

## CONVERGENCE OF A GENERALIZED WEIGHTED FLOW ALGORITHM FOR STOCHASTIC PARTICLE COAGULATION

LEE DEVILLE

Department of Mathematics  
University of Illinois at Urbana-Champaign, 1409 W. Green Street  
Urbana, IL 61801, USA

NICOLE RIEMER

Department of Atmospheric Sciences  
University of Illinois at Urbana-Champaign, 1301 W. Green Street  
Urbana, IL 61801, USA

MATTHEW WEST

Department of Mechanical Science and Engineering  
University of Illinois at Urbana-Champaign, 1206 W. Green Street  
Urbana, IL 61801, USA

(Communicated by Nawaf Bou-Rabee)

**ABSTRACT.** We introduce a general family of Weighted Flow Algorithms for simulating particle coagulation, generate a method to optimally tune these methods, and prove their consistency and convergence under general assumptions. These methods are especially effective when the size distribution of the particle population spans many orders of magnitude, or in cases where the concentration of those particles that significantly drive the population evolution is small relative to the background density. We also present a family of simulations demonstrating the efficacy of the method.

**1. Weighted methods for coagulation equations.** Predicting the evolution of particle size distributions that are multi-dimensional in composition space and undergo coagulation is a challenge in many areas in science and engineering. Examples include the modeling of aerosol particles in the Earth’s atmosphere [27, 33, 38, 46], and of aerosol processes in industrial applications [10].

Recently, particle-resolved models have been introduced to tackle this problem [9, 32, 39]. These stochastic models fully resolve the multi-dimensional particle distribution as they simulate a representative group of particles distributed in composition space. Coagulation, condensation and evaporation, and other important processes can then be treated on an individual particle level. The first rigorous application of such a Monte Carlo approach was suggested by Gillespie, who developed the exact Stochastic Simulation Algorithm [16, 17, 18, 19] to treat the stochastic collision-coalescence process in clouds.

A straight-forward implementation of these methods assumes that each computational particle represents a physical particle. However, this naive approach can

---

2010 *Mathematics Subject Classification.* Primary: 65C35, 60K35; Secondary: 60J28.

*Key words and phrases.* Stochastic simulation, Martingales, particle coagulation, Smoluchowski equation.

quickly become computationally infeasible, the reason being the diversity of the underlying particle population—either with respect to size or to composition—and the relative abundance of the sub-populations. There are two distinct conceptual scenarios in which these problems arise, which we term the “size multiscaling problem” and the “compositional multiscaling problem”.

For the “size multiscaling problem”, consider the example of initially monodisperse aerosol particles of sub-micrometer size coagulating due to Brownian diffusion in the atmosphere [26, Chapter 16.4.2]. After some time of coagulation, the range of particle sizes present in the population can span many orders of magnitude, from tens of nanometers to several microns in diameter. Resolving the full particle size range is important since the small particles dominate the particle number concentration whereas the large particles dominate the particle mass concentration, and one typically aims to keep track of both. Moreover, the most likely coagulation events involve interactions of small and large particles, since the interactions are more likely to take place between particles with different diffusion properties [40, Chapter 13.3.1.3]. However, the number concentrations of particles in the sub-micron size range are typically several orders of magnitude larger than the particles in the super-micron size range [40, Chapter 8.2]. Therefore, to resolve the entire size spectrum accurately, the total number of computational particles in the simulation must be very large, otherwise the population of large particles will be under-resolved and the accuracy of the simulation significantly compromised. We will address this very problem in Section 6, where we simulate a cloud droplet population undergoing collision-coalescence, producing very few, but for the dynamics of the system crucially important, large drizzle drops [29].

A conceptually different problem is the “compositional multiscaling problem”; this arises when a certain type of particle is of special interest, and particles of this type are much rarer compared to the rest of the particles in the given size range. For example, the ability of an atmospheric aerosol population to absorb solar radiation is directly related to the aerosol impact on climate and hence is an important quantity [40, Chapter 24.2]. One of the few absorbing aerosol types is soot, a by-product of the incomplete combustion of carbon containing material [36]. Thus, to accurately predict how much solar radiation the aerosol population absorbs, the soot sub-population must be adequately captured. The size distribution of soot aerosol peaks in the accumulation mode range [22], a size range where many other particle types are present that do not contain soot [21, 37]. If the conditions are such that the number concentration of the soot sub-population is small compared to the non-soot particles, either a large number of computational particles needs to be expended to resolve the soot particles or the prediction of absorptivity will be inaccurate.

What this has in common with the previous example, however, is that there is an important sub-population of particles that have concentrations that are several orders of magnitude lower than other sub-populations. A naive representation of the particle population either leads to under-resolving the small but important sub-population, or to requiring very large numbers of computational particles in the simulation. One strategy to circumvent these difficulties is to use the concept of “weighted” computational particles, where a single computational particle corresponds to some number of real particles. This idea has been pursued by several groups in the past; the methods can be broadly divided into two categories, either the weightings are attached to the particles or they are functions of the particle

constituents. Variants of both particle-attached [35, 30, 47, 24, 25, 48, 49] and particle-function weightings [1, 6, 11, 12, 13, 14, 28, 41, 44] have been widely studied.

Building on this literature, and particularly on the work of Kolodko and Sabelfeld [28], we presented the development and application of the Weighted Flow Algorithms (WFA) for particle-resolved methods in DeVille et al. [7], as a generalized way of weighting computational particles to improve computational efficiency. Our analysis was able to demonstrate general conditions for consistency and optimality of this class of weighted methods. Applying this to the simulation of atmospheric aerosol particles, we used weighting functions that were power laws in the total particle mass, and showed that this can greatly increase the accuracy and performance of the simulation. One particular feature of these simulations was their tunability that allowed them to be optimized for different observables, e.g., one could choose parameters to make a simulation accurate in measuring particle number, and then change these parameters to have another simulation optimize for particle mass. Thus we have a family of simulations of such systems, each of which optimizes different observables to a different degree. To obtain a full simulation that takes into account several different observables (e.g., if we would like to track both particle number and mass well) it was necessary to perform several independent simulations, each optimized for their particular observable, and then combine them *a posteriori*. While this was done in DeVille et al. [7], a better method would be one that combines these different weightings during the running of the simulation.

We can now describe the content of this paper as follows: we have developed a method that solves both the size and compositional multiscaling problems at once, and also allows for a run-time mixing of multiple schemes. To do so, we have developed a particle method that allows for several different sub-populations that can interact; these sub-populations can correspond to populations at different length scales (speaking to the size multiscaling problem) or of different types (speaking to the compositional multiscaling problem). Each of these sub-populations can be weighted differently as the problem requires.

We also prove convergence of such methods under very weak assumptions. A more precise statement is included in the theorems below, but in essence we show that there are a family of “consistency conditions” that, when satisfied, will guarantee that a scheme developed with this method will converge to the correct solution in the limit of large numbers of particles. These conditions are easy to verify for a particular scheme. Moreover, we show how to develop an optimal method in the set of consistent methods.

Finally, we present a series of numerical simulations demonstrating the practical efficacy of the multiscaling methods developed in this paper.

**2. Problem formulation and summary of results.** For aerosol dynamics simulations, one of the most expensive portions of the calculations can be the parts that deal with coagulation of particles [15, 16]. The equations that govern coagulation are the Smoluchowski equations.

**2.1. Equations of interest.** The continuous Smoluchowski equations are:

$$\frac{\partial n(\boldsymbol{\mu})}{\partial t} = \frac{1}{2} \int_{[0, \boldsymbol{\mu}]} K(\boldsymbol{\nu}, \boldsymbol{\mu} - \boldsymbol{\nu}) n(\boldsymbol{\nu}) n(\boldsymbol{\mu} - \boldsymbol{\nu}) d\boldsymbol{\nu} - \int_{[0, \infty)} K(\boldsymbol{\nu}, \boldsymbol{\mu}) n(\boldsymbol{\nu}) n(\boldsymbol{\mu}) d\boldsymbol{\nu}, \quad (2.1)$$

where  $K(\boldsymbol{\mu}, \boldsymbol{\nu})$  is the coagulation rate between particles of masses  $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^n$  and we write  $[\mathbf{0}, \boldsymbol{\mu}]$  to denote the set of points that are elementwise between  $\mathbf{0}$  and  $\boldsymbol{\mu}$ .

Now choose any partition of  $[\mathbf{0}, \infty)$ , which we denote  $\{I_n\}_{n=0}^\infty$ . Then if we define

$$K_{ij} := \frac{1}{|I_i||I_j|} \int_{I_i \times I_j} K(\boldsymbol{\mu}, \boldsymbol{\nu}) d\boldsymbol{\mu} d\boldsymbol{\nu},$$

$$\hat{x}_i(t) := \int_{I_i} n(\boldsymbol{\mu}) d\boldsymbol{\mu},$$

it is not hard to see that (2.1) is asymptotically equivalent to

$$\frac{d}{dt} \hat{x}_i(t) = \sum_{j=1}^{i-1} \frac{1}{2} K_{j,i-j} \hat{x}_j(t) \hat{x}_{i-j}(t) - \sum_{j=1}^{\infty} K_{ij} \hat{x}_j(t) \hat{x}_i(t), \quad (2.2)$$

where the asymptotic correction goes to zero as  $\max_j |I_j| \rightarrow 0$ . It could be debated whether we should consider (2.1) or (2.2) as the fundamental object to consider. In this paper, we consider (2.2) as the fundamental object and consider how to simulate it stochastically.

**2.2. Summary of results of the paper.** We first develop a method of particle simulation for the Smoluchowski equation with the added feature that we are allowed to consider multiple, and possibly interacting, particle sub-populations. It is not surprising that one must choose a method so that the mean of the stochastic method corresponds to the evolution we are trying to approximate; however, we show below in Theorem 1 that this is also sufficient to obtain the correct mean for trajectories.

We then develop a family of consistency conditions in Section 4 that give the correct mean during the simulation. It turns out that, due to the many degrees of freedom in the parameters of the stochastic simulation, there are many choices that give consistency, so we also develop the conditions that give the optimal consistent method. Finally, we state and prove a family of theorems that guarantee that these methods are accurate in the limit of large population size.

**3. Formal overview of algorithm.** See Table 1 for a summary of the notation used throughout the paper.

**3.1. Computational coagulation.** We want to represent computational particles by both a mass and a tag representing their “class”. We denote  $\mathbb{M}$  as the set of classes and assume that masses are integer-valued. Thus the state of the system is the function

$$Q: \mathbb{M} \times \mathbb{N}_1 \times \mathbb{R}^+ \rightarrow \mathbb{N}_0, \quad (3.1)$$

$$Q_i^a(t) = \text{“the number of computational particles of class } a \text{ and mass } i \text{ at time } t\text{.”} \quad (3.2)$$

In this context, we use the word class very broadly: all it means is that we have made an *a priori* division of the particles. As can be seen from (3.1), each particle belongs to exactly one class, so the class can be regarded as an arbitrary additional particle property. The class is not a function of the other particle properties. The criterion for assigning particles to classes will depend on the application. As can be seen in Section 6, examples include: classes delineated by mass, with one class being large particles and the other small particles; classes delineated by composition, with one class containing the more common chemical constituents and the other class

Variables	Meaning	Range
$a, b, c, d$	physical particle class	$\mathbb{M} = \{1, 2, 3, \dots, M\}$
$\alpha, \beta, \gamma$	event outcomes	$\{0, 1\}$
$e$	basis vector	$\mathbb{M} \times \mathbb{N}_1 \rightarrow \mathbb{Z}$
$i, j, k$	physical particle size	$\mathbb{N}_1 = \{1, 2, 3, \dots\}$
$K$	coagulation kernel	$[0, \infty)$
$\lambda$	event rate	$[0, \infty)$
$N$	physical number concentration	$[0, \infty)$
$N_p$	total number of computational particles	$\mathbb{N}_0 = \{0, 1, 2, \dots\}$
$p$	probability	$[0, 1]$
$q, Q$	number of computational particles	$\mathbb{N}_0 = \{0, 1, 2, \dots\}$
$\rho$	event rate	$[0, \infty)$
$T$	selection rate function	$[0, \infty)$
$V$	computational volume	$(0, \infty)$
$w$	weighting function	$(0, \infty)$
$x, X$	physical particle concentration	$[0, \infty)$
$y, Y$	computational particle concentration	$[0, \infty)$
$\zeta$	event jump	$\mathbb{M} \times \mathbb{N}_1 \rightarrow \mathbb{Z}$

TABLE 1. Notation and variable ranges.

containing the rare chemical constituents for examples of classes. It may even be that otherwise identical particles are divided between two classes.

