representing a swarm of debris.

4.4. Boosting

Despite tracking only a small subset of the particles, the simulation still models every single interaction of the representative particles. For wide distributions of particle masses, this is a severe constraint on the efficiency of the simulation. For example, assume that a boulder with a mass of 1000 kg accretes pebbles of mass 1 g. To grow to twice its initial mass, the boulder must accrete 10^6 pebbles, and each interaction is modelled as a separate RPMC event, which slows down the simulation tremendously. This problem was noted by Zsom & Dullemond (2008, Sect. 2.3), who proposed to group accumulation events of small particles together, effectively decreasing the collision rate while proportionally increasing the gain of mass during an individual event. We now introduce a formalised version of this boosting procedure.

4.4.1. Grouping interactions

To overcome the inefficiency inherent in a method that tracks every individual collision event, we want to group together similar events. To this end, we introduce a boost factor β_{jk} for interactions between RP j and swarm k . If the collision can be assumed to lead to ‘hit-and-stick’ coagulation, the collision rate $\tilde{\lambda}_{jk}$ from Eq. (36) is substituted with the boosted collision rate

$$\tilde{\lambda}_{jk}^{\text{b}} = N_{jk}^{\text{b}} \lambda(\mathbf{q}_j, \mathbf{q}_k), \quad (41)$$

where we defined the boosted swarm multiplicity factor

$$N_{jk}^{\text{b}} := \beta_{jk}^{-1} N_{jk}, \quad (42)$$

and the mass transferred to the RP during a collision then is

$$m_j \rightarrow m'_j = m_j + \beta_{jk} m_k. \quad (43)$$

For interactions in the few-particles regime, we again impose $N_j \leq N_k$ without loss of generality. During a boosted coagulation, mass is then transferred between swarms j and k as per

$$\begin{aligned} M_j &\rightarrow M'_j = M_j + \beta_{jk} N_j m_k, \\ M_k &\rightarrow M'_k = M_k - \beta_{jk} N_j m_k. \end{aligned} \quad (44)$$

It is clear that

$$\beta_{jk} \geq 1 \quad (45)$$

must hold, where $\beta_{jk} = 1$ neutralises the boosting effect. In the following we discuss further constraints for β_{jk} , and how a suitable boost factor can be determined while obeying these constraints.

4.4.2. Constraining the boost factor

Zsom & Dullemond (2008) propose to set a mass accumulation threshold $q_m := \delta m/m$ and choose the boost factor such that each collision event grows the mass of the larger particle by at least q_m . For example, if we wanted to accept a relative mass gain of 5% during a single collision event, we would set $q_m = 0.05$ and choose an initial boost factor of

$$\beta_{jk}^0 := q_m \frac{m_j}{m_k}. \quad (46)$$

The boost factor β_{jk} is subject to additional constraints. If the collision operates in the many-particles regime, growth by coagulation will decrease the number of particles in swarm j as the mass of its RP grows,

$$N'_j = \frac{M'_j}{m'_j} = \frac{M_j}{m_j + \beta_{jk} m_k}. \quad (47)$$

To avoid an abrupt ‘regime change’, we would like to avoid boosting collisions such that the swarm particle count falls below the particle number threshold, and hence we require $N'_j \geq N_{\text{th}}$, which implies the constraint

$$\beta_{jk} \leq \frac{m_j}{m_k} \left(\frac{N_j}{N_{\text{th}}} - 1 \right). \quad (48)$$

For the few-particles regime, we had imposed $N_j \leq N_k$ without loss of generality. This allowed us to pretend that all particles in a swarm j grow by accumulating one particle from swarm k each. When introducing a boost factor, we must ensure that the new swarm mass M'_k remains a non-negative value; hence it follows from Eq. (44) that the boost factor must satisfy

$$\beta_{jk} \leq \frac{N_k}{N_j}. \quad (49)$$

An interaction in the few-particles regime may change the mass of swarm k . If swarm k is a many-particles swarm, the loss of mass might push its swarm particle count $N'_k = N_k - \beta_{jk} N_j$ below N_{th} , which we again want to avoid:

$$\beta_{jk} \leq \frac{N_k - N_{\text{th}}}{N_j}. \quad (50)$$

However, we want to allow for the depletion of a many-particles swarm, and thus apply this constraint only if the swarm would not be depleted because $\beta_{jk} < N_k/N_j$.

To summarise, closed-form expressions for the boost factor can be given as

$$\beta_{jk} = \max \left\{ 1, \min \left\{ \beta_{jk}^0, \frac{m_j}{m_k} \left(\frac{N_j}{N_{\text{th}}} - 1 \right) \right\} \right\} \quad (51)$$

for interactions in the many-particles regime, and as

$$\beta_{jk} = \begin{cases} \max \left\{ 1, \min \left\{ \frac{N_k - N_{\text{th}}}{N_j}, \beta_{jk}^0 \right\} \right\} & \text{if } N_k > N_{\text{th}} \\ & \text{and } \beta_{jk}^0 < \frac{N_k}{N_j} \\ \frac{N_k}{N_j} & \text{if } N_k > N_{\text{th}} \\ & \text{and } \beta_{jk}^0 \geq \frac{N_k}{N_j} \\ 1 & \text{if } N_k \leq N_{\text{th}} \end{cases} \quad (52)$$

for interactions in the few-particles regime.

4.4.3. Boosting and fragmentation

The boost factor effectively groups particles together before having them accreted. Boosting is therefore not directly applicable if collisions lead to other outcomes, such as pulverisation or cratering. Depending on the sophistication of the fragmentation model used, a given collision may lead to either coagulation or fragmentation with a certain probability, or only a fraction of the combined mass is shattered to smaller pieces. For an overview of the findings of laboratory studies of dust coagulation and fragmentation, see Güttler et al. (2010), and Blum (2018).

A simple way of keeping the benefits of boosting while allowing for fragmentation is to exploit the probabilistic nature of the simulation. When simulating a collision, the new mass of the RP is chosen randomly by sampling the fragment mass distribution as explained in Sect. 3.5. If a fraction $(1 - f_{\text{frag}})$ of the total mass remains in one cohesive body, then the probability that this massive body is chosen as the RP is $(1 - f_{\text{frag}})$. Equivalently, one could say that the collision rate $\tilde{\lambda}_{jk}$ is decomposed in two contributions for fragmentation and for coagulation,

$$\tilde{\lambda}_{jk} = f_{\text{frag}} \tilde{\lambda}_{jk} + (1 - f_{\text{frag}}) \tilde{\lambda}_{jk}. \quad (53)$$

A boost factor is then applied only to the partial collision rate corresponding to coagulation:

$$\tilde{\lambda}_{jk}^b = f_{\text{frag}} \tilde{\lambda}_{jk} + \beta_{jk}^{-1} (1 - f_{\text{frag}}) \tilde{\lambda}_{jk}. \quad (54)$$

Boosting decreases the effective collision rate, hence the likelihood f_{frag} that the RP is sampled from the mass distribution of fragments must increase:

$$f_{\text{frag}} \rightarrow \frac{f_{\text{frag}}}{f_{\text{frag}} + \beta_{jk}^{-1} (1 - f_{\text{frag}})} =: f_{\text{frag}}^b. \quad (55)$$

Bukhari Syed et al. (2017) show that the transition from bouncing to fragmentation is smooth: the mass of the largest remaining fragment

$$m'_{\text{max}} = (1 - f_{\text{frag}})(m_j + m_k), \quad (56)$$

which is equal to the mass of the heavier particle in the case of bouncing, $m'_{\text{max}} = \max\{m_j, m_k\}$, continuously decreases as the kinetic impact energy increases. If the swarm particle k is much lighter than RP j , $m_k \ll m_j$, a large number of cratering impacts are required to effectuate a significant change in the mass of the largest fragment. Similar to coagulation events, we can also group cratering events together by generalising the initial boost factor (Eq. (46)) and boosted mass transfer (Eq. (43)) as

$$\beta_{jk}^0 := q_m \frac{m_j}{|\delta m|}, \quad (57)$$

$$m_j \rightarrow m'_j = m_j + \beta_{jk} \delta m, \quad (58)$$

where the transferred mass is given as

$$\delta m := (1 - f_{\text{frag}})(m_j + m_k) - m_j. \quad (59)$$

5. Validating the statistical balance

The RPMC method is asymmetric at its core: when operating in the many-particles regime, it is assumed that a RP always interacts with a non-RP, and hence only the properties of the RP are changed. This asymmetry seems natural in the atomistic picture

illustrated in Sect. 2. However, in the RP–swarm picture, correctness is not obvious. Consider the grains-and-boulders example originally given in Zsom & Dullemond (2008, Sect. 3.1). We have an initial ensemble of particles with only two different masses: N_B^0 heavy bodies with mass m_B^0 , henceforth called *boulders*, and N_g^0 light particles with mass $m_g < m_B^0$, from now on referred to as *grains*. We assume that mass is the only relevant particle property. We further assume that collisions are always of the hit-and-stick type, and that they happen only between boulders and grains. Because boulders do not collide with each other, their number cannot decrease. Likewise, because grains do not suffer mutual collisions, they cannot grow. Therefore, the number of boulders must remain invariant as they grow by accreting grains. In the RP–swarm picture, when a grain-mass RP hits and accretes a boulder, its entire swarm is converted to boulders, causing a sudden increase in the number of boulders. Conversely, when a boulder-mass RP j accretes a grain, its mass grows, $m_j \rightarrow m_j + m_g$, but the mass of its swarm is not changed, $M_j \rightarrow M_j$. Therefore, the number of boulders in the swarm $N_j = M_j/m_j$ decreases, $M_j/m_j \rightarrow M_j/(m_j + m_g)$. This section will formally prove that, for the specific case of the grains-and-boulders example, the two effects indeed cancel statistically.

5.1. Grains and boulders

We first define the collision rate as

$$\lambda(m, m') = \begin{cases} \Lambda(m, m') & \min\{m, m'\} < m_B^0 \text{ and} \\ & \max\{m, m'\} \geq m_B^0 \\ 0 & \text{otherwise,} \end{cases} \quad (60)$$

where the mutual collision rate $\Lambda(m, m')$ is an arbitrary commutative function of the masses m, m' . Boulders can therefore collide with grains, but collisions among boulders and among grains are suppressed.

