



Accelerated simulation of stochastic particle removal processes in particle-resolved aerosol models



J.H. Curtis^a, M.D. Michelotti^b, N. Riemer^a, M.T. Heath^b, M. West^{c,*}

^a Department of Atmospheric Sciences, University of Illinois at Urbana–Champaign, 105 S. Gregory St., Urbana, IL 61801, USA

^b Department of Computer Science, University of Illinois at Urbana–Champaign, 201 North Goodwin Avenue, Urbana, IL 61801, USA

^c Department of Mechanical Science and Engineering, University of Illinois at Urbana–Champaign, 1206 W. Green St., Urbana, IL 61801, USA

ARTICLE INFO

Article history:

Received 27 December 2015

Received in revised form 13 June 2016

Accepted 16 June 2016

Available online 21 June 2016

Keywords:

Aerosol modeling

Stochastic simulations

Particle methods

Aerosol removal

ABSTRACT

Stochastic particle-resolved methods have proven useful for simulating multi-dimensional systems such as composition-resolved aerosol size distributions. While particle-resolved methods have substantial benefits for highly detailed simulations, these techniques suffer from high computational cost, motivating efforts to improve their algorithmic efficiency. Here we formulate an algorithm for accelerating particle removal processes by aggregating particles of similar size into bins. We present the Binned Algorithm for particle removal processes and analyze its performance with application to the atmospherically relevant process of aerosol dry deposition. We show that the Binned Algorithm can dramatically improve the efficiency of particle removals, particularly for low removal rates, and that computational cost is reduced without introducing additional error. In simulations of aerosol particle removal by dry deposition in atmospherically relevant conditions, we demonstrate about 50-times increase in algorithm efficiency.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Monte Carlo particle techniques are used in various disciplines to solve population balance equations numerically. Particle methods have been applied to cloud physics [1–3], polymer science [4–6], and aerosol physics [7–15], the focus of this paper. Particle-resolved techniques are particularly advantageous for simulating multidimensional systems and do not suffer from artificial numerical diffusion found in implementations of finite difference methods. Unfortunately, the computational expense of these techniques remains a substantial drawback.

An atmospheric aerosol population is a multidimensional system in the space of aerosol composition. Aerosol particles consist of a complex mixture of chemical species, such as soluble inorganic salts and acids, insoluble crustal materials (dust), trace metals, and carbonaceous materials [16–19]. The aerosol life-cycle consists of three stages: formation, transformation, and removal. Atmospheric aerosols can originate from primary emission sources or may be formed by nucleation. The composition of aerosol particles evolves over time due to coagulation with other particles and to mass transfer to and from gas-phase species. Ultimately, aerosol particles are removed from the atmosphere by precipitation (“wet deposition”) or by deposition to the ground or other surfaces in the absence of precipitation (“dry deposition”).

* Corresponding author.

E-mail address: mwest@illinois.edu (M. West).

To model the multidimensional aerosol size distribution, particle-resolved models [10] explicitly resolve the evolution of individual aerosol particles within a representative volume of air. These processes can include coagulation, gas-particle mass transfer, emission, dilution with background air, and wet or dry deposition.

In this paper we use the particle-resolved model PartMC-MOSAIC (Particle Monte Carlo—Model for Simulating Aerosol Interactions and Chemistry) [20] as a testbed. While particle-based models have the advantage of avoiding *a priori* assumptions regarding particle composition, they can be very expensive computationally, resulting in the need for highly efficient algorithms [10,21–23]. Efficient implementations, such as the PartMC-MOSAIC code, have storage cost proportional to the number of particles, computational cost for evaporation/condensation proportional to the number of particles, and computational cost for coagulation proportional to the number of coagulation events.

In this paper, we focus on the simulation of dry deposition as an important representative of aerosol removal processes, however our method applies to single-event particle removal processes in general. Dry deposition—the removal of particles in absence of precipitation—determines the particles' residence time in the atmosphere. Calculating the dry deposition rate is highly complex due to its dependence on physical and chemical properties of the aerosol, land surface characteristics, and meteorological conditions. As dry deposition is a size-dependent process, it also shapes the aerosol size distribution.