We want to consider a splitting method to simulate (2.2) where we use a two-step process to determine which event to perform next. The fundamental computation is the propensity of the pair of particles  $(a, i)$  and  $(b, j)$  to coagulate and become the particle  $(c, i + j)$ . The two steps are:

1. Choose a triplet of particles  $((a, i), (b, j), (c, i + j))$  with some probability, where  $(a, i)$  and  $(b, j)$  are represented by some positive number of computational particles, i.e.,  $q_i^a(t) > 0, q_j^b(t) > 0$ . The rate at which we choose this triplet will be denoted  $T_{ij}^{abc}$ .
2. Next, we decide what to do with this triplet; we can either remove an  $(a, i)$  particle or not, remove a  $(b, j)$  particle or not, and create a  $(c, i + j)$  particle or not. We thus define the eight probabilities  $p_{ij}^{abc}(\alpha, \beta, \gamma)$ , where  $\alpha, \beta, \gamma \in \{0, 1\}$  denote whether the particle of type  $a, b$ , or  $c$  exists after the coagulation.

We require, for all  $a, b, c \in \mathbb{M}$  and  $i, j \in \mathbb{N}_1$ , that

$$\sum_{\alpha, \beta, \gamma \in \{0, 1\}} p_{ij}^{abc}(\alpha, \beta, \gamma) = 1. \quad (3.3)$$

In the standard (what we might call “non-split”) particle method, we would only allow  $\alpha = \beta = 0$  and  $\gamma = 1$ , meaning that when the triplet is chosen, we always destroy the first two particles and create the third. We could then interpret  $T_{ij}^{abc}$  to be the rate of coagulating  $(a, i)$  and  $(b, j)$  and obtaining  $(c, i + j)$ . In this split method, we can see that the propensity to create the particle of type  $(c, i + j)$  in this coagulation is now given by the product  $T_{ij}^{abc} \sum_{\alpha, \beta \in \{0, 1\}} p_{ij}^{abc}(\alpha, \beta, 1)$ , and similarly for the destruction of the other two particles.

One note here about “split” and “non-split” methods. The reader has no doubt noticed that we are allowing for transitions that, e.g., violate mass conservation, because we can create the  $(c, i + j)$  particle without destroying the  $(a, i)$  and  $(b, j)$

particles. If we were interested in simulating a finite population of particles, this would be a problem. However, this algorithm is intended to generate a finite particle estimator of an infinite particle limit, namely the Smoluchowski equations (2.1). This gives additional design flexibility in choosing the particle dynamics, and as we show in Theorems 1 and 2, all of the choices we make below are asymptotically correct. In fact, as we show in several examples in Section 6, using “fictitious” jump laws can actually approximate the infinite particle limit better than those choices which are “realistic”.

Thinking of these transitions as taking place in the context of a continuous time Markov chain (CTMC), we would like to define the rates of the CTMC as

$$\lambda_{ij}^{abc}(Q; \alpha, \beta, \gamma) := T_{ij}^{abc} \cdot p_{ij}^{abc}(\alpha, \beta, \gamma) \cdot Q_i^a(t) (Q_j^b(t) - \delta_{a=b, i=j}), \quad (3.4)$$

and the event jumps by

$$\zeta_{ij}^{abc}(\alpha, \beta, \gamma) := (\alpha - 1)e_i^a + (\beta - 1)e_j^b + \gamma e_{i+j}^c, \quad (3.5)$$

where  $e_i^a$  is the unit vector corresponding to one particle of size  $i$  and class  $a$ .

More concretely, this means that if we choose a family of independent Poisson processes of rate one,  $N_{ij\alpha\beta\gamma}^{abc}(t)$ , then we have

$$Q_i^a(t) = Q_i^a(0) + \sum N_{ij\alpha\beta\gamma}^{abc} \left( \int_0^t \lambda_{ij}^{abc}(Q_i^a(s); \alpha, \beta, \gamma) ds \right) \zeta_{ij}^{abc}(\alpha, \beta, \gamma). \quad (3.6)$$

Notice that if  $\alpha = 0$  (resp.  $\beta = 0$ ), we will remove one  $(a, i)$  (resp.  $(b, j)$ ) particle, and if  $\alpha = 1$  (resp.  $\beta = 1$ ), we will do nothing to the  $(a, i)$  (resp.  $(b, j)$ ) population. Conversely, if  $\gamma = 1$  we will create a particle of type  $(c, i+j)$  and if  $\gamma = 0$  we will do nothing to the  $(c, i+j)$  population. As we said above, the choice  $(\alpha, \beta, \gamma) = (0, 0, 1)$  corresponds to the “direct” coagulation method where we always remove the two smaller particles and always create the larger one. Every other possibility exists in the split scheme, even (in theory)  $(\alpha, \beta, \gamma) = (1, 1, 0)$ , which would have no effect on  $Q$  (of course, we would like this to happen infrequently during an actual simulation, and this is a question dealt with in Section 4.3 below).

Of course, once we specify  $T_{ij}^{abc}$  and  $p_{ij}^{abc}(\alpha, \beta, \gamma)$ , this specifies a particular numerical scheme. The question remains as to whether this corresponds in any sense to an approximation for (2.2). Let us define a computational volume  $V$ , and then define the computational particle concentration  $Y_i^a(t)$ , the physical particle concentration  $X_i^a(t)$ , and the binned Smoluchowski variable  $X_i(t)$  as follows:

$$Y_i^a(t) := \frac{1}{V} Q_i^a(t) \quad (3.7)$$

$$X_i^a(t) := Y_i^a(t) w_i^a \quad (3.8)$$

$$X_i(t) := \sum_{a \in \mathbb{M}} X_i^a(t) \quad (3.9)$$

where  $w_i^a$  is a “weighting function” which can be interpreted as follows: each computational particle of class  $a$  and size  $i$  corresponds to  $w_i^a$  physical particles of class  $a$  and size  $i$ . The variable  $X_i(t)$  corresponds to the concentration of all physical particles of any class with mass  $i$ , and the fundamental constraint is that  $X_i(t)$  should somehow simulate the dynamical system (2.2). Of course,  $X_i(t)$  is a random quantity, and  $\hat{x}_i(t)$  is deterministic, so some care is needed here.

We can deduce from (3.6) that

$$\mathbb{E}[Q(t+h) - Q(t) \mid Q(t)] = h \sum \lambda_{ij}^{abc}(Q; \alpha, \beta, \gamma) \zeta_{ij}^{abc}(\alpha, \beta, \gamma) + o(h), \quad (3.10)$$

or

$$\mathbb{E}[Y(t+h) - Y(t) \mid Y(t)] = h \sum V^{-1} \lambda_{ij}^{abc}(VY; \alpha, \beta, \gamma) \zeta_{ij}^{abc}(\alpha, \beta, \gamma) + o(h), \quad (3.11)$$

which is just a time-discretization for a continuous-time Markov chain [5, 8, 34]. If it turns out that the quantity  $V^{-1}\lambda(VY; \dots)$  is well-defined, and the variance of  $Y(t+h) - Y(t)$  goes to zero fast enough as  $V \rightarrow \infty$ , then (3.11) limits to a deterministic ordinary differential equation (ODE) for  $y = \mathbb{E}[Y]$ , and one can ask whether or not this ODE corresponds to (2.2).

In fact, we show below in Theorem 1 that if we define

$$\begin{aligned} \rho_{ij}^{abc}(Y; \alpha, \beta, \gamma) &:= \lim_{V \rightarrow \infty} \frac{1}{V} \lambda_{ij}^{abc}(VY; \alpha, \beta, \gamma) \\ &= VT_{ij}^{abc} p_{ij}^{abc}(\alpha, \beta, \gamma) Y_i^a(t) (Y_j^b(t) - V^{-1} \delta_{i=j, a=b}), \end{aligned} \quad (3.12)$$

and we scale  $T_{ij}^{abc}$  so that this quantity is well-defined in the limit  $V \rightarrow \infty$ , then in this limit,  $Y(t)$  is close to the deterministic quantity  $y(t)$ , where

$$\frac{d}{dt} y(t) = \sum_{\substack{a,b,c \in \mathbb{M} \\ i,j \in \mathbb{N}_1 \\ \alpha, \beta, \gamma \in \{0,1\}}} \rho_{ij}^{abc}(y; \alpha, \beta, \gamma) \zeta_{ij}^{abc}(\alpha, \beta, \gamma). \quad (3.13)$$

(We specify precisely what we mean by “close” below in Theorem 1.) We want this system to correspond to that given in (2.2), so if we define the ensemble averages

$$\hat{y}_i^a(t) := \mathbb{E}[Y_i^a(t)], \quad \hat{x}_i^a(t) := \mathbb{E}[X_i^a(t)], \quad \hat{x}_i(t) := \mathbb{E}[X_i(t)], \quad (3.14)$$

then (2.2) corresponds to a deterministic ODE for  $\hat{x}_i(t)$ , and then, ultimately,  $\hat{y}_i^a(t)$ . We want these equations to match, and thus we define:

**Definition 1** (Consistency). Specifying  $T_{ij}^{abc}$  and  $p_{ij}^{abc}(\alpha, \beta, \gamma)$  determines (3.13). Similarly, (2.2) gives an ODE for  $\hat{x}_i(t)$  and thus gives evolution equations for  $\hat{y}_i^a(t)$  (q.v. (4.3) below). If these choices give us the same ODE, we say they are consistent.

It turns out that to check whether or not a choice of method is consistent *a priori* is a rather complicated computation. We perform this computation in the general case in Section 4 below, but show that a simplifying Ansatz allows us to develop a much simpler set of conditions, given in (4.13).

One of the two main theoretical results of this paper is the following theorem.

**Theorem 1.** *Assume that  $K_{ij}$  has compact support, i.e., there exists an  $N > 0$  such that  $K_{ij} = 0$  whenever  $i > N$  or  $j > N$ . Assume that  $\tilde{T}_{ij}^{abc}$  and  $\tilde{p}_{ij}^{abc}(\alpha, \beta, \gamma)$  are chosen to satisfy the consistency conditions defined in (4.13). Let  $Q$  be a solution to the CTMC whose rates and transitions are given by (3.4, 3.12) and define  $Y = Q/V$  and  $X$  as in (3.9). Let  $\hat{x}$  solve (2.2). Then  $X$  converges to  $\hat{x}$  in the following sense: for any  $T > 0, h > 0$ , there exist  $C_1, C_2 \in (0, \infty)$  such that*

$$\mathbb{P} \left( \sup_{t \in [0, T]} |X(t) - \hat{x}(t)| > h \right) \leq C_1 \exp(-C_2 V).$$

**Remark 1.** The constraint that  $K_{ij}$  have compact support seems restrictive, especially since many realistic kernels are typically formally written as polynomials, examples including the linear or multiplicative kernels,  $K_{ij} = i + j, K_{ij} = ij$ . However, this restriction is not significant in practice, as we prove below. Since there is a conserved quantity (the total mass) in this system, it is not possible to ever have

a particle larger than this mass in (2.2). Any stochastic simulation of this equation that gives a good approximation is unlikely to ever generate such a particle, and if it does not then truncation makes no difference. (We make this statement precise in Theorem 2 below.)