Let us now represent this system by a selection of $n = n_g + n_B$ RPs, n_g of which represent the grains, and n_B represent the boulders. For notational convenience, we group the RP indices in two index sets \mathcal{I}_g and \mathcal{I}_B at any given time t , where \mathcal{I}_g holds the indices of the grain-mass RPs, and the indices in \mathcal{I}_B refer to boulder-mass RPs:

$$\begin{aligned} \mathcal{I}_g &:= \{i \in \{1, \dots, n\} : m_i < m_B^0\}, \\ \mathcal{I}_B &:= \{j \in \{1, \dots, n\} : m_j \geq m_B^0\}. \end{aligned} \quad (61)$$

In the RPMC simulation, the total number of grains N_g and total number of boulders N_B are given by

$$N_g = \sum_{i \in \mathcal{I}_g} N_i, \quad N_B = \sum_{j \in \mathcal{I}_B} N_j. \quad (62)$$

With the mutual collision rate between a boulder j and a grain conveniently abbreviated as

$$\Lambda_j := \Lambda(m_g, m_j), \quad (63)$$

we find the RP–swarm collision rates of Eq. (23) to be

$$\begin{aligned} \tilde{\lambda}_{ij} &= N_j \Lambda_j =: \tilde{\lambda}_{gj}, \\ \tilde{\lambda}_{ji} &= N_i \Lambda_j \end{aligned} \quad (64)$$

and the cumulative collision rates of Eq. (24) to be

$$\begin{aligned} \tilde{\lambda}_i &= \sum_{j \in \mathcal{I}_B} N_j \Lambda_j \equiv N_B \langle \Lambda \rangle_B =: \tilde{\lambda}_g, \\ \tilde{\lambda}_j &= N_g \Lambda_j =: \tilde{\lambda}_{B,j}, \end{aligned} \quad (65)$$

where $i \in \mathcal{I}_g$ and $j \in \mathcal{I}_B$, and where we defined the RP–swarm collision rate $\tilde{\lambda}_{gj}$ for any grain-mass RP with boulder swarm j , the cumulative grain-mass RP collision rate $\tilde{\lambda}_g$, and the cumulative RP collision rate $\tilde{\lambda}_{B,j}$ for boulder-mass RP j . The boulder average $\langle X \rangle_B$ of a particle-specific quantity X_j was defined as

$$\langle X \rangle_B \equiv \frac{1}{N_B} \sum_{j \in \mathcal{I}_B} N_j X_j. \quad (66)$$

A grain-mass RP can collide with a boulder, but it thereby turns into a boulder-mass RP which cannot collide with boulders anymore by choice of the collision rate function in Eq. (60). For a collision rate $\tilde{\lambda}_g$, the probability that a given grain-mass RP has suffered a collision with a boulder during time Δt is given by the exponential distribution

$$P(\tilde{\lambda}_g \Delta t) = 1 - e^{-\tilde{\lambda}_g \Delta t}. \quad (67)$$

The expected number of grains at time $t + \Delta t$ is therefore

$$\begin{aligned} E[N'_g] &= \sum_{i \in \mathcal{I}_g} e^{-\tilde{\lambda}_g \Delta t} N_i \\ &= N_g - \tilde{\lambda}_g N_B \Delta t + O(\Delta t^2). \end{aligned} \quad (68)$$

In the limit of $\Delta t \rightarrow 0$, this yields

$$\frac{dN_g}{dt} = \lim_{\Delta t \rightarrow 0} \frac{E[N'_g] - N_g}{\Delta t} = -\tilde{\lambda}_g N_B, \quad (69)$$

which is equivalent to Eq. (B.5) of the analytical model.

The RPMC method keeps swarm masses constant, $M'_j = M_j$. In the swarms that already had boulder mass at time t , the number of boulders must therefore decrease over a duration Δt :

$$\begin{aligned} E[N'_{B \rightarrow B}] &= \sum_{j \in \mathcal{I}_B} E\left[\frac{M'_j}{m'_j}\right] \\ &= \sum_{j \in \mathcal{I}_B} M_j E[(m'_j)^{-1}]. \end{aligned} \quad (70)$$

Growth of boulders is a Poisson point process, so for a boulder-mass RP $j \in \mathcal{I}_B$ with mass m_j at time t we have

$$\begin{aligned} E[(m'_j)^{-1}] &= \sum_{k=0}^{\infty} \frac{\text{Pois}_k(\tilde{\lambda}_{B,j} \Delta t)}{m_j + k m_g} \\ &= \frac{1}{m_j} \left(1 - \tilde{\lambda}_{B,j} \Delta t \frac{m_g}{m_j + m_g}\right) + O(\Delta t^2), \end{aligned} \quad (71)$$

and with $M_j = N_j m_j$, Eq. (70) evaluates to

$$\begin{aligned} E[N'_{B \rightarrow B}] &= \sum_{j \in \mathcal{I}_B} N_j \left(1 - \tilde{\lambda}_{B,j} \Delta t \frac{m_g}{m_j + m_g} + O(\Delta t^2)\right) \\ &= N_B - N_g m_g \Delta t \sum_{j \in \mathcal{I}_B} \frac{N_j \Lambda_j}{m_j + m_g} + O(\Delta t^2). \end{aligned} \quad (72)$$

At the same time, some of the grain-mass swarms become boulder-mass swarms after their RP hits a boulder and becomes part of it. The chance for a grain-mass RP $i \in \mathcal{I}_g$ to collide with a boulder is again given by the exponential distribution given in Eq. (67),

$$P(i) \equiv P(\tilde{\lambda}_g \Delta t) = \tilde{\lambda}_g \Delta t + \mathcal{O}(\Delta t^2). \quad (73)$$

Given that RP i undergoes a collision with a boulder, we now need to determine which boulder will be its collision partner. Because the collision rate is independent of the boulder mass, every boulder has the same chance of colliding with the given grain-mass RP i . The chance $P(j|i)$ that RP i collides with a boulder from swarm $j \in \mathcal{I}_B$ is therefore given by the relative collision rate of a given grain with swarm j ,

$$P(j|i) \equiv \frac{\tilde{\lambda}_{gj}}{\tilde{\lambda}_g} = \frac{N_j \Lambda_j}{N_B \langle \Lambda \rangle_B}. \quad (74)$$

If RP i collides with a boulder from swarm j , its entire swarm is converted to a swarm of boulders. The swarm mass remains the same, $M'_i = M_i$, while the grain RP sticks to the boulder and thereby grows to the mass $m_g + m'_j$ by time $t + \Delta t$. The number of boulders in the new swarm will thus be

$$\begin{aligned} E[N'_i] &= E\left[\frac{M'_i}{m_g + m'_j}\right] = M_i E[(m_g + m'_j)^{-1}] \\ &= \frac{M_i}{m_g + m_j} + \mathcal{O}(\Delta t). \end{aligned} \quad (75)$$

The total number of boulders in the swarms which were composed of grains at time t but have been converted to boulders by time $t + \Delta t$ is therefore

$$\begin{aligned} E[N'_{g \rightarrow B}] &= \sum_{i \in \mathcal{I}_g} P(i) \sum_{j \in \mathcal{I}_B} P(j|i) E[N'_i] \\ &= \Delta t \sum_{i \in \mathcal{I}_g} \sum_{j \in \mathcal{I}_B} N_j \frac{M_i \Lambda_j}{m_j + m_g} + \mathcal{O}(\Delta t^2). \end{aligned} \quad (76)$$

Noting that

$$\sum_{i \in \mathcal{I}_g} M_i = N_g m_g, \quad (77)$$

we find

$$E[N'_{g \rightarrow B}] = N_g m_g \Delta t \sum_{j \in \mathcal{I}_B} \frac{N_j \Lambda_j}{m_j + m_g} + \mathcal{O}(\Delta t^2). \quad (78)$$

We then obtain the expected total number of boulders at time $t + \Delta t$ by summing up the two contributions in Eqs. (72) and (78). The contributions cancel exactly to first order in Δt :

$$\begin{aligned} E[N'_B] &= E[N'_{B \rightarrow B}] + E[N'_{g \rightarrow B}] \\ &\stackrel{!}{=} N_B + \mathcal{O}(\Delta t^2). \end{aligned} \quad (79)$$

The RPMC method can thus be expected to keep the number of boulders N_B invariant,

$$\frac{dN_B}{dt} = \lim_{\Delta t \rightarrow 0} \frac{E[N'_B] - N_B}{\Delta t} \stackrel{!}{=} 0. \quad (80)$$

This result demonstrates in the RP–swarm picture that the two paradoxical properties exactly cancel, and that the RPMC method is statistically correct. It is worth pointing out that not only the totality of dN_B/dt vanishes, but for every individual $j \in \mathcal{I}_B$, the corresponding summands in Eqs. (72) and (78) cancel out. In other words, every swarm of boulders maintains its own statistical balance with the totality of grains.

5.2. Mutating swarm masses

As pointed out in Sect. 4.1, it is reasonable to have each RP i represent approximately the same relative mass $M_i \approx M/n$, where M is the total mass in the system. But our formal definition of the RPMC method does not impose any assumptions on the values of swarm masses M_i . This is crucial because the extension of the method presented in Sect. 4.3 transfers mass between swarms. Specifically, in an interaction that operates in the few-particles regime, a few-particles swarm – usually a single-particle swarm – may remove mass from a many-particles swarm. In the following we argue that this does not affect the statistical balance of the method.

It should first be noted that the statistical balance only concerns many-particles swarms, as only interactions between many-particles swarms operate in the asymmetric many-particles regime. To incorporate the effect of mass transfers, we admit that the mass M_i of any many-particles swarm i may be changed by an external source or sink modelled as

$$\frac{dM_i}{dt} = M_i f(m_i, t), \quad (81)$$

where the continuous function $f(m, t)$ is the expected relative inflow/outflow rate of particles of mass m at time t . We justify this approach by arguing that, if k is a many-particles swarm and j is a single-particle swarm, we have $N_j = 1$ and $N_k > N_{th}$, hence $N_j \ll N_k$, and therefore the change $N_k \rightarrow N_k - 1$ is small enough to be modelled as a continuous process.

With the source or sink model of Eq. (81), swarm mass M'_i at time $t + \Delta t$ can be expanded as

$$E[M'_i] = M_i + f(m_i, t) \Delta t. \quad (82)$$

This leads to a first-order deviation from N_g and N_B during time increment Δt ,

$$E[N'_g] = N_g + N_g (f_g - \tilde{\lambda}_g) \Delta t + \mathcal{O}(\Delta t^2), \quad (83)$$

$$E[N'_B] = N_B + N_B \langle f \rangle_B \Delta t + \mathcal{O}(\Delta t^2), \quad (84)$$

where the grain inflow rate f_g and the mean boulder inflow rate $\langle f \rangle_B$ are defined as

$$f_g \equiv f(m_g, t), \quad \langle f \rangle_B \equiv \frac{1}{N_B} \sum_{j \in \mathcal{I}_B} N_j f(m_j, t). \quad (85)$$

Any influence of inflow or outflow on the mass transfer terms in Eqs. (72) and (76) is of second order in Δt and hence does not upset the balance of Eq. (80). We also find

$$\frac{dN_g}{dt} = N_g (f_g - \tilde{\lambda}_g), \quad (86)$$

$$\frac{dN_B}{dt} = N_B \langle f \rangle_B, \quad (87)$$

which is equivalent to the result found by adding the source or sink term to the analytical model discussed in Appendix B.

6. Numerical tests

To assess the correctness and applicability of the improved RPMC method, we run a series of numerical tests which can be grouped in three parts.