A naive approach to simulating dry deposition results in computational cost proportional to the number of particles. The naive approach becomes computationally expensive for simulations that involve a large number of particles and becomes computationally prohibitive for inclusion of particle-resolved aerosol representation within 3D regional weather models. Here we present an algorithm, based on the Stochastic Simulation Algorithm (SSA), to improve the performance of particle removal due to dry deposition. The contribution of this paper is (1) a new algorithm that uses one-dimensional binning and geometric sampling while maintaining exactness and (2) its application to the process of dry deposition where we showed a very significant reduction in the computational cost with no additional error incurred.

The Stochastic Simulation Algorithm (SSA) [24,25] is a Monte Carlo procedure for producing an exact realization of a continuous-time Markov chain. The original formulation of SSA was designed for Markov chains that involved pairwise interacting entities, such as coagulating aerosol particles. However, it can be modified easily to consider events that involve only a single entity, such as stochastic particle deposition.

The high computational cost of SSA has motivated the development of more efficient variants for specific applications. Some of these retain the exactness of SSA (e.g. [26–28]), while others sacrifice exactness to gain numerical efficiency. Popular examples of approximate methods are Tau Leaping [29] and later variants [30–34], which can achieve greater efficiency by simulating all events within a small time interval τ but which does not generate exact realizations of the stochastic process. In [22], binned versions of SSA and Tau Leaping were developed that are more efficient for particle processes. These binned variants have similar properties to the original methods, in that Binned SSA is exact while Binned Tau Leaping can be more efficient but is not exact, and they can both be adapted to single-particle processes such as deposition.

In this paper, we present a new binned algorithm (Algorithm 2) that combines the advantages of Binned Tau Leaping and Binned SSA when applied to single-particle non-interacting processes such as particle deposition. This new algorithm is at least as efficient as Binned Tau Leaping and follows it in using a discrete time step τ and being simple to implement. However, it retains the exactness of Binned SSA, making it both fast and exact. We demonstrate the performance of the new algorithm using numerical results from an implementation of this algorithm that we integrated with the PartMC software library [10,35,36,21].

2. Formulation

Consider a set of possible “particles” \mathcal{P} . In our model problem, we model atmospheric aerosol particles and represent each particle by the volume of d different chemical species, where $d = 20$ is typical. We will also impose minimum and maximum volumes v_{\min} and v_{\max} . Thus, we will use the following set:

$$\mathcal{P} = \left\{ p \in \mathbb{R}^d : p \geq \mathbf{0}, v_{\min} \leq \|p\|_1 \leq v_{\max} \right\}. \quad (1)$$

Let $\pi \subset \mathcal{P}$ be the finite set of particles currently present in the simulation. There may be multiple identical particles, so π is a multiset in the sense of Knuth [37, p. 473]. Throughout a simulation, these particles may be affected by several processes. For atmospheric aerosol modeling, the set of particles π may be affected by emissions, coagulation, condensation, chemical reactions, etc.

Consider the process of particle deposition. Let $K : \mathcal{P} \rightarrow \mathbb{R}$ be a kernel denoting the rate of deposition. A particle $p \in \pi$ should be removed from the simulation at stochastic rate $K(p)$. If p and K are not affected by any other processes, the time at which p is removed from the simulation should be chosen from an exponential distribution with rate $K(p)$. However, p and K are likely to be affected by other processes, so it is better to take a time-stepping approach in which we must determine which particles should be removed from the simulation over a time step of size τ .

Algorithm 1 is a Naive Algorithm to simulate particle deposition over a time step of size τ , which checks each particle individually. The function `randUnif()` returns a uniform random number from $[0, 1)$. Note that Algorithm 1 requires $O(|\pi|)$ computation time, regardless of how many particles are actually removed. This is reasonable if we have many removals, but suppose the number of removals is much less than $|\pi|$. In this case, the algorithm is inefficient.