**4. Computations of consistency and optimality.** In this section, we first compute explicitly the consistency conditions on  $T_{ij}^{abc}$  and  $p_{ij}^{abc}(\alpha, \beta, \gamma)$ , i.e., the conditions on these functions which make (3.13) as a function of  $y$  evolve the same as (2.2) as a function of  $\hat{x}$ .

**4.1. Consistency.** Combining the definitions from (3.14, 3.7, 3.8, 3.9), we obtain

$$\hat{x}_i(t) = \sum_{a \in \mathbb{M}} \hat{y}_i^a(t) w_i^a. \quad (4.1)$$

Substituting this into (2.2), we obtain

$$\begin{aligned} \frac{d}{dt} \left( \sum_{d \in \mathbb{M}} \hat{y}_k^d(t) w_k^d \right) &= \sum_{i=1}^{k-1} \frac{1}{2} K_{i,k-i} \left( \sum_{a \in \mathbb{M}} \hat{y}_i^a(t) w_i^a \right) \left( \sum_{b \in \mathbb{M}} \hat{y}_{k-i}^b(t) w_{k-i}^b \right) \\ &\quad - \sum_{i=1}^{\infty} K_{ki} \left( \sum_{a \in \mathbb{M}} \hat{y}_i^a(t) w_i^a \right) \left( \sum_{d \in \mathbb{M}} \hat{y}_k^d(t) w_k^d \right), \end{aligned} \quad (4.2)$$

or

$$\begin{aligned} \sum_{d \in \mathbb{M}} w_k^d \frac{d}{dt} \hat{y}_k^d(t) &= \sum_{a,b \in \mathbb{M}} \sum_{i=1}^{k-1} \frac{1}{2} K_{i,k-i} \hat{y}_i^a(t) \hat{y}_{k-i}^b(t) w_i^a w_{k-i}^b \\ &\quad - \sum_{a,d \in \mathbb{M}} \sum_{i=1}^{\infty} K_{ki} \hat{y}_i^a(t) \hat{y}_k^d(t) w_i^a w_k^d. \end{aligned} \quad (4.3)$$

We now expand (3.13) using (3.1, 3.4, 3.5, 3.12).

$$\begin{aligned} \frac{d}{dt} y(t) &= \sum T_{ij}^{abc} p_{ij}^{abc}(\alpha, \beta, \gamma) y_i^a(t) y_j^b(t) ((1-\alpha)e_i^a + (1-\beta)e_j^b + \gamma e_{i+j}^c) \quad (4.4) \\ &= - \sum_{\substack{a,b,c \in \mathbb{M} \\ i,j \in \mathbb{N}_1 \\ \beta, \gamma \in \{0,1\}}} T_{ij}^{abc} p_{ij}^{abc}(0, \beta, \gamma) y_i^a(t) y_j^b(t) e_i^a \\ &\quad - \sum_{\substack{a,b,c \in \mathbb{M} \\ i,j \in \mathbb{N}_1 \\ \alpha, \gamma \in \{0,1\}}} T_{ij}^{abc} p_{ij}^{abc}(\alpha, 0, \gamma) y_i^a(t) y_j^b(t) e_j^b \\ &\quad + \sum_{\substack{a,b,c \in \mathbb{M} \\ i,j \in \mathbb{N}_1 \\ \alpha, \beta \in \{0,1\}}} T_{ij}^{abc} p_{ij}^{abc}(\alpha, \beta, 1) y_i^a(t) y_j^b(t) e_{i+j}^c. \end{aligned} \quad (4.5)$$

Writing out the specific coordinates of  $y(t)$ , we have

$$\frac{d}{dt} y_k^d(t) = - \sum_{\substack{a,b,c \in \mathbb{M} \\ i,j \in \mathbb{N}_1 \\ \beta, \gamma \in \{0,1\}}} T_{ij}^{abc} p_{ij}^{abc}(0, \beta, \gamma) y_i^a(t) y_j^b(t) \delta_{a=d, k=i}$$



$$\begin{aligned}
& - \sum_{\substack{a,b,c \in \mathbb{M} \\ i,j \in \mathbb{N}_1 \\ \alpha, \gamma \in \{0,1\}}} T_{ij}^{abc} p_{ij}^{abc}(\alpha, 0, \gamma) y_i^a(t) y_j^b(t) \delta_{b=d, k=j} \\
& + \sum_{\substack{a,b,c \in \mathbb{M} \\ i,j \in \mathbb{N}_1 \\ \alpha, \beta \in \{0,1\}}} T_{ij}^{abc} p_{ij}^{abc}(\alpha, \beta, 1) y_i^a(t) y_j^b(t) \delta_{c=d, k=i+j} \tag{4.6}
\end{aligned}$$

$$\begin{aligned}
& = - \sum_{\substack{b,c \in \mathbb{M} \\ j \in \mathbb{N}_1 \\ \beta, \gamma \in \{0,1\}}} T_{kj}^{dbc} p_{kj}^{dbc}(0, \beta, \gamma) y_k^d(t) y_j^b(t) \\
& - \sum_{\substack{a,c \in \mathbb{M} \\ i \in \mathbb{N}_1 \\ \alpha, \gamma \in \{0,1\}}} T_{ik}^{adc} p_{ik}^{adc}(\alpha, 0, \gamma) y_i^a(t) y_k^d(t) \\
& + \sum_{\substack{a,b \in \mathbb{M} \\ i=1, \dots, k-1 \\ \alpha, \beta \in \{0,1\}}} T_{i,k-i}^{abd} p_{i,k-i}^{abd}(\alpha, \beta, 1) y_i^a(t) y_{k-i}^b(t). \tag{4.7}
\end{aligned}$$

Making the change of dummy variables in the first sum of  $b \mapsto a, j \mapsto i, \beta \mapsto \alpha$ , and switching these same variables in the third sum, we obtain

$$\begin{aligned}
\frac{d}{dt} y_k^d(t) & = - \sum_{\substack{a,c \in \mathbb{M} \\ i \in \mathbb{N}_1 \\ \alpha, \gamma \in \{0,1\}}} T_{ki}^{dac} p_{ki}^{dac}(0, \alpha, \gamma) y_k^d(t) y_i^a(t) \\
& - \sum_{\substack{a,c \in \mathbb{M} \\ i \in \mathbb{N}_1 \\ \alpha, \gamma \in \{0,1\}}} T_{ik}^{adc} p_{ik}^{adc}(\alpha, 0, \gamma) y_i^a(t) y_k^d(t) \\
& + \frac{1}{2} \sum_{\substack{a,b \in \mathbb{M} \\ i=1, \dots, k-1 \\ \alpha, \beta \in \{0,1\}}} T_{i,k-i}^{abd} p_{i,k-i}^{abd}(\alpha, \beta, 1) y_i^a(t) y_{k-i}^b(t) \\
& + \frac{1}{2} \sum_{\substack{a,b \in \mathbb{M} \\ i=1, \dots, k-1 \\ \alpha, \beta \in \{0,1\}}} T_{i,k-i}^{bad} p_{i,k-i}^{bad}(\beta, \alpha, 1) y_i^b(t) y_{k-i}^a(t). \tag{4.8}
\end{aligned}$$

If we symmetrize

$$\tilde{T}_{ij}^{abc} := T_{ij}^{abc} + T_{ji}^{bac}, \tag{4.9}$$

$$\tilde{p}_{ij}^{abc}(\alpha, \beta, \gamma) := \frac{T_{ij}^{abc} p_{ij}^{abc}(\alpha, \beta, \gamma) + T_{ij}^{bac} p_{ij}^{bac}(\beta, \alpha, \gamma)}{\tilde{T}_{ij}^{abc}}, \tag{4.10}$$

then we can combine two terms and obtain

$$\frac{d}{dt} y_k^d(t) = - \sum_{\substack{a,c \in \mathbb{M} \\ i \in \mathbb{N}_1 \\ \alpha, \gamma \in \{0,1\}}} \tilde{T}_{ik}^{adc} \tilde{p}_{ik}^{adc}(\alpha, 0, \gamma) y_k^d(t) y_i^a(t)$$

$$+ \frac{1}{2} \sum_{\substack{a,b \in \mathbb{M} \\ i=1, \dots, k-1 \\ \alpha, \beta \in \{0,1\}}} \tilde{T}_{i,k-i}^{abd} \tilde{p}_{i,k-i}^{abd}(\alpha, \beta, 1) y_i^a(t) y_{k-i}^b(t). \quad (4.11)$$

We want to compare (4.11) and (4.3). Summing (4.11) over  $d$  with the appropriate weighting gives

$$\begin{aligned} \sum_{d \in M} w_k^d \frac{d}{dt} y_k^d(t) = & - \sum_{\substack{a,c,d \in \mathbb{M} \\ i \in \mathbb{N}_1 \\ \alpha, \gamma \in \{0,1\}}} \tilde{T}_{ik}^{adc} \tilde{p}_{ik}^{adc}(\alpha, 0, \gamma) w_k^d y_k^d(t) y_i^a(t) \\ & + \frac{1}{2} \sum_{\substack{a,b,d \in \mathbb{M} \\ i=1, \dots, k-1 \\ \alpha, \beta \in \{0,1\}}} \tilde{T}_{i,k-i}^{abd} \tilde{p}_{i,k-i}^{abd}(\alpha, \beta, 1) w_k^d y_i^a(t) y_{k-i}^b(t). \end{aligned} \quad (4.12)$$

Thus for consistency we must choose

$$\sum_{d \in \mathbb{M}, \alpha, \beta \in \{0,1\}} \tilde{T}_{i,k-i}^{abd} \tilde{p}_{i,k-i}^{abd}(\alpha, \beta, 1) w_k^d = K_{i,k-i} w_i^a w_{k-i}^b, \quad (4.13a)$$

$$\sum_{c \in \mathbb{M}, \alpha, \gamma \in \{0,1\}} \tilde{T}_{ik}^{adc} \tilde{p}_{ik}^{adc}(\alpha, 0, \gamma) w_k^d = K_{ki} w_i^a w_k^d. \quad (4.13b)$$

We call these the consistency conditions on  $T, p$  as per Definition 1. These are still quite complicated as written, so we will make a simplifying Ansatz in the next section.

**4.2. A simplifying Ansatz.** We will now make a simplifying assumption on the probability distribution. As stated above, the probability distribution  $p_{ij}^{abc}(\alpha, \beta, \gamma)$  has no constraints other than (3.3).

However, we now assume that the creation or destruction of each of the three particles involved in a coagulation are chosen independently, i.e., in a coagulation involving  $(a, i)$  and  $(b, j)$  coagulating into  $(c, i + j)$ , we choose the three events “whether  $(a, i)$  is destroyed”, “whether  $(b, j)$  is destroyed”, and “whether  $(c, i + j)$  is created” independently.

More formally, we define numbers

$$0 \leq p_{\text{des},ij}^{abc} \leq 1, \quad 0 \leq p_{\text{cre},ij}^{abc} \leq 1 \quad (4.14)$$

and

$$p_{\text{des},ij}^{abc}(\alpha) := \begin{cases} p_{\text{des},ij}^{abc} & \text{if } \alpha = 0, \\ 1 - p_{\text{des},ij}^{abc} & \text{if } \alpha = 1, \end{cases} \quad (4.15)$$

$$p_{\text{cre},ij}^{abc}(\gamma) := \begin{cases} p_{\text{cre},ij}^{abc} & \text{if } \gamma = 1, \\ 1 - p_{\text{cre},ij}^{abc} & \text{if } \gamma = 0, \end{cases} \quad (4.16)$$

and we assume that

$$\tilde{p}_{ij}^{abc}(\alpha, \beta, \gamma) = p_{\text{des},ij}^{abc}(\alpha) p_{\text{des},ji}^{bac}(\beta) p_{\text{cre},ij}^{abc}(\gamma). \quad (4.17)$$

We see here that if we interpret  $p_{\text{des},ij}^{abc}$  as the probability of destroying particle  $(\alpha, i)$  during a coagulation event, and  $p_{\text{cre},ij}^{abc}$  as the probability of creating particle  $(c, i + j)$  during this event, then (4.17) is simply asserting that these creations and destructions are done independently.