In Sect. 6.1, we first simulate two well-known coagulation kernels for which analytical solutions are available, the constant and the linear kernel, leading to self-similar mass distributions. Equivalent tests were already conducted in Zsom & Dullemond (2008); we additionally verify that unequal samplings of the mass leave the result unimpaired as claimed in Sect. 4.1.

Second, in Sect. 6.2 we study the transition to the few-particles regime and the simulation of runaway growth. A full-scale non-representative Monte Carlo simulation of the constant kernel is compared to a simulation with the extended RPMC method. We also study runaway growth by simulating the product kernel, a test kernel with an analytical solution available which develops a very sharp runaway characteristic. Third, we test a runaway kernel that mimics gravitational focussing, finding no effect of the simulation parameters on the evolution of the mass distribution as long as the resolution criterion in Eq. (39) is met.

Third, in Sect. 6.3 we adopt two variants of the grains-and-boulders example given by Zsom & Dullemond (2008) for which analytical predictions can be made. We study how well the RPMC method retains the statistical balance depending on the number of RPs used.

6.1. Self-similar solutions

The constant kernel

$$\lambda(m, m') = \Lambda = \text{const.} \quad (88)$$

and the linear kernel

$$\lambda(m, m') = \Lambda_0 (m + m') \quad (89)$$

are well-known special cases of the Smoluchowski equation Smoluchowski (1916)

$$\begin{aligned} \frac{d}{dt} f(m, t) = & -f(m, t) \int_0^\infty dm' \lambda(m, m') f(m', t) \\ & + \frac{1}{2} \int_0^\infty dm' \lambda(m', m - m') f(m', t) f(m - m', t), \end{aligned} \quad (90)$$

an integrodifferential equation for the number density distribution $f(m, t)$ which describes coagulation processes in a continuous mass distribution. The first term represents the loss of particles at mass m due to coagulation with other particles, while the second term describes the gain of particles by coagulation of masses $m' + (m - m') = m$.

The results for a standard equal-mass sampling, $M_k = M/n\forall k$, are shown in Fig. 2, finding very good agreement with the analytical solutions, which are summarised in Appendix A. The same tests were re-run with heterogeneous swarm mass samplings. For the constant kernel, swarm masses were drawn from the log-normal swarm number density distribution

$$f(M') \propto \exp\left[-\frac{(\log M' - \mu)^2}{2\sigma^2}\right], \quad (91)$$

where we identified $\sigma \equiv \mu/20$ and chose μ such that the mean of the distribution is $\exp(\mu + \sigma^2/2) = M/n$. The samples M_k were

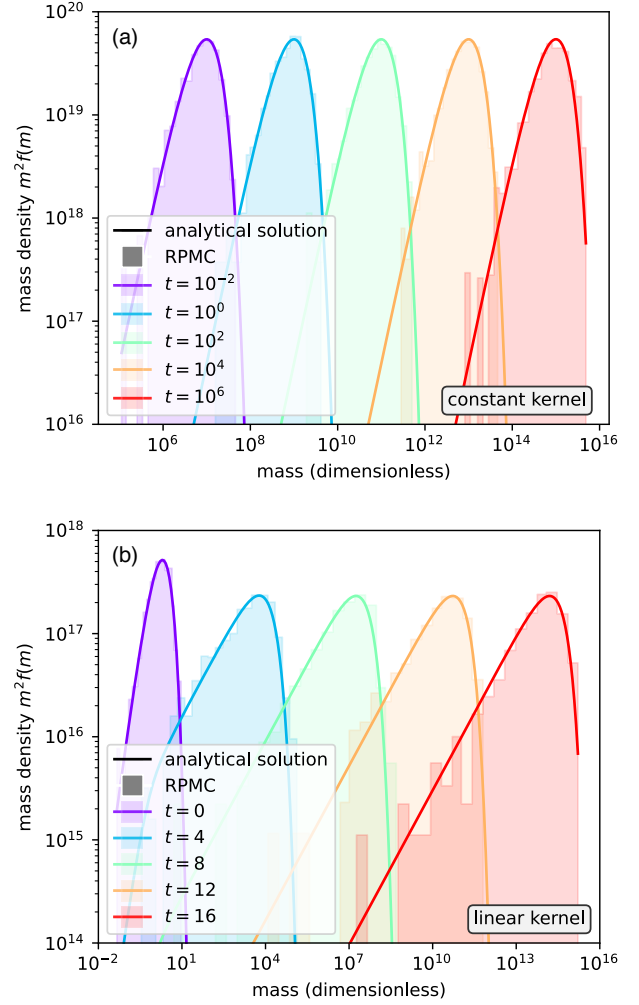


Fig. 2. Analytical solutions and RPMC simulations for standard coagulation tests: (a) constant kernel $\lambda(m, m') = \Lambda = \text{const}$; (b) linear kernel $\lambda(m, m') = \Lambda_0 (m + m')$. The RPMC simulation runs use $n = 1024$ RPs.

normalised such that $\sum_k M_k = M = 10^{20}$ held exactly. A log-space distribution was chosen because we wanted to assess the resiliency of the method not only to small variations in swarm mass; with the log-normal distribution given, swarm masses span several orders of magnitude. For the linear kernel test, we sampled the RP masses m_k from the initial particle number density distribution $f(m, 0)$ in Eq. (A.2) and then chose a constant swarm particle count $N_k \equiv \mathcal{N} \forall k$ such that $\sum_k m_k N_k = \mathcal{N} \sum_k m_k = M = 10^{18}$: instead of an equal-mass sampling, we chose an equal-count sampling of swarm masses. While this was done mainly for reasons of simplicity, the resulting swarm mass distribution also spans a few orders of magnitude. Both simulations yielded results not significantly different from those obtained with equal-mass samplings, demonstrating that the RPMC method works well despite considerable variations of swarm masses. The results are not shown separately because they are visually indiscernible from those in Fig. 2.

6.2. Regime transition and runaway growth

To quantify how well the regime transition works, we study three scenarios. First, we again run a simulation with the constant kernel in Eq. (88), but we let it run until all mass is concentrated in

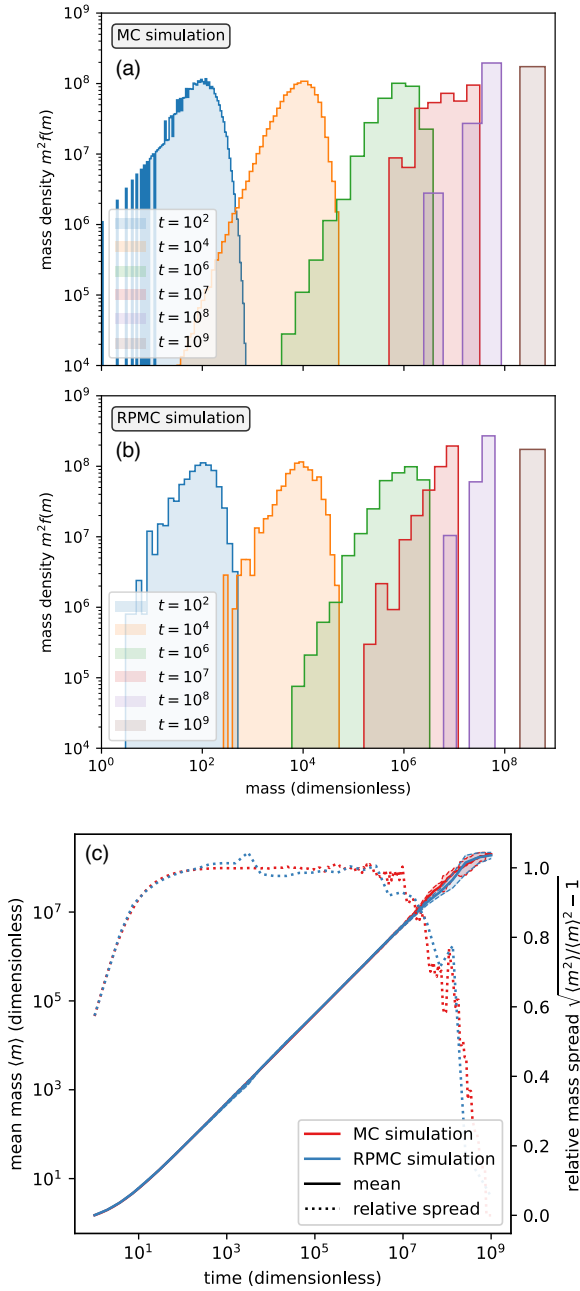


Fig. 3. Transition into the few-particles regime for the constant kernel Eq. (88): (a) Fiducial non-representative Monte Carlo simulation with $n = 10^8$ particles; (b) RPMC simulation with $n = 1\,024$ RPs; (c) mean mass $\langle m \rangle$ and relative mass spread $\sqrt{\langle m^2 \rangle / \langle m \rangle^2} - 1$ of both simulations in comparison. The discontinuities in (a) at $t = 10^2$ are an aliasing effect caused by histogram binning. (a) and (b) show snapshots of a single simulation, while (c) shows results averaged over 10 runs, with statistical error bounds indicated by the dashed curves.

a single particle. We note that the analytical solution in Eq. (A.1) is not applicable in this case because it is a solution to the continuous Smoluchowski equation (Eq. (90)), which approximates discrete coagulation processes only for particle number densities much greater than unity, $f(m, t) \gg 1$. Instead, we run a fiducial simulation of $n = 10^8$ RPs with swarm particle counts $N_k = 1 \forall k$ – that is, every RP only represents itself – and compare this to a regular RPMC simulation with $n = 1\,024$ RPs that transition into

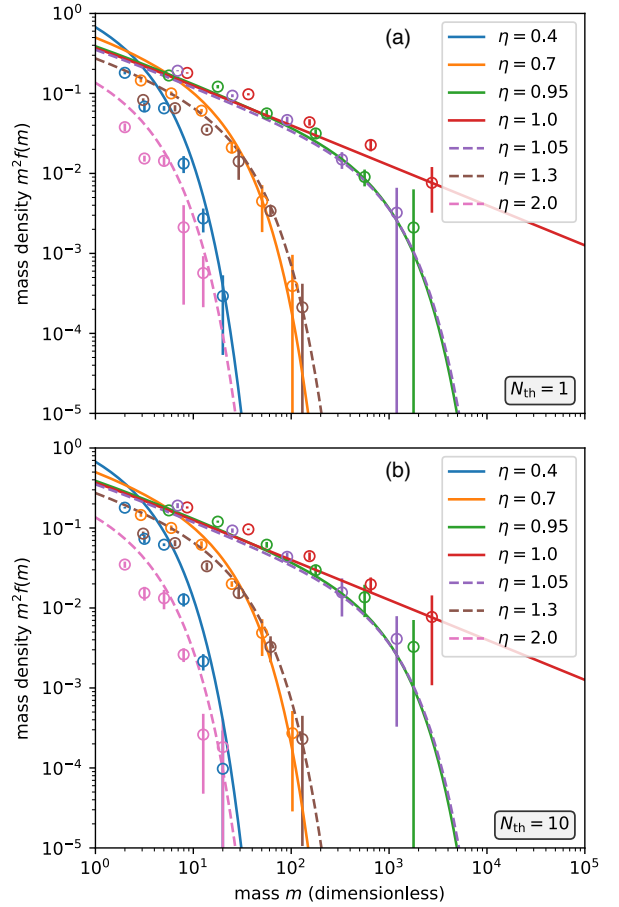


Fig. 4. Particle mass density at different times for the product kernel coagulation test, compared to the analytical solution in Eq. (A.9) (solid curves). The RPMC simulation uses $n = 2\,048$ particles and different values of the particle number threshold: (a) $N_{th} = 1$; (b) $N_{th} = 10$. The results shown were averaged over 5 runs, with a standard deviation indicated by the error bars.

the few-particles regime as they grow. As can be seen in Fig. 3, the two simulation runs are in good agreement.