Algorithm 1 Naive Algorithm.

```

1: for  $p \in \pi$  do
2:   if randUnif() <  $1 - e^{-\tau K(p)}$  then
3:     remove  $p$  from  $\pi$ 
4:   end if
5: end for

```

Theorem 2.1. *Algorithm 1 produces an exact sampling of the probability of the output π at time τ .*

Proof. This follows immediately from independence of the trials for each particle. \square

Let us consider a binned approach. Partition \mathcal{P} into m sets, denoted r_1, r_2, \dots, r_m . These sets will be referred to as bins. While many binning strategies are in principle possible, in practice we follow Michelotti et al. [22] and use a magnitude operator $|\cdot|: \mathcal{P} \rightarrow \mathbb{R}$ such as particle volume, diameter, mass, etc., and define bins by $r_i = \{p \in \mathcal{P} : \nu_{i-1} < |p| \leq \nu_i\}$ for each $i = 1, \dots, m$ and a strictly increasing sequence $\nu_0, \dots, \nu_m \in \mathbb{R}$ of bin boundaries. We will use the following definitions to describe the bins:

$$r(p) = r_i \text{ where } i \text{ is chosen such that } p \in r_i, \tag{2}$$

$$\pi_i = \pi \cap r_i, \tag{3}$$

$$K_{\text{up}}(r_i) \geq K(p) \text{ for all } p \in r_i. \tag{4}$$

The sets π_i are referred to as the bin contents and partition the set of particles according to the bins. The function K_{up} provides an upper bound on the kernel for each bin. Its value may be known explicitly, determined by an optimization algorithm, or estimated by sampling possible values and applying an overestimation factor.

Algorithm 2 Binned Algorithm.

```

1: for  $i = 1, 2, \dots, m$  do
2:    $k \leftarrow |\pi_i| + 1$ 
3:   loop
4:      $k \leftarrow k - \text{randGeom}(1 - e^{-\tau K_{\text{up}}(r_i)})$ 
5:     if  $k \leq 0$  then
6:       break out of loop
7:     end if
8:      $p \leftarrow k$ -th element from  $\pi_i$ 
9:     if randUnif() <  $(1 - e^{-\tau K(p)}) / (1 - e^{-\tau K_{\text{up}}(r_i)})$  then
10:      remove  $p$  from  $\pi_i$ 
11:     end if
12:   end loop
13: end for

```

In [Algorithm 2](#) we present a Binned Algorithm to simulate particle deposition over a time step of size τ . The function `randGeom(s)` returns a geometric random number with success rate s (we will use the version of geometric random numbers defined as the number of Bernoulli trials needed until a success is observed, implying `randGeom(s)` is at least 1). This algorithm utilizes the binning structure and iterates over particles with geometrically-sampled jumps. This strategy avoids the need to check every particle, significantly reducing the computation time.

Theorem 2.2. *The probability distribution of the random output π is the same for both [Algorithm 1](#) and [Algorithm 2](#).*

Proof. For each particle $p \in \pi$ initially present, the probability that [Algorithm 1](#) removes p from π is $1 - e^{-\tau K(p)}$. It is clear in both algorithms that the decision using `randUnif()` to remove p is independent of the decision to remove any other particle. Thus, we need to demonstrate only that [Algorithm 2](#) removes p from π with probability $1 - e^{-\tau K(p)}$.

Let i be the bin index corresponding to particle p , so that $p \in \pi_i$. [Algorithm 2](#) iterates over the bin contents π_i using step-sizes of length `randGeom(1 - e-τKup(ri))`. Recall that the definition of a random geometric number is the number of Bernoulli trials needed until a success is observed. Thus, the probability that p will even be considered for removal is $1 - e^{-\tau K_{\text{up}}(r_i)}$. If p is considered for removal, then we remove p with probability $(1 - e^{-\tau K(p)}) / (1 - e^{-\tau K_{\text{up}}(r_i)})$ (note that this is between 0 and 1). Thus, the probability that p will be removed is

$$\Pr(p \text{ removed}) = \left(1 - e^{-\tau K_{\text{up}}(r_i)}\right) \frac{1 - e^{-\tau K(p)}}{1 - e^{-\tau K_{\text{up}}(r_i)}} = 1 - e^{-\tau K(p)}, \tag{5}$$

which is the desired result. \square

Corollary 2.3. *Algorithm 2 produces an exact sampling of the probability of the output π at time τ .*