Observe that, by definition,

$$\sum_{\alpha \in \{0,1\}} p_{\text{des},ij}^{abc}(\alpha) = \sum_{\gamma \in \{0,1\}} p_{\text{cre},ij}^{abc}(\gamma) = 1 \text{ for all } a, b, c, i, j. \quad (4.18)$$

Writing out (4.13) under this Ansatz, we obtain

$$\begin{aligned} K_{i,k-i} w_i^a w_{k-i}^b &= \sum_{c \in \mathbb{M}} \tilde{T}_{i,k-i}^{abc} p_{\text{cre},i,k-i}^{abc} w_k^c, \\ K_{ki} w_i^a &= \sum_{c \in \mathbb{M}} \tilde{T}_{ik}^{abc} p_{\text{des},ki}^{bac}. \end{aligned} \quad (4.19)$$

It is more convenient to write these formulas if we switch  $(a, i)$  and  $(b, j)$  in the second equation and relabel  $j - i \mapsto j$  in the first equation, obtaining:

$$K_{ij} w_i^a w_j^b = \sum_{c \in \mathbb{M}} \tilde{T}_{ij}^{abc} p_{\text{cre},ij}^{abc} w_{i+j}^c, \quad (4.20)$$

$$K_{ij} w_j^b = \sum_{c \in \mathbb{M}} \tilde{T}_{ij}^{abc} p_{\text{des},ij}^{abc}. \quad (4.21)$$

**Remark 1.** In the case of just one type, where  $\mathbb{M} = \{1\}$ , there is no sum on the RHS. This reduces to

$$K_{ij} w_i w_j = \tilde{T}_{ij} p_{\text{cre},ij} w_{i+j}, \quad (4.22)$$

$$K_{ij} w_j = \tilde{T}_{ij} p_{\text{des},ij}, \quad (4.23)$$

which is consistent with our earlier results in [7] (note the argument order for  $p_{\text{death}}$  is reversed here).

**4.3. Optimality.** Even under the independence Ansatz, there are still many choices we can make for  $T, p$ . In this section we discuss optimality, in the sense of having the least number of events generated by the method.

Since they are probabilities, we require

$$0 \leq p_{\text{cre},ij}^{abc} \leq 1, \quad 0 \leq p_{\text{des},ij}^{abc} \leq 1, \quad (4.24)$$

and this implies that we can ensure the consistency conditions if and only if

$$\sum_{c \in \mathbb{M}} w_{i+j}^c \tilde{T}_{ij}^{abc} \geq K_{ij} w_i^a w_j^b, \quad (4.25)$$

$$\sum_{c \in \mathbb{M}} \tilde{T}_{ij}^{abc} \geq K_{ij} w_j^b, \quad (4.26)$$

$$\sum_{c \in \mathbb{M}} \tilde{T}_{ij}^{abc} \geq K_{ij} w_i^a, \quad (4.27)$$

where we use symmetry of  $\tilde{T}$ .

For optimality we want to solve

$$\min_{\tilde{T}} \sum_{c \in \mathbb{M}} \tilde{T}_{ij}^{abc} \quad (4.28)$$

subject to the above constraints and

$$\tilde{T}_{ij}^{abc} \geq 0. \quad (4.29)$$

In general there will be many optimal solutions. One solution that is always optimal is to find

$$c_{i+j}^* = \operatorname{argmax}_{c \in \mathbb{M}} w_{i+j}^c \quad (4.30)$$

and then set

$$\tilde{T}_{ij}^{abc} = \begin{cases} K_{ij} \max\left(\frac{w_i^a w_j^b}{w_{i+j}^c}, w_i^a, w_j^b\right) & \text{if } c = c_{i+j}^*, \\ 0 & \text{otherwise.} \end{cases} \quad (4.31)$$

This expression is consistent for any choice of  $c_{i+j}^*$ . It is also optimal under the additional constraint that  $\tilde{T}_{ij}^{abc} = 0$  for  $c \neq c_{i+j}^*$ .

One nice practical possibility would be to require that  $c^*$  is either  $a$  or  $b$  (as these are the two groups involved in the coagulation event already), and to pick  $c^*$  to be the group with the lowest weight, thereby minimizing the statistical error.

Note that we can compute the creation and destruction probabilities by using the fact that  $\tilde{T}_{ij}^{abc} = 0$  for  $c \neq c^*$ , so

$$K_{ij} w_i^a w_j^b = w_{i+j}^{c^*} \tilde{T}_{ij}^{abc^*} p_{\text{cre},ij}^{abc^*}, \quad (4.32)$$

$$K_{ij} w_j^b = \tilde{T}_{ij}^{abc^*} p_{\text{des},ij}^{abc^*}, \quad (4.33)$$

giving

$$p_{\text{cre},ij}^{abc^*} = \frac{K_{ij} w_i^a w_j^b}{w_{i+j}^{c^*} \tilde{T}_{ij}^{abc^*}}, \quad (4.34)$$

$$p_{\text{des},ij}^{abc^*} = \frac{K_{ij} w_j^b}{\tilde{T}_{ij}^{abc^*}}. \quad (4.35)$$

If we define

$$M_{ij}^{abc} := \min\left(w_i^a, w_j^b, w_{i+j}^c\right) \quad (4.36)$$

then we see that  $\tilde{T}$  can be written

$$\tilde{T}_{ij}^{abc} = \delta_{c,c^*} \frac{K_{ij} w_i^a w_j^b}{M_{ij}^{abc}} := \begin{cases} \frac{K_{ij} w_i^a w_j^b}{M_{ij}^{abc}} & \text{if } c = c^*, \\ 0 & \text{otherwise.} \end{cases} \quad (4.37)$$

and

$$p_{\text{cre},ij}^{abc^*} = \frac{M_{ij}^{abc^*}}{w_{i+j}^{c^*}}, \quad p_{\text{des},ij}^{abc^*} = \frac{M_{ij}^{abc^*}}{w_i^a}. \quad (4.38)$$

Note again that the argument order convention for  $p_{\text{des}}$  is reversed from [7]. These optimal probabilities will be used in all numerical experiments in Section 6.

**5. Theorem and proof.** We will prove Theorem 1, which we restate here for completeness.

**Theorem 1.** *Assume that  $K_{ij}$  has compact support, i.e., there exists an  $N > 0$  such that  $K_{ij} = 0$  for  $i > N$  or  $j > N$ . Assume that  $\tilde{T}_{ij}^{abc}$  and  $\tilde{p}_{ij}^{abc}(\alpha, \beta, \gamma)$  are chosen to satisfy the consistency conditions defined in (4.13). Let  $Q$  be a solution to the CTMC whose rates and transitions are given by (3.4, 3.12) and define  $Y = Q/V$*

and  $X$  as in (3.9). Let  $\hat{x}$  solve (2.2). Then  $X$  converges to  $\hat{x}$  in the following sense: for any  $T > 0, h > 0$ , there exist  $C_1, C_2 \in (0, \infty)$  such that

$$\mathbb{P} \left( \sup_{t \in [0, T]} |X(t) - \hat{x}(t)| > h \right) \leq C_1 \exp(-C_2 V).$$

**Remark 2.** Before we prove the theorem we give some motivation for the convergence in terms of generators. The generator of the stochastic process [34, 42] is

$$\mathcal{L}f(q) = \sum_{\substack{a, b, c \in \mathbb{M} \\ i, j \in \mathbb{N}_1 \\ \alpha, \beta, \gamma \in \{0, 1\}}} \lambda_{ij}^{abc}(q; \alpha, \beta, \gamma) [f(q + \zeta_{ij}^{abc}(\alpha, \beta, \gamma)) - f(q)]. \quad (5.1)$$

We will need to write this triple sum in many places below; whenever we write a sum without limits in the remainder of this section, we mean the triple sum which appears in (5.1). If we define

$$\rho_{ij}^{abc}(y; \alpha, \beta, \gamma) := \lim_{V \rightarrow \infty} \frac{1}{V} \lambda_{ij}^{abc}(Vy; \alpha, \beta, \gamma) \quad (5.2)$$

$$= VT_{ij}^{abc} p_{ij}^{abc}(\alpha, \beta, \gamma) y_i^a(t) (y_j^b(t) - V^{-1} \delta_{i=j, a=b}), \quad (5.3)$$

and  $\tilde{f}(y) := f(Vy)$ , then we have a new operator

$$\mathcal{L}_V \tilde{f}(y) = \sum V \rho_{ij}^{abc}(y; \alpha, \beta, \gamma) [\tilde{f}(q + V^{-1} \zeta_{ij}^{abc}(\alpha, \beta, \gamma)) - \tilde{f}(q)]. \quad (5.4)$$

Computing the first nonzero term in the Taylor series expansion on the right-hand side gives

$$\mathcal{L}_V \tilde{f}(y) \approx \sum \rho_{ij}^{abc}(y; \alpha, \beta, \gamma) \nabla \tilde{f}(q) \cdot \zeta_{ij}^{abc}(\alpha, \beta, \gamma) \quad (5.5)$$

$$= \nabla \tilde{f}(q) \cdot \sum \rho_{ij}^{abc}(y; \alpha, \beta, \gamma) \zeta_{ij}^{abc}(\alpha, \beta, \gamma). \quad (5.6)$$

This is the Liouville equation for the ODE

$$\frac{d}{dt} y(t) = \sum_{\substack{a, b, c \in \mathbb{M} \\ i, j \in \mathbb{N}_1 \\ \alpha, \beta, \gamma \in \{0, 1\}}} \rho_{ij}^{abc}(y; \alpha, \beta, \gamma) \zeta_{ij}^{abc}(\alpha, \beta, \gamma). \quad (5.7)$$

*Proof of Theorem 1.* The main result that we will use in the proof is Kurtz's theorem [31], [42, Theorem 5.3]. In fact, the statement of Theorem 1 is precisely the same as [42, Theorem 5.3] with the added assumption that the rescaled rates  $\rho_{ij}^{abc}(y; \alpha, \beta, \gamma)$  be Lipschitz and uniformly bounded as functions of  $y_i^a(t)$ , and that the vectors  $y_i^a(t)$  live in a finite-dimensional space.

The functions  $\rho_{ij}^{abc}(y; \alpha, \beta, \gamma)$  are locally Lipschitz, being polynomial. Moreover, under the assumption of compact support for  $K_{ij}$ , they become globally Lipschitz. Similarly, since  $K_{ij} = 0$  if  $i, j \geq N$ , we can also consider the finite-dimensional truncation of (2.2) without loss of generality, and note that this makes all of the sets of indices that we sum over above finite.  $\square$

For the theorem to be true, we need a kernel of compact support. Assume now that we have a coagulation kernel  $K_{ij}$  that has unbounded support. For any  $N > 0$ , we can consider the truncation of this kernel in the obvious way, by choosing

$$K_{ij}^{(N)} := \begin{cases} K_{ij} & \text{if } i, j < N, \\ 0 & \text{otherwise.} \end{cases} \quad (5.8)$$

The obvious question to ask is, if we choose two different truncations,  $K_{ij}^{(N)}$  and  $K_{ij}^{(N')}$ , what is the probability that we get a different answer in the simulation scheme described in this paper?