Next, we study the product kernel

$$\lambda(m, m') = \Lambda_0 m m'. \quad (92)$$

The coagulation process produces a runaway particle at dimensionless time $\eta = 1$. The analytical solution for the product kernel is discussed in Appendix A.

In Fig. 4 we compare the particle number density of the analytical solution in Eq. (A.9) to an RPMC simulation with $n = 2\,048$ RPs, finding good agreement even in the runaway regime at $\eta > 1$. The figure can be directly compared to Fig. 5 in Ormel & Spaans (2008). Figure 5, which is modelled after Fig. 2 in Wetherill (1990), shows the analytical prediction for the number of particles, the mass of small bodies, and the mass of the runaway particle. The runaway particle mass produced by the RPMC simulation is also shown and in good agreement with the prediction.

In our third test of runaway growth, we use a collision kernel inspired by gravitational focussing

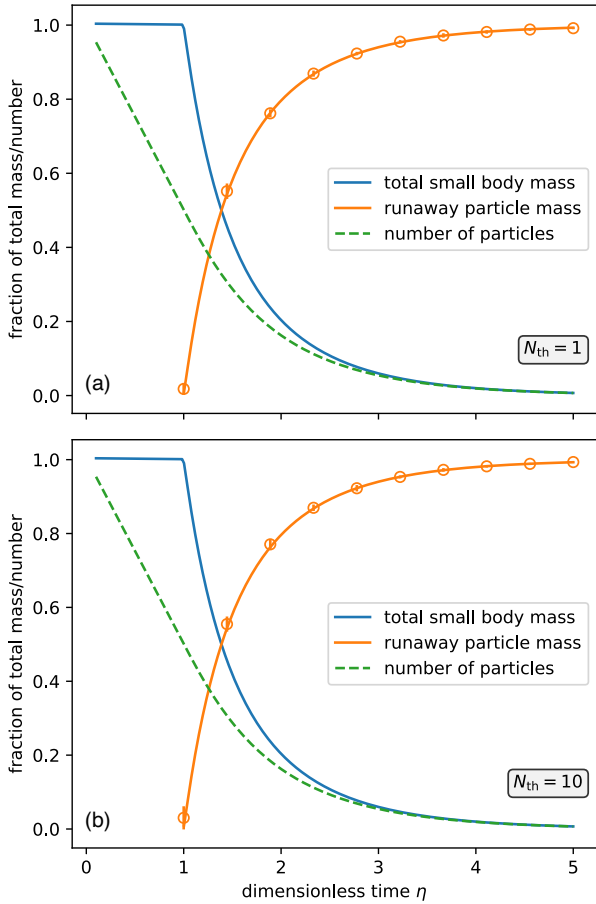


Fig. 5. Fraction of particle mass in the runaway particle for the product kernel coagulation test, compared to the analytical solution in Eq. (A.9) (solid curves). The RPMC simulation uses $n = 2048$ particles and different values of the particle number threshold: (a) $N_{\text{th}} = 1$; (b) $N_{\text{th}} = 10$. The results shown were averaged over 5 runs, with a standard deviation indicated by the error bars.

(cf. e.g. Armitage 2017, Sect. III.B.1):

$$\lambda(m, m') = \Lambda_0 \max\{m, m'\}^{2/3} \left[1 + \left(\frac{\max\{m, m'\}}{m_{\text{th}}} \right)^{2/3} \right] \quad (93)$$

with some constant coefficient Λ_0 and a threshold mass m_{th} . The collision rate therefore scales with $m^{2/3}$ for masses below m_{th} and with $m^{4/3}$ for masses above, which implies that particles above the threshold mass sweep up mass much more efficiently. As with any power-law collision rate, we expect the initial mass distribution to relax to a self-similar state while in the $\propto m^{2/3}$ regime. As the largest particles approach the threshold mass, we expect them to start ‘running away’, making the mass distribution bimodal. Finally, the most massive particle will end up accumulating all the remaining mass.

Results are given in Fig. 6, confirming our expectations: the initial distribution quickly becomes self-similar and stays that way until $t \sim 30$. After the higher-mass end of the distribution exceeds the threshold mass $m_{\text{th}} = 10^7$, a second peak starts to form, clearly visible at $t = 50$. At $t = 60$ the runaway particles are already depleting the tail end of the mass distribution, and almost all mass has concentrated in a single particle by $t = 80$. We find no notable sensitivity to the number of RPs as long as the resolution criterion Eq. (39) is satisfied.

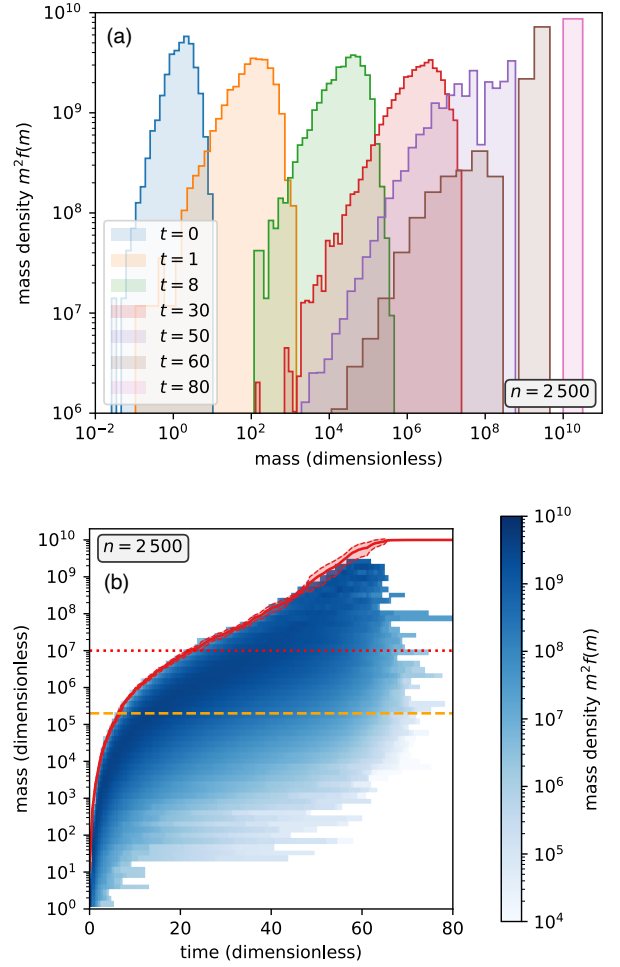


Fig. 6. RPMC simulation with the runaway collision rate in Eq. (93), a dimensionless threshold mass of $m_{\text{th}} = 10^7$ (indicated by dotted red horizontal line in (b)), a total mass of $M = 10^{10}$, a homogeneous swarm mass distribution, a particle number threshold of $N_{\text{th}} = 20$, and $n = 2500$ RPs. Every individual swarm i has mass $M_i = M/n$; particles are resolved individually once they grow beyond the mass $M/(nN_{\text{th}})$ (indicated by the dashed orange horizontal line in (b)). Each panel shows a time series of histograms. The mass-weighted particle number density is shown on the vertical axis of (a), whereas in (b) the mass-weighted histogram bin counts are colour-encoded on a logarithmic scale, and the mass of the runaway particle is shown separately (red curve). Histogram bin counts and mass of runaway particle are averaged over 10 runs, the error bounds of the runaway particle being indicated by red dashed curves.

6.3. Grains and boulders

We simulate the grains-and-boulders example introduced and analysed in Sect. 5 to validate the results and to confirm that statistical balance is indeed retained. We first consider a mass-independent mutual collision rate,

$$\Lambda(m, m') = \Lambda_0 \quad (94)$$

with some constant Λ_0 . An analytical prediction for the expected mean boulder mass $\langle m \rangle_{\text{B}}$ and its standard deviation $\sigma_{m_{\text{B}}}$ is developed in Appendix B. It is shown for different initial conditions in Fig. 7. The model was then simulated with the RPMC method for the same set of initial conditions; results are given in Fig. 8. We observe very good agreement with the analytical prediction.

Boulders remain boulders after a hit-and-stick collision with a grain. The number of boulders therefore cannot change,

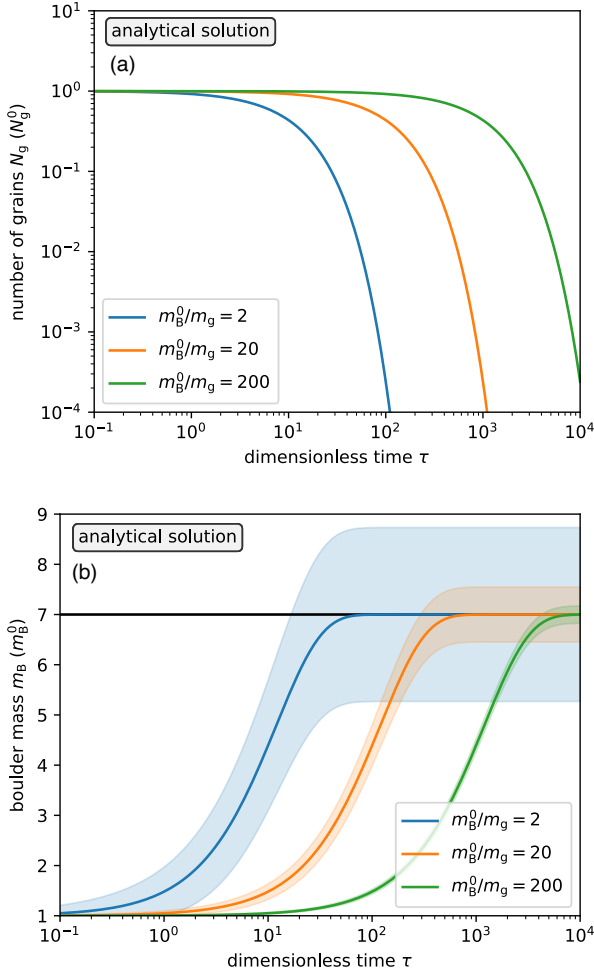


Fig. 7. Analytical solution given in Eqs. (B.11), (B.15) to the grains-and-boulders model introduced in Sect. 5 for different boulder-grain particle mass ratios: (a) total number of grains N_g ; (b) expected boulder mass $\langle m \rangle_B$. The total mass ratio of grains and boulders is $M_B^0/M_g^0 = 1/6$, so boulders are expected to grow to 7 times their initial mass, as indicated by the horizontal black line. The horizontal axes refer to dimensionless time $\tau = \Lambda N_g^0 t$. In (b), the filled area indicates the physical spread σ_{m_B} of the mass distribution given by Eq. (B.17).