Proof. This simply combines [Theorems 2.2 and 2.1](#). \square

In order to implement [Algorithm 2](#) efficiently, we must be careful in representing π_i . In particular, implementing π_i as an unsorted array of particles allows us to remove a particle from anywhere in the array in constant time. Note that we iterate backwards over this array so that the removal of a particle does not alter the particles we will consider in the future. Since we iterate over particles in order, there will be fewer cache misses than if we were to use an algorithm like Binned SSA, which accesses particles from the array at random.

The geometric distribution is easy to sample via the inverted cumulative distribution function, which gives

$$\text{randGeom}(s) \leftarrow 1 + \left\lfloor \frac{\ln(\text{randUnif}())}{\ln(1-s)} \right\rfloor. \quad (6)$$

Note that the way `randGeom` is called in [Line 4 of Algorithm 2](#) results in the further simplification

$$\text{randGeom}\left(1 - e^{-\tau K_{\text{up}}(r_i)}\right) \leftarrow 1 + \left\lfloor -\frac{\ln(\text{randUnif}())}{\tau K_{\text{up}}(r_i)} \right\rfloor. \quad (7)$$

When implementing this, we must take into account the possibility of integer overflow.

Define K_0 as the overall expected event rate and M as the overall expected candidate rate,

$$K_0 := \sum_{p \in \mathcal{P}} (1 - e^{-\tau K(r(p))}) \quad (8)$$

$$M := \sum_{p \in \mathcal{P}} (1 - e^{-\tau K_{\text{up}}(r(p))}) = \sum_{i=1}^m |\pi_i| (1 - e^{-\tau K_{\text{up}}(r_i)}). \quad (9)$$

Following the analysis in [Michelotti et al. \[22\]](#), we define the sampling efficiency of [Algorithm 2](#) for a given particle state by

$$\text{Eff} := \frac{K_0}{M}. \quad (10)$$

By definition, $0 \leq \text{Eff} \leq 1$ and $\text{Eff} \approx 1$ means that the algorithm is close to sampling the minimum number of candidate events. To study the sampling efficiency it is convenient to introduce the lower bound K_{\min} and overestimation K_{over} ,

$$K_{\min}(r_i) := \inf\{K(p) : p \in r_i\}, \quad (11)$$

$$K_{\text{over}} := \max\{K_{\text{up}}(r_i) - K_{\min}(r_i) : i = 1, \dots, m\}. \quad (12)$$

Theorem 2.4. *Algorithm 2 has $\text{Eff} \rightarrow 1$ as $K_{\text{over}} \rightarrow 0$ for fixed particle set π and fixed time step τ .*

Proof. We compute an upper bound for $M - K_0$ as follows.

$$\begin{aligned} M - K_0 &= \sum_{p \in \mathcal{P}} \left((1 - e^{-\tau K_{\text{up}}(r(p))}) - (1 - e^{-\tau K(r(p))}) \right) \\ &\leq \sum_{p \in \mathcal{P}} (e^{-\tau K_{\min}(r(p))} - e^{-\tau K_{\text{up}}(r(p))}) \\ &\leq |\pi| \tau K_{\text{over}}, \end{aligned}$$

where we use the fact that τ is a Lipschitz constant for $e^{-\tau x}$ on $\mathbb{R}_{\geq 0}$. Observing that M is bounded below by K_0 implies

$$0 \leq 1 - \text{Eff} = \frac{M - K_0}{M} \leq \frac{|\pi|}{K_0} \tau K_{\text{over}},$$

and because π and K_0 are fixed for a given set of particles this gives the desired result. \square

[Theorem 2.4](#) shows that the sampling efficiency of [Algorithm 2](#) improves as the upper bound K_{up} approaches the kernel lower bound for each bin, which would typically be achieved by decreasing the size of each bin and increasing the number of bins. However, increasing bin number will tend to increase the computational cost, so it is necessary to understand the resulting trade-off.