First, define  $M_t := \sum_{i \in \mathbb{N}_1, a \in \mathbb{M}} iy_i^a(t)$ , the total mass of the system at time  $t$ . Then notice that if  $|M_t| < \widetilde{M}$ , then  $y_i^a(t) = 0$  for  $i > \widetilde{M}$ . Moreover,

$$\sup_{y: \sum_{i \in \mathbb{N}_1, a \in \mathbb{M}} iy_i^a(t) < \widetilde{M}} \rho_{ij}^{abc}(y; \alpha, \beta, \gamma) < C\widetilde{M}^2,$$

so on the set  $y_i^a(t) < \widetilde{M}$ , we have that  $y$  has finitely-many non-zero components and  $\rho$  is uniformly bounded and Lipschitz, as discussed in the proof above. Moreover, notice that  $M_t$  is a martingale, i.e., that if  $t' > t$  then  $\mathbb{E}[M_{t'} | M_t] = M_t$ .

Then we have the following theorem.

**Theorem 2.** *Let  $N, N' > 2M_0$  and consider two truncated numerical methods, one with truncation parameter  $N$  and the other with truncation parameter  $N'$ . More concretely, consider any numerical methods given by (4.13), but where we replace  $K_{ij}$  with  $K_{ij}^{(N)}$  and  $K_{ij}^{(N')}$ ; call these solutions  $X_t^{(N)}$  and  $X_t^{(N')}$ , respectively. Then for any set  $A$  and  $T > 0$ , there exist  $C_1, C_2 > 0$  such that*

$$\sup_{t \in [0, T]} \left| \mathbb{P}(X_t^{(N)} \in A) - \mathbb{P}(X_t^{(N')} \in A) \right| \leq C_1 e^{-C_2 V}.$$

*Proof.* The first observation to be made here is that as long as  $M_t < \min(N, N')$ , then all of the statements in the theorem follow, because then the truncated kernels and thus the evolution (3.6) would be identical. So, in short, if we show that the probability that  $M_t > 2M_0$  is exponentially small, then the theorem follows.

Thus we consider either truncated process (we will choose  $X_t^{(N)}$  for concreteness) and compute the probability of obtaining a realization with  $M_t > 2M_0$ . Denote the probability space of all outcomes of the stochastic process as  $\Omega$ . Define  $\Omega_2 := \{\omega \in \Omega : M(X_t) > 2M_0\}$ , i.e., all outcomes in the probability space where the stochastic trajectory has twice the mass than it did at time zero. Notice that  $\mathbb{P}(\Omega_2) \geq \mathbb{P}(|M(X_t^{(N)}) - M(\widehat{x}(t))| > M_0)$ . Then by Theorem 1, for any  $h, T > 0$ , there exist  $C_3, C_4 > 0$  such that

$$\mathbb{P} \left( \sup_{t \in [0, T]} |X_t^{(N)} - \widehat{x}(t)| > h \mid \Omega_2^c \right) \leq C_3 e^{-C_4 V},$$

and, by the same theorem, there exist  $C_5, C_6 > 0$  such that  $\mathbb{P}(\Omega_2) \leq C_5 e^{-C_6 V}$ . Then there are  $C_1, C_2 > 0$  with

$$\begin{aligned} \mathbb{P} \left( \sup_{t \in [0, T]} |X_t^{(N)} - \widehat{x}(t)| > h \right) &= \mathbb{P} \left( \sup_{t \in [0, T]} |X_t^{(N)} - \widehat{x}(t)| > h \mid \Omega_2 \right) \mathbb{P}(\Omega_2) \\ &\quad + \mathbb{P} \left( \sup_{t \in [0, T]} |X_t^{(N)} - \widehat{x}(t)| > h \mid \Omega_2^c \right) \mathbb{P}(\Omega_2^c) \\ &\leq 1 \cdot C_5 e^{-C_6 V} + (C_3 e^{-C_4 V})(1 - C_5 e^{-C_6 V}) \\ &\leq C_1 e^{-C_2 V}. \end{aligned}$$

The theorem then follows by the triangle inequality.  $\square$

**6. Practical applications of method.** Here we present a series of exhibitions of the practical effectiveness of the methods developed in this paper. In Section 6.1 we give an example of compositional multiscaling, in Section 6.2 we exhibit the usefulness of size-dependent weightings, and in Section 6.3 we give an example of both at once.

**6.1. Independently weighted sub-populations.** In this section we address the “compositional multiscaling problem”, as introduced in Section 1. We use the formulation of Section 3 to weight particles in the rare sub-population differently to the rest of the particles. This allows more accurate statistics to be obtained for the sub-population while keeping the total particle number low and minimizing the computational cost.

Here we present this application in the context of two sub-populations: common sub-population 1 and rare sub-population 2. The extension to multiple sub-populations with varying frequency is straightforward. Take  $w_i^a$  to be independent of size  $i$ , so we have all particles in sub-population 1 weighted with  $w^1$  and all particles in sub-population 2 weighted with  $w^2$ . For coagulation events we will take the destination group  $c^*$  to be whichever of the coagulating groups  $a$  or  $b$  has the lowest weight, as described in Section 4.3. That is,

$$c_{i+j}^* = \begin{cases} 1 & \text{if } i = j = 1, \\ 2 & \text{otherwise,} \end{cases} \quad (6.1)$$

and we use the optimal expressions (4.37) and (4.38).

For a numerical test of coagulating independently weighted sub-populations, we use the Fuchs form of the Brownian coagulation kernel [26],

$$K_{i,j}^B = \frac{4\pi (r_i + r_j) (D_{p,i} + D_{p,j})}{\frac{r_i + r_j}{r_i + r_j + \sqrt{\delta_i^2 + \delta_j^2}} + \frac{4(D_{p,i} + D_{p,j})}{\sqrt{\bar{v}_{p,i}^2 + \bar{v}_{p,j}^2} (r_i + r_j)}}, \quad (6.2)$$

where  $r_i$  and  $r_j$  are the radii of the coagulating particles,  $D_{p,i}$  is the diffusion coefficient of particle  $i$  in air,  $\delta_i$  is the correction to account for particle  $i$  in the transition regime (see Eq. 15.34 in [26]), and  $\bar{v}_{p,i}$  is the thermal velocity of particle  $i$  in air. The exact expressions and constant values are specified in the file `coag_kernel_brown.F90` and its dependencies in the PartMC code [45].

Each sub-population begins at time  $t = 0$  with a log-normal distribution with geometric mean diameter  $D_g = 50$  nm and geometric standard deviation  $\log_{10} \sigma_g = 0.24$ . Sub-population 1 (blue in Figure 1) has an initial physical number concentration of  $N^1 = 10^5$  cm $^{-3}$  while sub-population 2 (red in Figure 1) has an initial physical number concentration of  $N^2 = 10^2$  cm $^{-3}$ . We assigned a density of 1000 kg m $^{-3}$  and a molecular weight of 18 g mol $^{-1}$  to the particles (pure water). The temperature was set to 288 K and the pressure was 10 $^5$  Pa throughout the simulation.

Sub-population 1 particles have weight  $w^1$  and sub-population 2 particles have weight  $w^2$ . The resolution of each particle sub-population is related by the ratio

$$r = \frac{w^1}{w^2}. \quad (6.3)$$

Two special cases are  $r = 1$ , corresponding to equal weights for all particles (the equal weight case in Figures 1 and 2), and  $r = N^1/N^2 = 10^3$ , corresponding to equal numbers of computational particles used to initially represent each distribution on average (the equal number case in Figures 1 and 2).

The initial mean numbers of computational particles in each sub-population, denoted  $N^{p1}$  and  $N^{p2}$ , sum to the total number of computational particles,  $N^{p1} + N^{p2} = N_p$ , and have weights corresponding to the physical number concentrations, so  $w_1 N^{p1} = N^1$  and  $w_2 N^{p2} = N^2$ . Using (6.3) gives

$$N^{p1} = \frac{N^1/N^2}{r + N^1/N^2} N_p, \quad (6.4)$$

$$N^{p2} = \frac{r}{r + N^1/N^2} N_p. \quad (6.5)$$

In the case simulated here the physical concentration ratio is  $N^1/N^2 = 10^3$ , so when  $r = 1$  we have  $N^{p1} \approx N_p$  and  $N^{p2} \approx N_p/10^3$ , meaning that almost all computational particles are in the common sub-population 1 and almost none are in the rare sub-population 2. Correspondingly, when  $r = N^1/N^2 = 10^3$  we have  $N^{p1} = N^{p2} = N_p/2$ , so the computational particles are evenly shared between the two sub-populations.

Figure 1 shows the number size distributions after  $t = 24$  h of simulation for the equal weight and equal number cases, compared to very accurate finite-volume simulations. The error bars show the 95% spread for the particle simulations. The finite-volume simulations use a two-population scheme, with one set of finite-volume bins tracking each sub-population. Each combination of sub-populations coagulate using a high-resolution finite-volume scheme [3], with the coagulation products from 1–1 and 2–2 coagulations remaining in the respective sub-population, and 1–2 cross-coagulations going to the rare sub-population 2 following (6.1).

As we might expect, in Figure 1 the equal weights method substantially under-resolves sub-population 2, while an equal-number weighting dramatically improves the resolution of sub-population 2 while only slightly degrading the sub-population 1 resolution.

The trade-off between sub-population 1 and sub-population 2 resolution as the ratio  $r$  is varied is quantified in Figure 2. Here ratios ranging from  $r = 1$  to  $r = 10^6$  are used for computational particle number varying from  $N_p = 10^3$  to  $N_p = 10^5$ . The special cases of equal weight ( $r = 1$ ) and equal number ( $r = 10^3$ ) are indicated by circles and squares, respectively.

For ratios between  $r = 1$  and  $r = 10^3$  we see a trade-off between sub-population 1 expected error and sub-population 2 expected error, with substantial improvements in sub-population 2 error accompanied by very small degradations in sub-population 1 error. Equal-number weighting gives a reduction in sub-population 2 error by about 17 times from equal weights, at the expense of an increase of about 1.4 times in sub-population 1 error.

For ratios above  $r = 10^3$ , the simulation has more computational particles resolving sub-population 2 than sub-population 1. This does not result in improved sub-population 2 error, however, because the dynamics of sub-population 2 are highly dependent on sub-population 1 for coagulation partners. We thus see degrading performance in both sub-populations as the ratio increases and more computational particles are used for sub-population 2.

As shown in Theorem 1, we expect that the particle solutions will converge to the mean-field solutions generated by the very accurate finite-volume method as the number of particles increases,  $N_p \rightarrow \infty$ . The particle error versus  $N_p$  is shown in Figure 3, and we see that the method is indeed converging as  $N_p$  increases, as predicted, with rate  $N_p^{-1/2}$ .



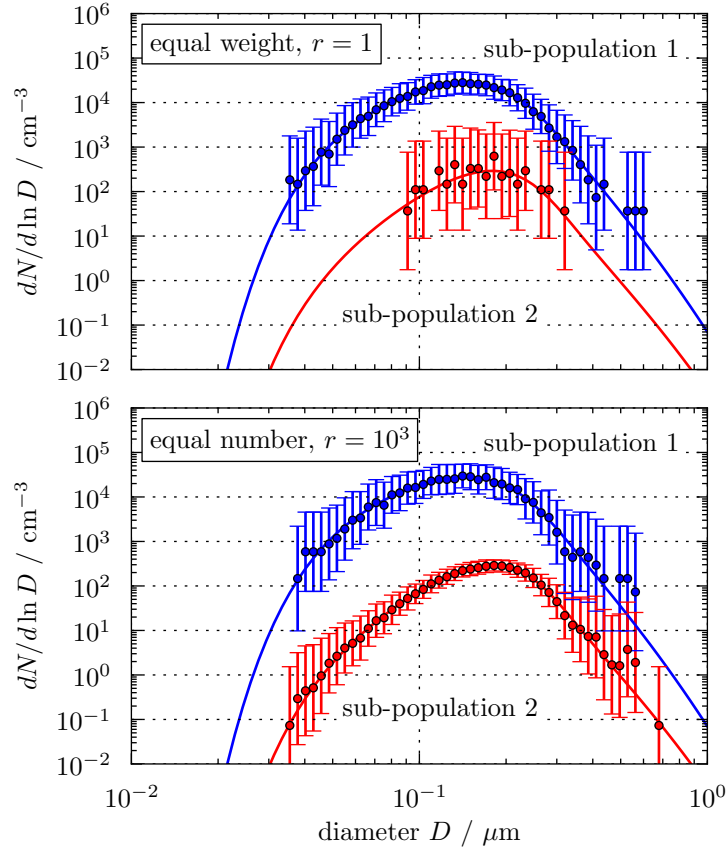


FIGURE 1. Size distributions for two particle sub-populations using the scheme described in Section 6.1 with  $N_p = 10^3$  particles. Top: equal weightings for both distributions. Bottom: equal computational number for each distribution. The points are the mean of the particle process, while the error bars show 95% spread of realizations, not confidence intervals for the mean. The solid lines are very accurate finite-volume solutions. Using an equal-number rather than equal-weight weighting reduces sub-population 2 expected error by about 17 times, with an increase in sub-population 1 expected error of only about 1.4 times.