$dN_B/dt = 0$, and thus

$$N_B = N_B^0. \quad (95)$$

When simulating an ensemble of grains and boulders with the RPMC method, the number of boulders is

$$N_B = \sum_{\substack{j=1 \\ m_j \geq m_B^0}}^n \frac{M_j}{m_j}. \quad (96)$$

If the simulation operates in the many-particles regime, the swarm masses M_j do not change. Therefore, as boulder-mass RPs grow, the number of boulders per swarm $N_j = M_j/m_j$ decreases. At the same time, some grain-mass RPs collide with a boulder and therefore become boulder-mass RPs. Both effects contribute to the sum in Eq. (96), and as argued before we expect them to cancel exactly.

Figure 9a visualises this statistical balance. As can be seen, the two contributions operate on slightly different timescales. It

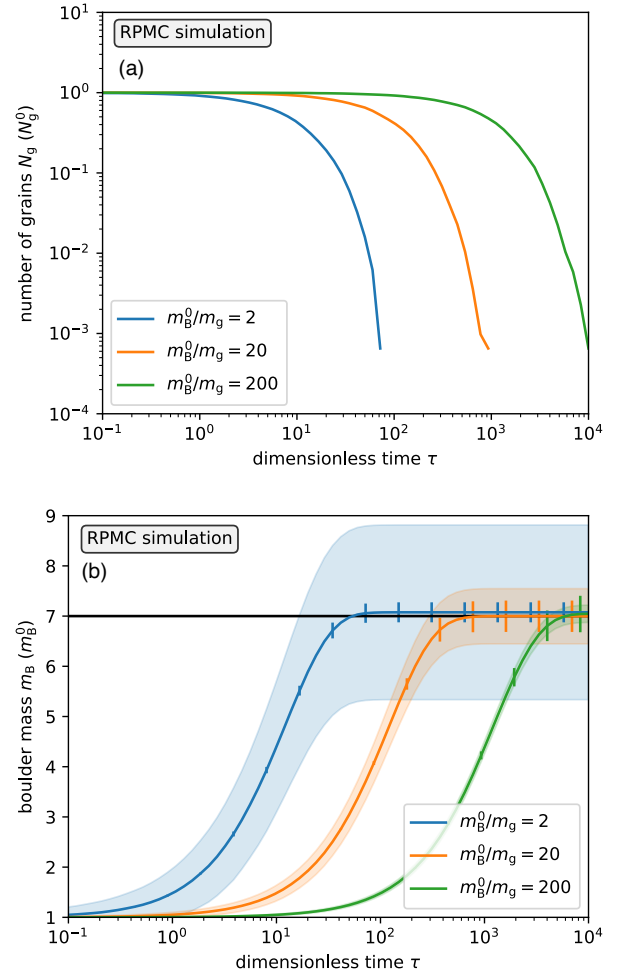


Fig. 8. RPMC simulation of the grains-and-boulders model introduced in Sect. 5 for different boulder-grain particle mass ratios for comparison with Fig. 7: (a) total number of grains N_g ; (b) average boulder mass $\langle m \rangle_B$. The simulation uses $n = 1792$ RPs divided into $n_B = 256$ boulder-mass RPs and $n_g = 6n_B = 1536$ grain-mass RPs with equal-weight swarms, $M_i = M_j \forall i, j \in \{1, \dots, n\}$, and an initial number of $N_g^0 = 10^8$ grains. The boulder masses in (b) are averaged over 10 consecutive runs; the filled area indicates the physical spread σ_{m_B} of the mass distribution, and the error bars indicate the standard deviation of the mean boulder mass over 10 runs. The horizontal black line indicates the expected final boulder mass $m_B^\infty = 7m_B^0$.

is more likely for a boulder to hit a grain than for a grain to hit a boulder; but when a boulder-mass RP accumulates a grain, it just grows a little and thus represents slightly fewer boulders than before, whereas a grain-mass RP that hits a boulder will instantly convert its entire swarm to boulders, causing a surge in total boulder count. We can observe the relative smoothness of the particle number decay as well as the more erratic particle number growth in Fig. 9a.

We naturally expect the resolution of the simulation to improve if we add more RPs. The influence of the number of RPs on the accuracy of the result is studied in Fig. 9b for different RP counts. The statistical precision of the final boulder mass is estimated as the standard deviation of the final masses in repeated runs of the simulation and shown as error bars around the mean value. We find it to be insensitive to the particle mass ratio m_B^0/m_g . However, the physical spread of the mass distribution, whose standard deviation is visualised as a filled area, does

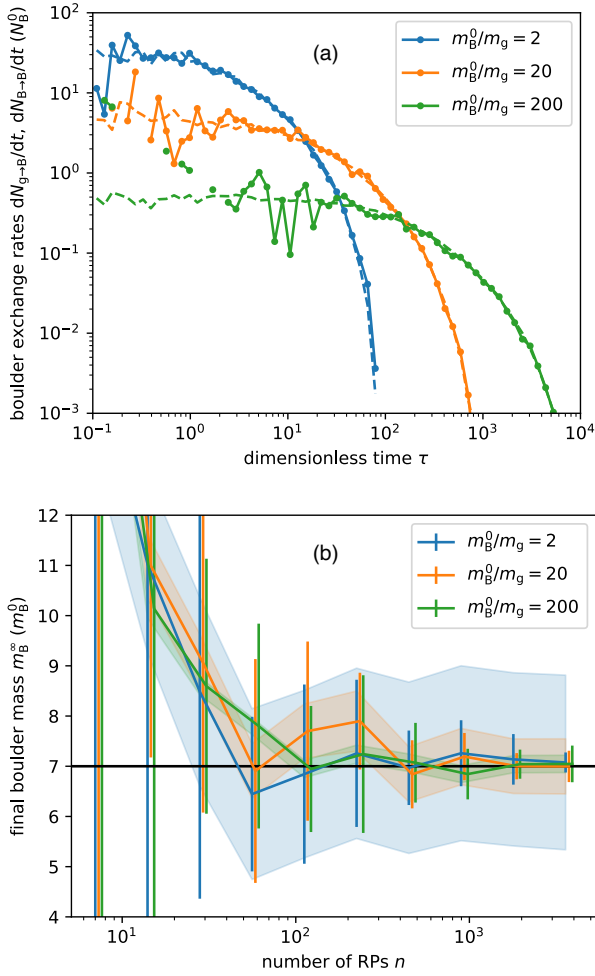


Fig. 9. Further analysis of the RPMC simulation of the grains-and-boulders model: (a) contributions to dN_B/dt : $dN_{B \rightarrow B}/dt$ (solid curve), $-dN_{B \rightarrow B}/dt$ (dashed curve); (b) statistics of final boulder mass m_B^{∞} as a function of RP count n : standard deviation of the mean over 10 runs (error bars) and physical spread σ_{m_B} (filled area). Simulation parameters as in Fig. 8. In (a), the full $n = 1792$ RPs were used; in (b), the number of RPs was varied. n was divided into n_B boulder-mass RPs and $n_g = 6n_B$ grain-mass RPs with equal-weight swarms, $M_i = M_j \forall i, j \in \{1, \dots, n\}$. The horizontal black line indicates the expected final boulder mass $m_B^{\infty} = 7m_B^0$.

depend on the particle masses and the total masses as predicted by Eq. (B.17),

$$\frac{\sigma_{m_B}^{\infty}}{m_B^0} \equiv \lim_{t \rightarrow \infty} \frac{\sigma_{m_B}}{m_B^0} = \left(\frac{M_B^0 m_B^0}{M_g^0 m_g} \right)^{-1/2}. \quad (97)$$

The larger the ratios M_B^0/M_g^0 and m_B^0/m_g , the smaller the physical spread σ_{m_B} .

Next, we consider a mutual collision rate linearly correlated to mass,

$$\Lambda(m, m') = \Lambda_0 (m + m') \quad (98)$$

with some constant Λ_0 . Fig. 10 compares the analytical prediction of the average boulder mass $\langle m \rangle_B$ and its standard deviation σ_{m_B} derived in Appendix B to an equivalent RPMC simulation, finding very good agreement.

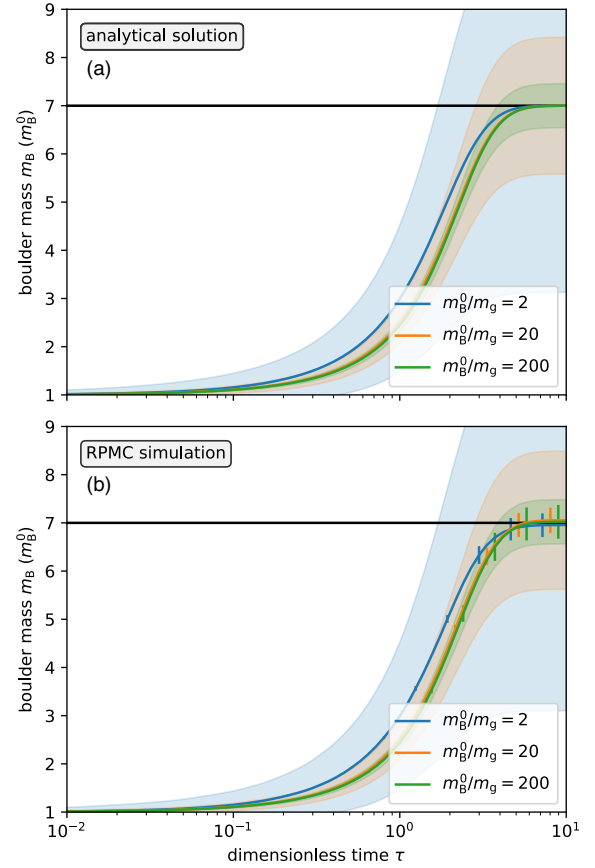


Fig. 10. Average boulder mass $\langle m \rangle_B$ and the physical spread σ_{m_B} for a mass-dependent collision rate $\Lambda(m, m') = \Lambda_0 (m + m')$ for different boulder–grain particle mass ratios: (a) analytical model as given by Eqs. (B.19), (B.21); (b) RPMC simulation. The simulation uses $n = 1792$ RPs divided into $n_B = 256$ boulder-mass RPs and $n_g = 6n_B = 1536$ grain-mass RPs with equal-weight swarms, $M_i = M_j \forall i, j \in \{1, \dots, n\}$. The boulder masses in (b) are averaged over 10 runs; the filled area indicates the physical spread σ_{m_B} of the mass distribution, and the error bars indicate the standard deviation of the mean boulder mass over 10 runs. The horizontal black line indicates the expected final boulder mass $m_B^{\infty} = 7m_B^0$. The total mass ratio of grains and boulders is $M_B^0/M_g^0 = 1/6$, so boulders are expected to grow to 7 times their initial mass, as indicated by the horizontal black line. The horizontal axes refer to dimensionless time $\tau = \Lambda N_g^0 t$.

7. Discussion

The proposed extensions to the RPMC method are strictly additive in the sense that they do not change how the method operates in the many-particles regime to which the original RPMC method was constrained. As such, the considerations of Zsom & Dullemond (2008, Sect. 3.2) with regard to the required number of representative particles continue to apply. In Sect. 3.3 of the same paper, two main limitations of the method are discussed. The first limitation – that swarms must consist of many particles – is overcome by our extension, but the other – that, as an explicit method, the RPMC method is ill-suited for simulating ‘stiff’ problems in which two adverse effects nearly balance each other – remains and is in fact aggravated by our extension of the method. When following the traditional implementation strategy, the variability of the number of representative particles in the extended method adds substantially to the computational cost; however, in a companion paper, we devise a different implementation strategy which is much less sensitive to the number of RPs.

7.1. Limitations

The original RPMC method was not well-suited to simulate systems in which two strong effects nearly balance each other, leading only to a small net change. As an example, [Zsom & Dullemond \(2008, Sect. 3.3\)](#) suggest a system in which coagulation and fragmentation processes lead to a ‘semi-steady state’ in which particles grow and fragment many times, and explains that, as an explicit method, the RPMC method must simulate the individual growth and fragmentation processes, rendering the problem ‘stiff’. This limitation is not removed by our extension of the method, and can in fact be further aggravated by it: If a RP in such a system grows large enough that its swarm particle count falls below the particle number threshold N_{th} , it will be split up to individual RPs. If all of these RPs subsequently fragment to small particles which then recommence the growth cycle, the number of RPs will steadily grow while the particle number distribution remains stable. With an ever-increasing number of RPs, the simulation will quickly exceed its computational resources.

The root of this problem is an asymmetry in our extension of the method: we split up swarms as their particle mass grows larger, but we never merge multiple swarms into one. Unlike splitting, which can be done without consequences for the statistical outcome of the simulation, merging usually entails a loss of information because the particle properties $\mathbf{q}_j, \mathbf{q}_k$ of two swarms j, k to be merged will rarely be exactly identical, and hence have to be averaged over in some way. A merging step was omitted for simplicity, and because it was deemed unnecessary for our main goal in extending the method, which is to simulate planetesimal growth to and beyond the runaway growth regime. However, there is nothing fundamentally precluding a merging step in the simulation, and such an addition would indeed be necessary before the method can be used to simulate semi-steady processes (which would however retain their stiffness even with merging). Merging was an integral part of the Monte Carlo method of [Ormel & Spaans \(2008\)](#); in this method, different ‘species’ (the semantic equivalent of swarms in the RPMC method) could be merged by averaging their properties. A concrete example of such a merging prescription is given in [Krijt et al. \(2015, Sect. 3.4\)](#). A similar approach could in principle be incorporated into the RPMC method.

7.2. Computational cost

It was mentioned that a growing number of RPs may significantly increase the computational cost of the simulation. In the traditional implementation described in [Zsom & Dullemond \(2008\)](#), n^2 RP–swarm interaction rates must be computed, stored, and updated for n RPs. Each time a swarm is split up into N_{th} individual RPs, this cost increases accordingly. The companion paper (Beutel et al., in prep.) explores an alternative sampling method which avoids computing all n^2 RP–swarm interaction rates. Using a bucketing scheme, this alternative sampling method makes the computational cost less sensitive to the number of RPs n and more dependent on the breadth of the occupied subspace of the parameter space \mathbb{Q} , and thereby makes the improved RPMC method computationally viable.

7.3. Boosting

In Sect. 4.4, a boosting scheme was proposed to ensure efficient simulation of growth processes where large bodies accrete large amounts of small bodies. Although an intuitive justification for the approach was given, it remained unclear how it impacted the

statistical result of the simulation. This impact can be studied by means of the grains-and-boulders example:

In Sect. 6.3, we studied the influence of the boulder–grain mass ratio $m_{\text{B}}^0/m_{\text{g}}$ on the simulation outcome. We predicted (cf. Fig. 7) and confirmed numerically (cf. Fig. 8) a spread of the final boulder mass that was anticorrelated with the boulder–grain mass ratio $m_{\text{B}}^0/m_{\text{g}}$. Now, a boost factor $\beta_{jk} > 1$ would be equivalent to accreting more massive grains with a lower collision rate, and the effect of boosting is therefore quantitatively similar to the spread of the final boulder mass distribution in Eq. (97). Although the spread of the mass distribution increases as the boulder–grain mass ratio is increased, it was found that the fidelity of the RPMC method, quantified by the uncertainty of the expected final boulder mass, did not depend on the mass ratio (cf. Fig. 8b). Analogously, boosting seems justifiable provided that the spread of the mass distribution incurred does not exceed the desired precision of the result. This requirement may be significant when dealing with sharply peaked distributions as in the grains-and-boulders example, but it becomes much less restricting for ensembles with a broad mass distribution such as the standard coagulation tests (cf. Fig. 2) which were conducted with a mass accumulation threshold of $q_m = 5\%$, leading to no significant deviation from analytical predictions.

7.4. Avoiding non-integral particle numbers

As mentioned in Sect. 4.3, non-integral swarm particle counts N_k pose a challenge when swarms become few-particles swarms and are thus split up to individual RPs. We note that the boost factor implicitly helps mitigating this problem. A many-particles swarm can become a few-particles swarm in two scenarios: (1) by accumulating mass through a collision in the many-particles regime, or (2) by losing mass through a collision in the few-particles regime. Equation (48) constrains the boost factor such that, in scenario (1), the new swarm particle count does not fall below the particle number threshold, $N'_j \geq N_{\text{th}}$, thus obtaining a ‘precision landing’ of $N'_j = N_{\text{th}}$ if permitted by the other constraints. For the few-particles regime, by exhausting the constraint $\beta_{jk} \leq N_k/N_j$ given in Eq. (49) we can let swarm j deplete swarm k completely by choosing $\beta_{jk} = N_k/N_j$, again if permitted by the other constraints.

8. Summary and conclusions

We have presented a method for stochastic simulation of particle–particle interactions. The method builds upon the RPMC method introduced by [Zsom & Dullemond \(2008\)](#). The most severe limitation of the RPMC method was that, due to its asymmetric construction, the growth of a given representative particle was effectively constrained by the amount of mass it represented. Our extension of the method overcomes this growth barrier by allowing the coexistence of, and the transition between, representative particles and individual, self-representing particles. With our method, it is now possible to accurately model runaway growth processes while largely retaining the structural advantages of the RPMC method such as the stability of the sampling weights.

In Sect. 2, we gave a formal definition of the RPMC method, which we subsequently extended in Sect. 4.3. In Sect. 5 we proved for a generalised variant of the grains-and-boulders example given in [Zsom & Dullemond \(2008\)](#) that, despite the asymmetry in the modelling of interactions, the RPMC method is in fact statistically balanced and produces correct results.

Finally, in Sect. 6 we conducted extensive numerical testing to validate that the extended RPMC method indeed produces plausible results for different cases of orderly growth and runaway growth.

In our extended version of the RPMC method, representative particles are added to the simulation as particles transition between the representative many-particles regime and the individual single-particle regime. We have thereby abandoned a key feature of the original RPMC method: a stable number of representative particles, which was useful because it allowed for predictable performance and simple implementation. In a traditional implementation of the RPMC method, adding new representative particles would introduce substantial computational cost. In the companion paper (Beutel et al., in prep.), we present an implementation strategy that not only alleviates the cost of adding near-identical representative particles to the simulation but can also overcome the $O(n^2)$ computational complexity and storage requirements of the traditional method.

With our extension, the RPMC method can be used to study growth processes over a vast dynamic range, extending into the oligarchic regime where individual bodies arise which cannot be treated stochastically. In this regard, it is similar to the Monte Carlo method of Ormel & Spaans (2008); in the few-particles regime, our method indeed faces similar challenges. A possible advantage of our method over other approaches may be that its main benefit, constant resolution through the intrinsic preservation of an equal-mass sampling, is retained. Interactions between many-particle swarms keep swarm masses unchanged; and swarms can lose but not gain mass through interactions with few-particles swarms, which implies that the resolution is never impaired.

With the proposed method, planetary growth processes can now be studied for the entire range of masses up to full-grown planets. Regimes of runaway growth or oligarchic growth, where individual particles accumulate large fractions of the total mass, can be faithfully represented. In addition, the transition to individual particles gives rise to the possibility of embedding a representative particle Monte Carlo simulation in another type of simulation which traces particles individually, for example a gravitational N -body simulation.

The original RPMC method has been used for coagulation problems involving more than just the mass as a particle property, for instance spatial resolution (Drażkowska et al. 2013, 2014) or additional intrinsic particle properties (e.g. Zsom et al. 2010; Krijt & Ciesla 2016; Krijt et al. 2016; Windmark et al. 2012). The added capability, discussed in this paper, of transitioning to the few-particles regime allows to address a variety of problems involving the growth from pebbles, via planetesimals, to planets; runaway and oligarchic growth, and the subsequent N -body dynamics between the oligarchs, possibly driven by pebble accretion onto these oligarchs; or planetesimal and planet formation in dust rings while incorporating the N -body dynamics of these objects, with the newly formed planets starting to stir the pebbles and planetesimals in the ring.

Acknowledgements. We would like to thank Robert Strzodka, Til Birnstiel, Sebastian Stammer, Joanna Drażkowska, Tommy Chi Ho Lau (劉智昊), and Natalia Dzyurkevich for valuable feedback and useful discussions. We also thank the referee, Sebastiaan Krijt, for his insightful comments and questions which helped us to improve the paper significantly. This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2181/1-390900948 (the Heidelberg STRUCTURES Excellence Cluster).