All particle simulations were implemented using the accelerated sampling scheme of Curtis et al. [4], so the computational cost is proportional to the number of coagulation events. In this subsection the number of coagulation events is almost independent of the weighting scheme chosen, because the two sub-populations have approximately equal size distributions and the coagulation kernel is only size-dependent, with no dependence on which sub-population contains the particle. Thus the computational cost is approximately constant for each value of  $N_p$  in Figure 2, independent of the weight ratio  $r$ .

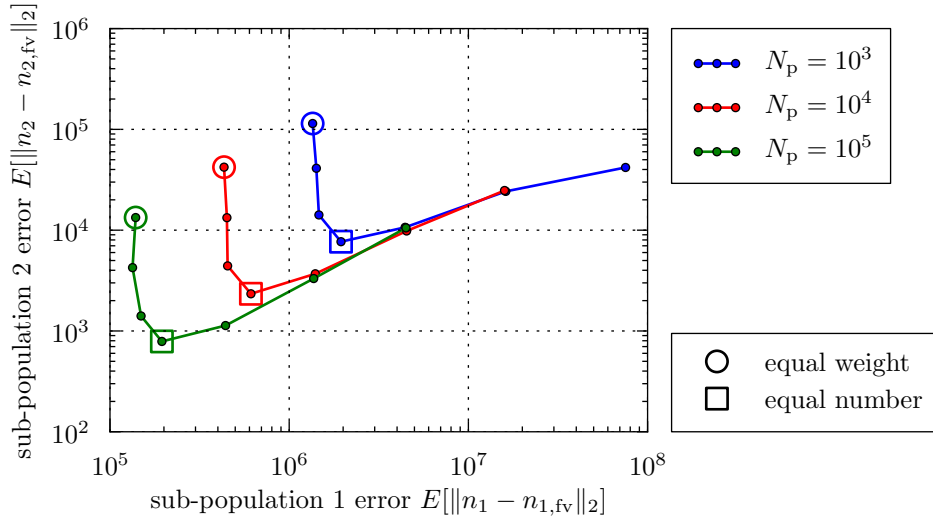


FIGURE 2. Errors for two particle sub-populations for varying number of particles  $N_p$ . The weight ratio  $r = w^1/w^2$  is 1, 10,  $10^2$ ,  $10^3$ ,  $10^4$ ,  $10^5$ , and  $10^6$ , moving from upper-left to down and to the right on each line. The circle points correspond to equal weights for both sub-populations (top panel in Figure 1), while the square points correspond to equal numbers of computational particles for each sub-population (bottom panel in Figure 1). Error bars are not shown as they are visually negligible.

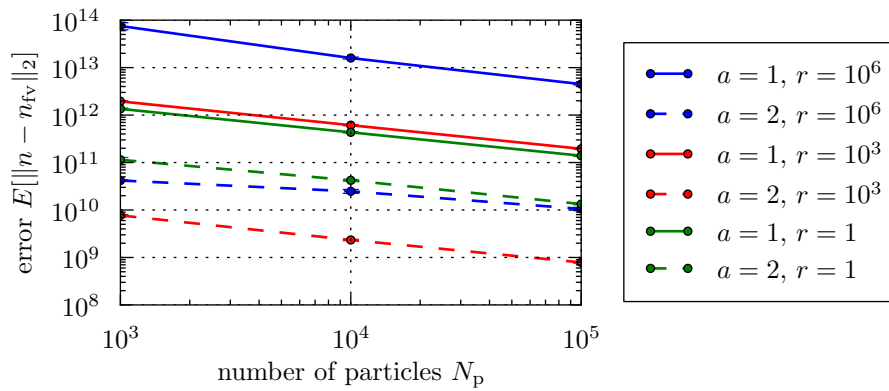


FIGURE 3. Convergence of the size distribution using the weighting scheme from Section 6.1 for different weight ratios  $r$  for the distribution of group  $a = 1$  particles and group  $a = 2$  particles. The baseline for computing the error is a very accurate finite-volume solution  $n_{fv}$ .

**6.2. Combined size-dependent weightings.** Another practical problem that often arises in particle simulations concerns the relative scarcity of large particles formed by many coagulation events from small precursors, namely the “size multi-scaling problem” discussed in Section 1. This occurs physically in raindrop formation, where approximately  $10^6$  small precursor droplets with diameter  $D \approx 10 \mu\text{m}$  must coagulate to form a single, comparatively rare, rain droplet with diameter  $D \approx 1 \text{ mm}$  [29].

While it would in principle be possible to classify small and large particles into different sub-populations and use the differential weighting scheme discussed in Section 6.1, in practice it is often difficult to predict in advance just where the transition between these two sub-populations should be chosen, and how they should be independently weighted. In addition, it is undesirable to introduce an artificial cutoff in the accuracy of the simulation as a function of particle size, such as would arise from using discrete sub-populations to classify and weight particles.

An alternative approach, as developed in the Weighted Flow Algorithm (WFA) [7, 28], weights particles with a smooth function of particle size. In DeVille et al. [7] it is shown that power-law weightings of the form  $w(D) \propto D^\alpha$  are efficient for many physical simulation problems, where  $D$  is the particle diameter. This is equivalent to taking  $w_i^a \propto i^{\alpha/3}$  in our notation.

While such exponential size-dependent weightings can be very efficient, they typically require *a priori* knowledge of the system evolution, so that the correct weighting function exponent can be selected. We can overcome this limitation and develop an *a posteriori* adaptive scheme by using the framework from Section 3. This extends the concept of composite weighting schemes [7, Section 6.7] to allow interactions between the component weighted particle sets.

For simplicity of exposition, we will only consider particles to be differentiated by size in this section, and we will discuss multiple sub-populations with size-dependent weightings in Section 6.3. We divide the particle population into two groups, so that each particle group can be doubled and halved independently to control total computational particle number, but the weight  $w_i^a$  will be independent of group number  $a$ .

To construct the weighting function, take precursor weight factors  $\nu_i^b$  for  $b \in \mathbb{M}$  and computational volumes  $V^b$ . Taking a total computational volume  $V$ , we compute the weights  $w_i^a$  by the harmonic mean

$$w_i^a = \frac{1}{V} \left( \sum_{b \in \mathbb{M}} \left( \frac{\nu_i^b}{V^b} \right)^{-1} \right)^{-1}. \quad (6.6)$$

The harmonic mean is used as it corresponds to the minimum-variance estimator for a composite simulation [7, Section 6.7]. In the numerical results for this section we use two precursor weight factors given by  $\nu_i^1 = 1$  and  $\nu_i^2 = i^{-1}$  (corresponding to  $\alpha = 0$  and  $\alpha = -3$  diameter weighting exponents), and we use the optimal expressions (4.37) and (4.38).

For a numerical example we use a sedimentation kernel with an initial log-normal distribution of particles, having geometric mean diameter  $D_g = 20 \mu\text{m}$  and geometric standard deviation  $\log_{10} \sigma_g = 0.2$ . As in the example in Section 6.1, we assigned a density of  $1000 \text{ kg m}^{-3}$  and a molecular weight of  $18 \text{ g mol}^{-1}$  to the particles. The temperature was set to  $288 \text{ K}$  and the pressure was  $10^5 \text{ Pa}$ .

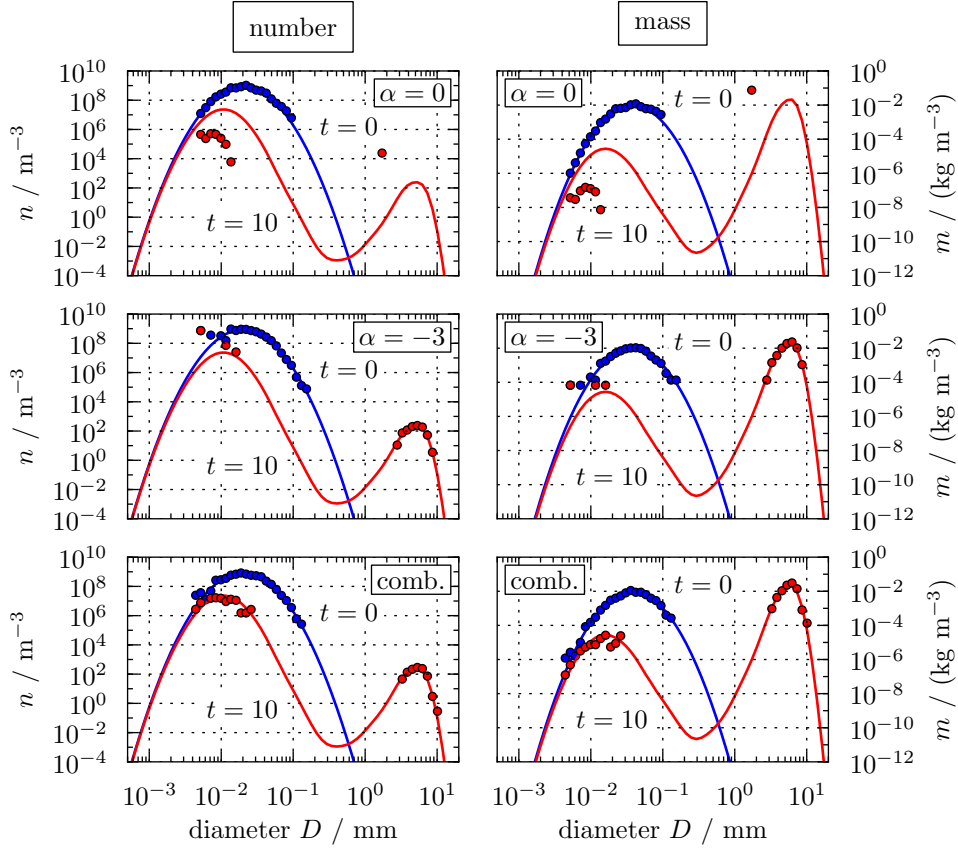


FIGURE 4. Size distributions (left column: number, right column: mass) for a sedimentation kernel computed using three different weighting schemes and  $N_p = 10^3$  particles. Top row: flat weighting. Center row: inverse-mass weighting. Bottom row: Combined flat-and-inverse-mass weighting as described in Section 6.2. The blue and red circles show the particle solution at times  $t = 0$  and  $t = 10$  min, respectively, while the solid lines are very accurate finite-volume solutions. Observe that the combined weighting scheme accurately captures both the number and mass size distributions.

The sedimentation kernel is given by

$$K_{i,j}^S = \pi (r_i + r_j) E_{i,j} |v_{s,i} - v_{s,j}|, \quad (6.7)$$

where  $r_i$  is the radius of particle  $i$ ,  $E_{i,j}$  is the collision efficiency for particles  $i$  and  $j$  according to Hall [20], and  $v_{s,i}$  is the settling velocity of particle  $i$  according to Beard [2]. The exact expressions and constant values are specified in the file `coag_kernel_sedi.F90` and its dependencies in the PartMC code [45].

We take exponential weightings with exponents ranging from  $\alpha = 0$  (flat weighting) to  $\alpha = -3$  (inverse-mass weighting), as well as a weighting composed of the combination of  $\alpha = 0$  and  $\alpha = -3$ , as described above.

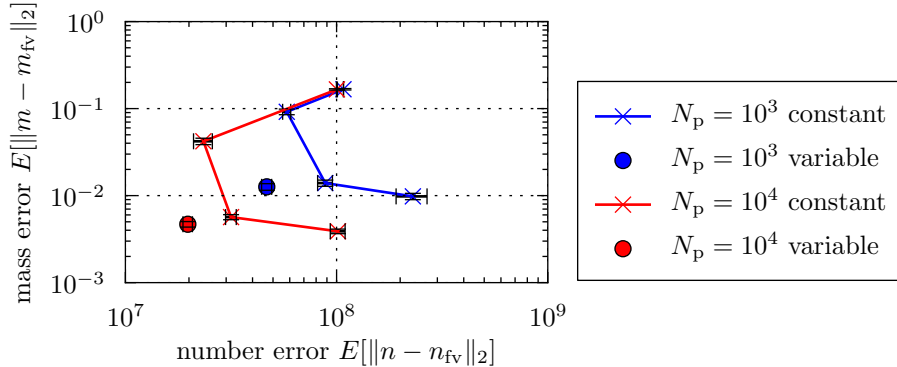


FIGURE 5. Number and mass errors for the sedimentation simulations described in Section 6.2 shown in Figure 4 for varying number of particles  $N_p$ . The solid lines show size-weighted simulations with varying weight exponent  $\alpha \in \{0, -1, -2, -3\}$  (ordered from top to bottom). The filled circles are combined flat-and-inverse-mass weighted simulations.

Figure 4 shows the number (left column) and mass (right column) size distributions at time  $t = 0$  and after  $t = 10$  min of simulation for  $N_p = 10^3$  computational particles and three different weighting schemes. We see that the  $\alpha = 0$  (flat) weighting (top row) completely fails to capture the development of the second size peak around  $D = 5$  mm and so mis-predicts both the number and size distributions. The  $\alpha = -3$  weighting (center row) is able to capture the second size peak and resolves the mass distribution fairly well, but fails to resolve the number distribution adequately. In contrast, the combined weighting scheme (bottom row) resolves both the number and size distributions well and accurately captures the evolution of the system.

The errors for varying exponent  $\alpha$  and number of particles  $N_p$  are shown in Figure 5. Changing the exponent from  $\alpha = 0$  (upper points) through to  $\alpha = -3$  (lower points) tends to increase number error while reducing mass error. This is because larger particles have lower weight with negative exponents, leading to greater resolution at larger sizes. The filled circles show the errors for the combined  $\alpha = 0$  and  $\alpha = -3$  weightings, and we see that this delivers equal-best mass error with better-than-best number error for a given number of computational particles  $N_p$ . Compared to the simple  $\alpha = 0$  weight, the combined weighting has at least a factor of 2 improvement in number error and more than 20 times improvement in mass error. Compared to the simple  $\alpha = -3$  weight, the combined weighting has about the same mass error and about 5 times small number error.

The benefits of the combined weighting are less substantial when compared to the simple  $\alpha = -1$  or  $\alpha = -2$  weights. The main advantage of the combined weighting, however, is that it delivers excellent error performance without needing to be tuned to the particular simulation. Looking at the red solid number curves (finite-volume simulations) in Figure 4, we see that the peak-to-peak straight line fit has a slope of about  $\alpha = -1.8$  on the log-scale, which is why the  $\alpha = -2$  weight works well. For different initial conditions or simulation parameters, however, this weight will not perform as well. In contrast, the combined weighting scheme adapts

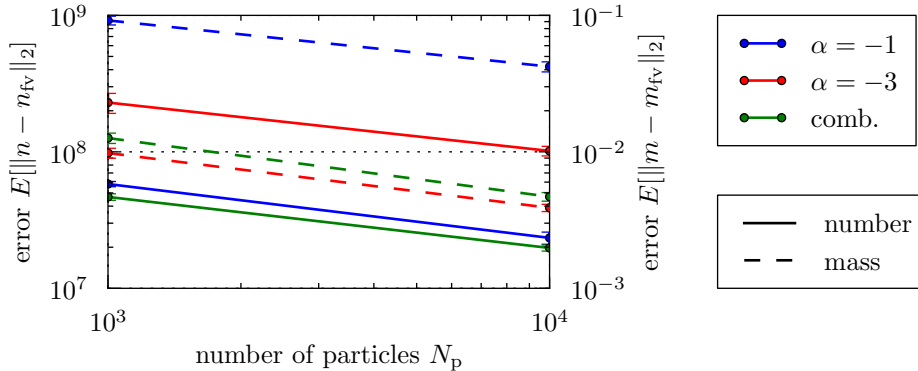


FIGURE 6. Convergence of the number and mass size distribution using the weighting scheme from Section 6.2 for different weight exponents  $\alpha$  and for the combined weighting. The baseline for computing the error is a very accurate finite-volume solution with number distribution  $n_{fv}$  and mass distribution  $m_{fv}$ . The exponents  $\alpha = -1$  and  $\alpha = -2$  have similar behavior (not plotted).

to the simulation and so delivers excellent resolution without needing to know the solution behavior in advance.

Theorem 1 implies that the particle solutions will converge to the mean-field solutions as the number of particles  $N_p$  increases. The particle error versus  $N_p$  is shown in Figure 6 for the number and mass distributions, which shows that the method is in fact converging as  $N_p \rightarrow \infty$ . For the flat weighting ( $\alpha = 0$ ) the convergent regime does not begin until  $N_p \geq 10^4$ , as is expected from Figure 4, where we see very poor behavior for  $\alpha = 0$  with only  $N_p = 10^3$  particles.

The computational cost of the simulations shown in Figure 4 is again linear in the number of coagulations that occur, as described in Section 6.1. The inverse-mass ( $\alpha = -3$ ) and combined ( $\alpha = 0$  and  $\alpha = -3$ ) weightings result in simulations that are more expensive than the flat ( $\alpha = 0$ ) weighting, because there are more coagulation events being resolved. This is because the sedimentation kernel (6.7) is largest between particles of different sizes, so the simulations that better resolve the large particles will simulate more coagulation events between small and large particles.

**6.3. Size-dependent sub-population weightings.** The two practical problems addressed by the schemes of Sections 6.1 and 6.2, namely the composition multiscaling problem and the size multiscaling problem, frequently occur simultaneously in application scenarios [23, 43]. In such cases, highly efficient simulations can be obtained by combining the independent sub-population weightings of Section 6.1 with combined size-dependent weightings of Section 6.2.

We will not present a comprehensive convergence study of the combined algorithm, but rather illustrate its behavior in Figure 7. This shows the Brownian kernel test-case from Section 6.1 with four different simulations, and both the number and mass output distributions. The four cases (top to bottom) in Figure 7 are: (1) a naive unmodified simulation, with equal weights for all particles; (2) sub-population weights using the Section 6.1 scheme, but no size-dependence; (3) size-dependent

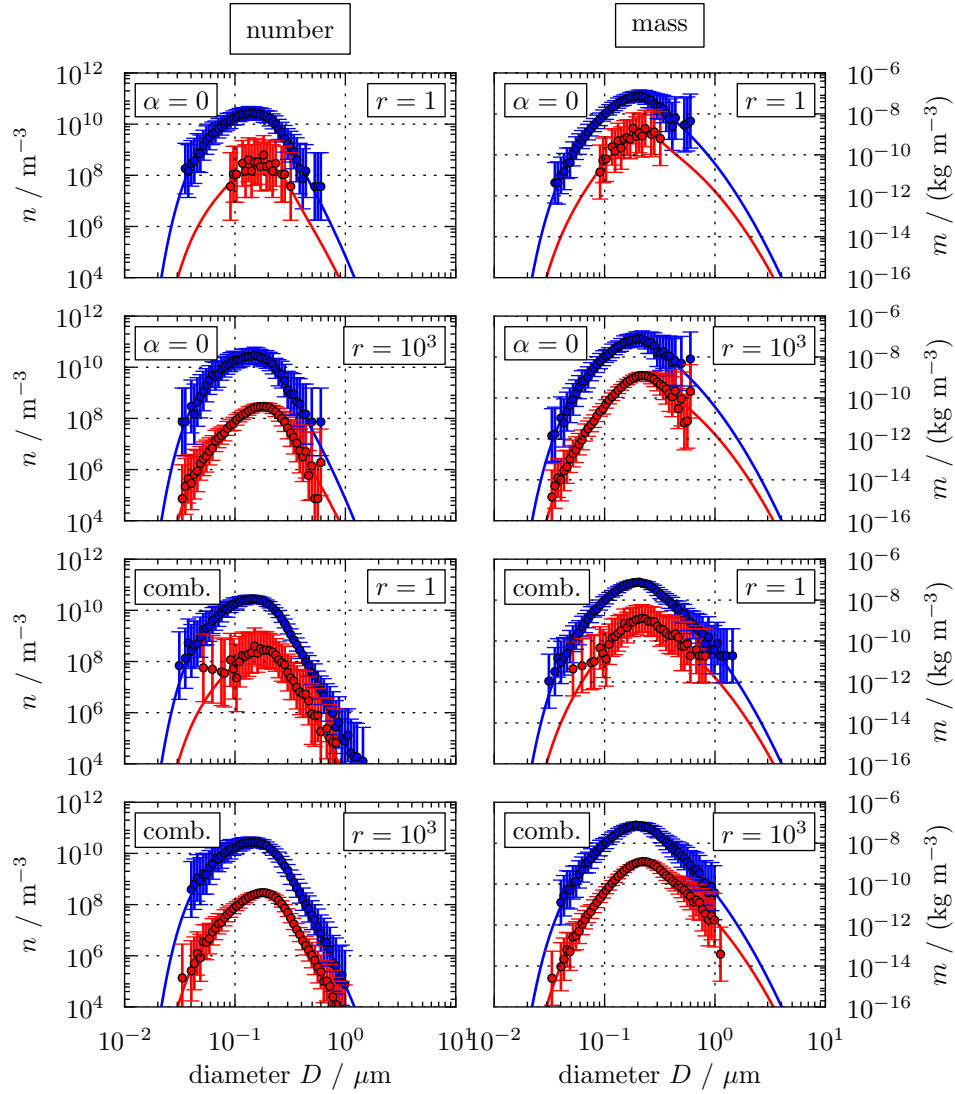


FIGURE 7. Size distributions (left column: number, right column: mass) for a Brownian kernel simulation with two particle classes (blue and red), with the same parameters as in Figure 1. Top row: flat size weighting ( $\alpha = 0$ ) with equal weights in each class ( $r = 1$ ). Second row: flat size weighting ( $\alpha = 0$ ) with equal number in each class ( $r = 10^3$ ). Third row: combined  $\alpha = 0$  and  $\alpha = -3$  size weighting with equal weights in each class ( $r = 1$ ). Bottom row: combined  $\alpha = 0$  and  $\alpha = -3$  size weighting with equal number in each class ( $r = 10^3$ ).

weights using the combined scheme of Section 6.2, but no sub-population differentiation; and (4) using the schemes of Sections 6.1 and 6.2 simultaneously.

The observed numerical behavior corresponds to that described in detail in Sections 6.1 and 6.2. The simulations (2 and 4) with differentially-weighted sub-populations have dramatically improved error in sub-population 2, as explained in Section 6.1. The simulations (3 and 4) with combined size-dependent weighting have excellent behavior for both number and mass distributions, rather than just the number distribution, as explained in Section 6.2. In summary, simulation 4 combines the advantages of both the previously outlined methods.

**7. Conclusions.** This work describes the method development of particle simulations for the Smoluchowski equation where multiple particle sub-populations exist and possibly interact. Our particular target problem considered sub-populations that may exhibit large differences in their abundances. This is a common feature in cloud microphysics and atmospherically relevant aerosol applications and poses a challenging modeling problem. We conceptualized this challenge as the “size multiscaling problem” and the “compositional multiscaling problem”.