References

- Armitage, P. J. 2017, ArXiv e-prints [arXiv:[astro-ph/0701485](https://arxiv.org/abs/1707.01485)]
 Birnstiel, T., Dullemond, C. P., & Brauer, F. 2010, *A&A*, **513**, A79
 Birnstiel, T., Fang, M., & Johansen, A. 2016, *Space Sci. Rev.*, **205**, 41
 Bitsch, B., Lambrechts, M., & Johansen, A. 2015, *A&A*, **582**, A112
 Blum, J. 2018, *Space Sci. Rev.*, **214**, 52
 Bukhari Syed, M., Blum, J., Wahlberg Jansson, K., & Johansen, A. 2017, *ApJ*, **834**, 145
 Drażkowska, J., Windmark, F., & Dullemond, C. P. 2013, *A&A*, **556**, A37
 Drażkowska, J., Windmark, F., & Dullemond, C. P. 2014, *A&A*, **567**, A38
 Drażkowska, J., Li, S., Birnstiel, T., Stammer, S. M., & Li, H. 2019, *ApJ*, **885**, 91
 Dullemond, C. P., & Dominik, C. 2005, *A&A*, **434**, 971
 Emsenhuber, A., Mordasini, C., Burn, R., et al. 2021, *A&A*, **656**, A69
 Gillespie, D. T. 1975, *J. Atmos. Sci.*, **32**, 1977
 Greenberg, R., Hartmann, W. K., Chapman, C. R., & Wacker, J. F. 1978, *Icarus*, **35**, 1
 Güttler, C., Blum, J., Zsom, A., Ormel, C. W., & Dullemond, C. P. 2010, *A&A*, **513**, A56
 Johansen, A., Oishi, J. S., Mac Low, M.-M., et al. 2007, *Nature*, **448**, 1022
 Johansen, A., Youdin, A., & Mac Low, M.-M. 2009, *ApJ*, **704**, L75
 Kokubo, E., & Ida, S. 1998, *Icarus*, **131**, 171
 Krijt, S., & Ciesla, F. J. 2016, *ApJ*, **822**, 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
 Lambrechts, M., & Johansen, A. 2012, *A&A*, **544**, A32
 Ohtsuki, K., Nakagawa, Y., & Nakazawa, K. 1990, *Icarus*, **83**, 205
 Okuzumi, S., Tanaka, H., & Sakagami, M.-a. 2009, *ApJ*, **707**, 1247
 Okuzumi, S., Tanaka, H., Kobayashi, H., & Wada, K. 2012, *ApJ*, **752**, 106
 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
 Ormel, C. W., Dullemond, C. P., & Spaans, M. 2010, *Icarus*, **210**, 507
 Ormel, C. W., Liu, B., & Schoonenberg, D. 2017, *A&A*, **604**, A1
 Safronov, V. S. 1964, *Probl. Cosmogony*, **6**, 71
 Schneider, A. D., & Bitsch, B. 2021, *A&A*, **654**, A71
 Smoluchowski, M. V. 1916, *Phys. Zeitsch.*, **17**, 557
 Stammer, S. M., Birnstiel, T., Panić, O., Dullemond, C. P., & Dominik, C. 2017, *A&A*, **600**, A140
 Tanaka, H., Himeno, Y., & Ida, S. 2005, *ApJ*, **625**, 414
 Trubnikov, B. A. 1971, *Sov. Phys. Dokl.*, **16**, 124
 Wahlberg Jansson, K., & Johansen, A. 2014, *A&A*, **570**, A47
 Wahlberg Jansson, K., Johansen, A., Bukhari Syed, M., & Blum, J. 2017, *ApJ*, **835**, 109
 Weidenschilling, S. J. 1980, *Icarus*, **44**, 172
 Wetherill, G. W. 1990, *Icarus*, **88**, 336
 Wetherill, G. W., & Stewart, G. R. 1993, *Icarus*, **106**, 190
 Windmark, F., Birnstiel, T., Güttler, C., et al. 2012, *A&A*, **540**, A73
 Wurm, G., & Teiser, J. 2021, *Nat. Rev. Phys.*, **3**, 405
 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
 Zsom, A., Ormel, C. W., Dullemond, C. P., & Henning, T. 2011a, *A&A*, **534**, A73
 Zsom, A., Sándor, Z., & Dullemond, C. P. 2011b, *A&A*, **527**, A10

Appendix A: Analytical solutions and equal-weight samplings for standard coagulation tests

For the constant kernel Eq. (88) and initial conditions $f(m, 0) = N^0 \delta_D(m - m^0)$, the Smoluchowski equation (90) has the analytical solution

$$\begin{aligned} f(m, t) &= N^0 g^2 (1 - g)^{m/m^0 - 1}, \\ g &\equiv \left(1 + \frac{\tau}{2}\right)^{-1}, \\ \tau &\equiv M \Lambda t \end{aligned} \quad (\text{A.1})$$

with the total dimensionless mass M (e.g. Ohtsuki et al. 1990).

Sampling an initial set of RPs with equal-weight swarms is easy because the definition of the number density distribution $f(m, 0)$ implies that all RPs have the same mass initially. We simply select an ensemble of n RPs with homogeneous particle masses m^0 and swarm masses M/n .

For the linear kernel Eq. (89) with an initial number density distribution

$$\begin{aligned} f(m, 0) &= N_0 \bar{m}^{-1} e^{-m/\bar{m}}, \\ N_0 &\equiv \frac{M}{\bar{m}}, \end{aligned} \quad (\text{A.2})$$

and the initial average mass \bar{m} , the Smoluchowski equation has the analytical solution

$$\begin{aligned} f(m, t) &= N_0 m^{-1} \frac{g}{\sqrt{1-g}} e^{-m/\bar{m}(2-g)} I_1\left(2 \frac{m}{\bar{m}} \sqrt{1-g}\right), \\ g &\equiv e^{-\tau}, \\ \tau &\equiv M \Lambda_0 t \end{aligned} \quad (\text{A.3})$$

with the total dimensionless mass M , where $I_1(\cdot)$ is the modified Bessel function of the first kind (e.g. Ohtsuki et al. 1990).

Drawing equal-weight samples for the non-trivial initial mass distribution of the linear kernel test is a bit more involved. Let us assume that some number density distribution $f(m)$ is to be sampled by n RPs of masses m and equal weight

$$M = mN(m) \stackrel{!}{=} \frac{M}{n}. \quad (\text{A.4})$$

Because of the equal-weight sampling, masses of RPs must instead be drawn from the *mass-weighted density distribution*

$$f_R(m) = M^{-1} m f(m). \quad (\text{A.5})$$

For the initial number density distribution of the linear kernel test (Eq. (A.2)), the mass-weighted density distribution is

$$f_R(m, 0) = \bar{m}^{-1} \frac{m}{\bar{m}} e^{-m/\bar{m}}. \quad (\text{A.6})$$

We can then draw RP masses with inverse transform sampling by first computing the cumulative representative number density distribution

$$\begin{aligned} F_R(m, 0) &= \int_0^m dm' f_{RP}(m, 0) \\ &= 1 - \left(\frac{m}{\bar{m}}\right) e^{-m/\bar{m}}, \end{aligned} \quad (\text{A.7})$$

then inverting the relation $\xi = F_R(m, 0)$,

$$m = -\bar{m} \left[1 + W_{-1} \left(\frac{-1 + \xi}{e} \right) \right], \quad (\text{A.8})$$

where $W_k(z)$ is the Lambert W function, and then drawing a uniform random number $\xi \in [0, 1)$.

An analytical solution for the product kernel Eq. (92) was constructed by Trubnikov (1971) and made applicable for times $\eta > 1$ by Wetherill (1990). The particle number density is

$$\begin{aligned} f(k, \eta) &= \frac{(2k)^{k-1}}{k!k} \left(\frac{\eta}{2}\right)^{k-1} e^{-k\eta}, \\ k &\equiv \frac{m}{m^0}, \\ \eta &\equiv \Lambda_0 N^0 t, \end{aligned} \quad (\text{A.9})$$

where m^0 is the initial particle mass, N^0 is the initial number of particles of mass m^0 , $k \in \mathbb{N}$ is a dimensionless mass argument, and η is a dimensionless time coordinate.

As with the constant kernel test, all particles have the same mass initially, making the sampling of equal-weight swarms trivial.

Wetherill (1990) explains that, although the equation solved by Trubnikov (1971) becomes inconsistent for $\eta > 1$ when a runaway particle has separated from the continuous distribution, the solution in Eq. (A.9) still correctly describes the small body mass distribution, that is, the mass distribution of all particles except for the runaway particle. Using conservation of mass, we can infer the mass of the runaway particle by direct summation of Eq. (A.9).

Appendix B: Analytical model for grains-and-boulders test

In this section we develop an analytical model for the grains-and-boulders example used in Sect. 5. We derive a system of differential equations for the remaining number of grains N_g , the average boulder mass $\langle m \rangle_B$ and the variance of the boulder mass distribution $\sigma_{m_B}^2$, and we give analytical expressions for the cases of constant and linear mutual collision rates, cf. Eqs. (94, 98).

For any given grain-mass RP, the grain-boulder collision rate is

$$\lambda_g = \sum_{\substack{j=1 \\ m_j \geq m_B^0}}^N \Lambda_j = N_B \langle \Lambda \rangle_B, \quad (\text{B.1})$$

where we define the boulder average $\langle X \rangle_B$ of a particle-specific quantity X_j as

$$\langle X \rangle_B \equiv \frac{1}{N_B} \sum_{\substack{j=1 \\ m_j \geq m_B^0}}^N X_j, \quad (\text{B.2})$$

(note that in Eq. (66) the boulder average was defined differently in the context of the RPMC method) and where the mutual collision rate of any grain with a boulder j was abbreviated as

$$\Lambda_j := \Lambda(m_g, m_j), \quad (\text{B.3})$$

The number of grains N_g decreases stochastically, so we can only reason about the expected value of N_g . At some time t , the expected value will change over a time increment Δt with the grain–boulder collision rate given in Eq. (B.1),

$$E[N'_g] = N_g (1 - \lambda_g \Delta t), \quad (\text{B.4})$$

where the time dependencies of quantities X are henceforth abbreviated as $X \equiv X(t)$, $X' \equiv X(t + \Delta t)$. We thus obtain the differential equation

$$\frac{dN_g}{dt} = -N_B N_g \langle \Lambda \rangle_B. \quad (\text{B.5})$$

The growth of a given boulder mass $m_j \geq m_B^0$ is described by a Poisson point process. The chance that boulder j accumulates k grains during a given time increment Δt ,

$$m'_j = m_j + km_g, \quad (\text{B.6})$$

is given by the Poisson distribution with the probability mass function $\text{Pois}_k(\lambda_{B,j} \Delta t)$ defined as

$$\text{Pois}_k(x) = \frac{x^k e^{-x}}{k!} \quad (\text{B.7})$$

for which the expected value of k equals the variance,

$$E[k] = \text{Var}[k] = x. \quad (\text{B.8})$$

For a time increment Δt , the expected average value of m'_j is thus

$$E[\langle m' \rangle_B] = \langle m \rangle_B + m_g \langle \Lambda \rangle_B \Delta t. \quad (\text{B.9})$$

By taking the difference quotients to the limit $\Delta t \rightarrow 0$, we find

$$\frac{d\langle m \rangle_B}{dt} = m_g N_g \langle \Lambda \rangle_B, \quad (\text{B.10})$$

where we again note that the number of boulders cannot change, $N_B = \text{const}$. By relating Eqs. (B.5) and (B.10), the average boulder mass $\langle m \rangle_B$ can thus be expressed in terms of the number of grains N_g as

$$\langle m \rangle_B = m_B^0 + m_g \frac{N_g^0 - N_g}{N_B}. \quad (\text{B.11})$$

To find a differential equation for the variance of the boulder mass $\sigma_{m_B}^2$ as a function of time, we evaluate the variance at times t and $t + \Delta t$,

$$\sigma_{m_B}^2 = \langle m^2 \rangle_B - \langle m \rangle_B^2, \quad (\text{B.12})$$

$$\begin{aligned} \sigma_{m'_B}^2 &= \langle m'^2 \rangle_B - \langle m' \rangle_B^2 \\ &= \frac{1}{N_B} \sum_{j=1}^N \sum_{k=0}^{\infty} \text{Pois}_k(N_g \Lambda_j \Delta t) (m_j + km_g)^2 \\ &\quad - [\langle m \rangle_B + m_g N_g \langle \Lambda \rangle_B \Delta t]^2. \end{aligned} \quad (\text{B.13})$$

By subtracting Eq. (B.12) from Eq. (B.13) and taking the difference quotient for $\Delta t \rightarrow 0$, we obtain

$$\frac{d\sigma_{m_B}^2}{dt} = \langle \Lambda \rangle_B N_g m_g^2 + 2m_g [\langle m \Lambda \rangle_B - \langle m \rangle_B \langle \Lambda \rangle_B], \quad (\text{B.14})$$

where the first term describes the shot noise originating from the discrete nature of the collision events, and the second term represents the covariance of boulder masses and collision rates.

For a constant mutual collision rate, $\Lambda(m, m') = \Lambda_0$, Eq. (B.5) is solved by exponential decay,

$$N_g = N_g^0 e^{-\lambda_g t}, \quad (\text{B.15})$$

where N_g^0 is the initial number of grains and the grain–boulder collision rate λ_g in Eq. (B.1) simplifies to

$$\lambda_g = N_B \Lambda_0. \quad (\text{B.16})$$

Because collision rates and boulder masses are uncorrelated, the variance is fully determined by shot noise, and we can directly integrate Eq. (B.14):

$$\sigma_{m_B}^2 = m_g^2 \frac{\langle m \rangle_B - m_B^0}{m_g}, \quad (\text{B.17})$$

where we note that the initial variance is zero, $(\sigma_{m_B}^0)^2 = 0$, because all boulders have the same mass initially. In a homogeneous Poisson point process, the variance of the number of collisions equals the expected number of collisions. We see that this also holds true for our non-homogeneous Poisson process if collision rates do not depend on particle masses.

Let us now study a linear mutual collision rate, $\Lambda(m, m') = \Lambda_0 (m + m')$ with some constant Λ_0 . By evaluating Eq. (B.2), we find

$$\langle \Lambda \rangle_B = \Lambda_0 (m_g + \langle m \rangle_B). \quad (\text{B.18})$$

Eq. (B.5) then has the solution

$$N_g = N_g^0 \frac{b}{(b - a) \exp[b N_g^0 t] + a}, \quad (\text{B.19})$$

where we defined the abbreviations

$$\begin{aligned} a &:= \Lambda_0 m_g, \\ b &:= a + \Lambda_0 (m_g + m_B^0) \frac{N_B}{N_g^0}. \end{aligned} \quad (\text{B.20})$$

Using Eq. (B.14) and identifying the variance as per Eq. (B.12), we find the variance of the average boulder mass governed by the differential equation

$$\frac{d\sigma_{m_B}^2}{dt} = \langle \Lambda \rangle_B N_g m_g^2 + 2m_g \Lambda_0 \sigma_{m_B}^2, \quad (\text{B.21})$$

where the covariance term describes the broadening of the variance due to the mass dependency of collision rates: smaller boulders will grow more slowly than average, whereas larger boulders will sweep up grains faster.

Appendix C: RPMC approximation and statistical balance

In Item 2 of Sect. 4.2, one of the problems with extending the RPMC method to few-particles swarms was identified in the fact that the simplified RP–swarm collision rate of Eq. (23) overestimates the RP–swarm collision rate Eq. (19), which

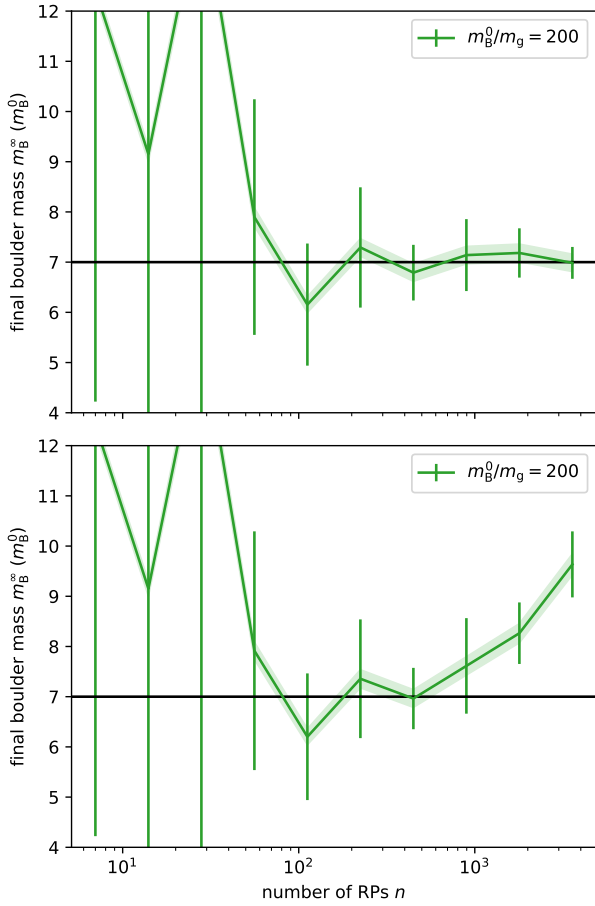


Fig. C.1: As in Fig. 9b, but with an initial number of only $N_g^0 = 8 \cdot 10^6$ grains: (a) using the approximate collision rates $\tilde{\lambda}_{jk} \approx \tilde{\lambda}_{jk} = \lambda(\mathbf{q}_j, \mathbf{q}_k)$, and (b) using the true collision rates $\tilde{\lambda}_{jk} = (1 - \delta_{jk})/2\lambda(\mathbf{q}_j, \mathbf{q}_k)$ instead. As can be seen, using the true collision rates in the many-particles regime would distort the statistical balance of the RPMC method.

becomes significant for $N_k \sim 1$. An obvious way of attempting to mitigate this problem would be to use the RP-swarm collision rate $\tilde{\lambda}_{jk}$ instead of its approximation $\tilde{\lambda}_{jk}$. But while this would lead to a statistically correct interaction rate, the resulting mass distribution would be skewed. In Fig. C.1, it can be seen that the approximation $\tilde{\lambda}_{jk} \approx \tilde{\lambda}_{jk}$ must be made for the RPMC method to be statistically balanced: without the approximation, the final boulder mass m_B^∞ does not converge to the expected value of $7m_B^0$ as the number of RPs n is increased.

The statistical imbalance can be understood by considering the different origins of the N_g and N_j factors in Eqs. (78) and (72): some originate from collision rates $\tilde{\lambda}_{B,j}$ and $\tilde{\lambda}_{g,j}$ and would hence be adjusted by summands of $-1/2$ and $-n_g/2$, respectively, whereas others came from summation over all grain-mass and boulder-mass swarms and thus are not adjusted. The statistical balance equation (80) would then end up different from 0 with a spurious dependence on the simulation parameters n_g and n_B^0 .

Appendix D: List of symbols

A cross-referenced overview of the most commonly used symbols in this work is given in Table D.1.

| Symbol | Description | Reference |
|------------------------------------|---|---------------------|
| β_{jk} | boost factor | Sect. 4.4 |
| $\Lambda(\mathbf{q}, \mathbf{q}')$ | raw grain–boulder collision rate | Sect. 5 |
| Λ_j | mutual collision rate between boulder j and a grain | Eq. (63); Eq. (B.3) |
| $\lambda(\mathbf{q}, \mathbf{q}')$ | raw collision rate | Sect. 3.1 |
| λ_{jk} | true collision rate (with self-collision suppressed) | Eq. (4) |
| λ_j | cumulative collision rate of particle j | Eq. (5) |
| λ | total rate of collisions | Eq. (6) |
| $\bar{\lambda}_{jk}$ | RP–swarm collision rate | Eq. (19) |
| $\bar{\lambda}_j$ | cumulative collision rate of RP j | Eq. (20) |
| $\bar{\lambda}$ | total rate of collisions of representative particles | Eq. (9) |
| $\tilde{\lambda}_{jk}$ | simplified RP–swarm collision rate | Eq. (23) |
| $\tilde{\lambda}_j$ | simplified cumulative collision rate of RP j | Eq. (24) |
| $\tilde{\lambda}$ | simplified total rate of collisions of representative particles | Eq. (24) |
| $\tilde{\lambda}_{jk}^b$ | boosted collision rate | Eq. (41) |
| λ_g | grain–boulder collision rate | Eq. (B.1) |
| $\tilde{\lambda}_{gj}$ | RP–swarm collision rate for a grain-mass RP with boulder j | Eq. (64) |
| $\tilde{\lambda}_g$ | cumulative grain-mass RP collision rate | Eq. (65) |
| $\sigma_{m_B}^2$ | variance of boulder masses | Eq. (B.12) |
| $\mathcal{I}_g, \mathcal{I}_B$ | index sets of grains and boulders | Eqs. (61) |
| M | total mass in the system | Sect. 3.1 |
| M_k | total mass of particles in swarm k | Eq. (13) |
| m_g | grain mass | Sect. 5 |
| m_B^0 | initial boulder mass | Sect. 5 |
| N | number of physical particles | Sect. 3.1 |
| N_k | number of physical particles in swarm k | Sect. 3.1 |
| N_{jk} | swarm multiplicity factor | Eq. (37) |
| N_{jk}^b | boosted swarm multiplicity factor | Eq. (42) |
| N_{th} | particle number threshold | Sect. 4.3 |
| N'_g | number of grains after duration Δt | Eq. (68) |
| $N'_{B \rightarrow B}$ | number of boulders remaining in boulder-mass swarms after duration Δt | Eq. (70) |
| $N'_{g \rightarrow B}$ | number of boulders emerging from grain-mass swarms after duration Δt | Eq. (76) |
| N_g | number of grains | Eq. (62) |
| N_g^0 | initial number of grains | Sect. 5 |
| N_B | number of boulders | Eq. (62) |
| n | number of representative particles | Sect. 3.2 |
| q_m | mass accumulation threshold | Sect. 4.4.2 |
| \mathbf{q}_k | properties of particle k | Sect. 3.1 |
| \mathbf{q}^{coll} | properties of particle sampled from collision outcome | Sect. 3.5 |
| W_k | sampling weight of RP k | Eq. (34) |
| $\langle \cdot \rangle_B$ | boulder average | Eq. (66); Eq. (B.2) |

Table D.1: List of commonly used symbols.