We proved convergence for these methods under very weak assumptions and developed a family of consistency conditions that result in the correct mean during the simulation. Due to the many degrees of freedom in the parameters of the stochastic simulation, we showed that many choices gave consistency, so we also developed the conditions for the optimal consistent method. We stated and proved a family of theorems to guarantee that these methods are accurate in the limit of large population size.

Finally, we presented numerical simulations illustrating the applicability and efficacy for two atmospheric relevant aerosol systems, one representative of the size multiscaling problem using the sedimentation kernel, and the other representative of the compositional multiscaling problem, using the Brownian kernel. Since in practice the two problems often occur simultaneously, we also showed that the methods can be combined and result in excellent behavior for both number and mass distributions.

**Acknowledgments.** This work was funded by the National Science Foundation (NSF) under grants CMG-0934491, CMMI-1150490, and AGS-1254428 and the Department of Energy (DOE) under grant DE-SC0019192. The authors gratefully acknowledge the use of the Taub cluster parallel computing resource provided by the Computational Science and Engineering Program at the University of Illinois.

## REFERENCES

- [1] H. Babovsky, [On a Monte Carlo scheme for Smoluchowski’s coagulation equation](#), *Monte Carlo Methods and Appl.*, **5** (1999), 1–18.
- [2] K. V Beard, [Terminal velocity and shape of cloud and precipitation drops aloft](#), *J. Atmos. Sci.*, **33** (1976), 851–864.
- [3] A. Bott, [A flux method for the numerical solution of the stochastic collection equation](#), *J. Atmos. Sci.*, **55** (1998), 2284–2293.
- [4] J. H. Curtis, M. D. Michelotti, N. Riemer, M. Heath and M. West, [Accelerated simulation of stochastic particle removal processes in particle-resolved aerosol models](#), *J. Comput. Phys.*, **322** (2016), 21–32.
- [5] M. H. A. Davis, *Markov Models and Optimization*, Chapman and Hall, Boundary Row, London, 1993.
- [6] E. Debry, B. Sportisse and B. Jourdain, [A stochastic approach for the numerical simulation of the general dynamics equations for aerosols](#), *J. Comput. Phys.*, **184** (2003), 649–669.



- [7] L. DeVile, N. Riemer and M. West, [Weighted flow algorithms \(WFA\) for stochastic particle coagulation](#), *J. Comput. Phys.*, **230** (2011), 8427–8451.
- [8] J. L. Doob, *Stochastic Processes*, Wiley Classics Library. John Wiley & Sons Inc., New York, 1990. ISBN 0-471-52369-0. Reprint of the 1953 original, A Wiley-Interscience Publication.
- [9] Y. Efendiev and M. R. Zachariah, [Hybrid Monte Carlo method for simulation of two-component aerosol coagulation and phase segregation](#), *J. Colloid Interf. Sci.*, **249** (2002), 30–43.
- [10] Y. Efendiev, H. Struchtrup, M. Luskin and M. R. Zachariah, [A hybrid sectional-moment model for coagulation and phase segregation in binary liquid nanodroplets](#), *J. Nanopart. Res.*, **4** (2002), 61–72.
- [11] A. Eibeck and W. Wagner, [An efficient stochastic algorithm for studying coagulation dynamics and gelation phenomena](#), *SIAM J. Sci. Comput.*, **22** (2000), 802–821.
- [12] A. Eibeck and W. Wagner, [Approximative solution of the coagulation-fragmentation equation by stochastic particle systems](#), *Stochastic Anal. Appl.*, **18** (2000), 921–948.
- [13] A. Eibeck and W. Wagner, [Stochastic particle approximations for Smoluchowski’s coagulation equation](#), *Ann. Appl. Probab.*, **11** (2001), 1137–1165.
- [14] A. Eibeck and W. Wagner, [Stochastic interacting particle systems and nonlinear kinetic equations](#), *Ann. Appl. Probab.*, **13** (2003), 845–889.
- [15] D. T. Gillespie, [The stochastic coalescence model for cloud droplet growth](#), *J. Atmos. Sci.*, **29** (1972), 1496–1510.
- [16] D. T. Gillespie, [An exact method for numerically simulating the stochastic coalescence process in a cloud](#), *J. Atmos. Sci.*, **32** (1975), 1977–1989.
- [17] D. T. Gillespie, [A general method for numerically simulating the stochastic time evolution of coupled chemical reactions](#), *J. Comput. Phys.*, **22** (1976), 403–434.
- [18] D. T. Gillespie, [Exact stochastic simulation of coupled chemical reactions](#), *J. Phys. Chem.*, **81** (1977), 2340–2361.
- [19] D. T. Gillespie, *Markov Processes: An Introduction for Physical Scientists*, Academic Press, 1992.
- [20] W. D. Hall, [A detailed microphysical model within a two-dimensional dynamic framework: Model description and preliminary results](#), *J. Atmos. Sci.*, **37** (1980), 2486–2507.
- [21] L. E. Hatch, J. M. Creamean, A. P. Ault, J. D. Surratt, M. N. Chan, J. H. Seinfeld, E. S. Edgerton, Y. Su and K. A. Prather, [Measurements of isoprene-derived organosulfates in ambient aerosols by aerosol time-of-flight mass spectrometry—part 1: Single particle atmospheric observations in Atlanta](#), *Environ. Sci. Technol.*, **45** (2011), 5105–5111.
- [22] L. M. Hildemann, G. R. Markowski, M. C. Jones and G. R. Cass, [Submicrometer aerosol mass distributions of emissions from boilers, fireplaces, automobiles, diesel trucks, and meat-cooking operations](#), *Aerosol Sci. Technol.*, **14** (1991), 138–152.
- [23] M. Hughes, J. K. Kodros, J. R. Pierce, M. West and N. Riemer, [Machine learning to predict the global distribution of aerosol mixing state metrics](#), *Atmosphere*, **9** (2018), 15.
- [24] R. Irizarry, [Fast Monte Carlo methodology for multivariate particulate systems—I: Point ensemble Monte Carlo](#), *Chem. Eng. Sci.*, **63** (2008), 95–110.
- [25] R. Irizarry, [Fast Monte Carlo methodology for multivariate particulate systems—II:  \$\tau\$ -PEMC](#), *Chem. Eng. Sci.*, **63** (2008), 111–121.
- [26] M. Z. Jacobson, *Fundamentals of Atmospheric Modeling*, Cambridge University Press, 2005.
- [27] M. Z. Jacobson, R. P. Turco, E. J. Jensen and O. B. Toon, [Modeling coagulation among particles of different composition and size](#), *Atmos. Environ.*, **28** (1994), 1327–1338.
- [28] A. Kolodko and K. Sabelfeld, [Stochastic particle methods for Smoluchowski coagulation equation: Variance reduction and error estimations](#), *Monte Carlo Methods Appl.*, **9** (2003), 315–339.
- [29] A. B. Kostinski and R. A. Shaw, [Fluctuations and luck in droplet growth by coalescence](#), *Bull. Amer. Meteor. Soc.*, **86** (2005), 235–244.
- [30] G. Kotalczyk and F. E. Kruijs, [A Monte Carlo method for the simulation of coagulation and nucleation based on weighted particles and the concepts of stochastic resolution and merging](#), *J. Comput. Phys.*, **340** (2017), 276–296.
- [31] T. G. Kurtz, [Strong approximation theorems for density dependent Markov chains](#), *Stochastic Processes Appl.*, **6** (1977/78), 223–240.
- [32] A. Maisels, F. E. Kruijs and H. Fissan, [Direct simulation Monte Carlo for simultaneous nucleation, coagulation, and surface growth in dispersed systems](#), *Chem. Eng. Sci.*, **59** (2004), 2231–2239.

- [33] R. McGraw and D. L. Wright, [Chemically resolved aerosol dynamics for internal mixtures by the quadrature method of moments](#), *J. Aerosol Sci.*, **34** (2003), 189–209.
- [34] B. Øksendal, *Stochastic Differential Equations*, Universitext. Springer-Verlag, Berlin, sixth edition, 2003. ISBN 3-540-04758-1. An introduction with applications.
- [35] R. I. A. Patterson, W. Wagner and M. Kraft, [Stochastic weighted particle methods for population balance equations](#), *J. Comput. Phys.*, **230** (2011), 7456–7472.
- [36] A. Petzold, J. A. Ogren, M. Fiebig, P. Laj, S.-M. Li, U. Baltensperger, T. Holzer-Popp, S. Kinne, G. Pappalardo, N. Sugimoto, et al., [Recommendations for reporting “black carbon” measurements](#), *Atmos. Chem. Phys.*, **13** (2013), 8365–8379.
- [37] X. Qin, K. A. Pratt, L. G. Shields, S. M. Toner and K. A. Prather, [Seasonal comparisons of single-particle chemical mixing state in Riverside, CA](#), *Atmos. Environ.*, **59** (2012), 587–596.
- [38] N. Riemer, H. Vogel, B. Vogel and F. Fiedler, [Modeling aerosols on the mesoscale  \$\gamma\$ , part I: Treatment of soot aerosol and its radiative effects](#), *J. Geophys. Res.*, **108** (2003), 4601.
- [39] N. Riemer, M. West, R. A. Zaveri and R. C. Easter, [Simulating the evolution of soot mixing state with a particle-resolved aerosol model](#), *J. Geophys. Res.*, (2009), D09202.
- [40] J. H. Seinfeld and S. Pandis, *Atmospheric Chemistry and Physics*, Wiley, 2016.
- [41] S. Shima, K. Kusano, A. Kawano, T. Sugiyama and S. Kawahara, [The super-droplet method for the numerical simulation of clouds and precipitation: A particle-based and probabilistic microphysics model coupled with a non-hydrostatic model](#), *Q. J. R. Meteorol. Soc.*, **135** (2009), 1307–1320.
- [42] A. Shwartz and A. Weiss, *Large Deviations for Performance Analysis*, Chapman & Hall, London, 1995.
- [43] J. Tian, N. Riemer, M. West, L. Pfaffenberger, H. Schlager and A. Petzold, [Modeling the evolution of aerosol particles in a ship plume using PartMC-MOSAIC](#), *Atmos. Chem. Phys.*, **14** (2014), 5327–5347.
- [44] C. G. Wells and M. Kraft, [Direct simulation and mass flow stochastic algorithms to solve a sintering-coagulation equation](#), *Monte Carlo Methods Appl.*, **11** (2005), 175–197.
- [45] M. West, N. Riemer, J. Curtis, M. Michelotti and J. Tian, [PartMC: Particle-resolved Monte-Carlo atmospheric aerosol simulation](#), version 2.5.0, 2018.
- [46] C. Yoon and R. McGraw, [Representation of generally mixed multivariate aerosols by the quadrature method of moments: II. Aerosol dynamics](#), *J. Aerosol Sci.*, **35** (2004), 577–598.
- [47] H. Zhao and C. Zheng, [Correcting the Multi-Monte Carlo Method for particle coagulation](#), *Powder Technol.*, **193** (2009), 120–123.
- [48] H. Zhao, C. Zheng and M. Xu, [Multi-Monte Carlo Method for coagulation and condensation/evaporation in dispersed systems](#), *J. Colloid Interface Sci.*, **286** (2005), 195–208.
- [49] H. Zhao, F. E. Kruijs and C. Zheng, [Reducing statistical noise and extending the size spectrum by applying weighted simulation particles in Monte Carlo simulation of coagulation](#), *Aerosol Sci. Technol.*, **43** (2009), 781–793.

*E-mail address:* [rdeville@illinois.edu](mailto:rdeville@illinois.edu)

*E-mail address:* [nriemer@illinois.edu](mailto:nriemer@illinois.edu)

*E-mail address:* [mwest@illinois.edu](mailto:mwest@illinois.edu)