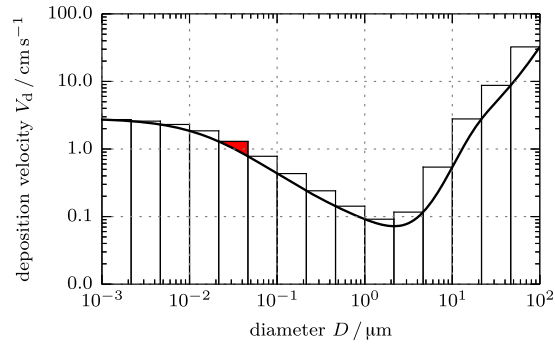


Fig. 1. Schematic of Binned Algorithm applied to a dry deposition velocity curve. Binned Algorithm applies a logarithmic grid along the particle diameter axis where particles in a given bin have similar removal rates. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Algorithm 2 has expected running time $O(m + M)$. Note that M is bounded above by $|\pi|$, so we can expect the running time to be at most $O(|\pi|)$, assuming we have fewer bins than particles. Note also that M is bounded below by the expected number of removals to occur in this time step. If the expected number of removals is near $|\pi|$, then we will achieve performance similar to the Naive Algorithm. However, if the $1 - e^{-\tau K_{up}(r_i)}$ values are small, then $M \ll |\pi|$ and we may have a much faster algorithm. Naturally we also want to have $m \ll |\pi|$, so there is some trade-off between the number of bins and the removal candidate rate. Depending on the running time constants, it may be beneficial to consider a hybrid algorithm where we use the naive approach if $1 - e^{-\tau K_{up}(r_i)} \approx 1$ and use the geometric approach otherwise.

3. Implementation for particle removal due to dry deposition in PartMC

Due to the complexity of the physical and chemical processes, dry deposition is typically parameterized within models by the particle dry deposition velocity, which captures the removal processes from Brownian diffusion, turbulent transport, particle impaction with the surface, particle interception with surface features, and gravitational settling. Dry deposition velocity is a function of particle size and density, as well as environmental conditions. For very small particles, which behave much like gases, Brownian diffusion is an effective pathway for particle removal. For very large particles, gravitational settling becomes a more effective removal pathway due to large particle mass. Within the intermediate size range, particles experience a relative minimum in removal rates due to those transport processes being less effective.

To parameterize particle loss due to dry deposition within PartMC, we implemented the size-dependent dry deposition scheme described in [38]. To determine the dry deposition velocity, equations (1), (2), (3), (5), (6), (7c), (8) and (9) of [38] are evaluated based on per-particle size and composition. The exact implementation of this scheme is detailed further in Appendix A.

Given computed particle deposition velocities and subsequent loss rates, particles are possibly removed from the aerosol population. This removal proceeds either by the Naive Algorithm, given by Algorithm 1, or the Binned Algorithm, given by Algorithm 2. The Naive Algorithm proceeds by determining each particle's deposition rate and tests for possible particle removal. The Binned Algorithm is illustrated by Fig. 1, where a representative dry deposition velocity curve is shown with the particle diameter axis divided into 15 logarithmically spaced bins for particle sorting. The efficiency of this algorithm relies on minimizing the amount of overestimation for a given bin, shaded for one bin in red as an example. In practice, a larger number of bins is used to minimize the amount of overestimation. Particles within a given bin consist of different chemical components and therefore particle density varies amongst the particle population within one bin. To address this, the upper bound of the removal rate within a given bin is calculated based on the particle diameters of the bin while assuming that the particle is composed of the species with the largest density. The result is that, within a bin, particles with lower densities have a lower adjusted probability of being removed which makes the accept-reject procedure slightly less efficient. In practice, it turns out that differences in particle density have only a small impact on removal rates because density in atmospherically-relevant particles varies by a factor of about 3.

4. Numerical experiments

4.1. Aerosol distribution functions

We take $N(D)$ to be the cumulative number distribution, which gives the number of particles per unit volume that have a diameter less than D . Given the cumulative number distribution, we define the number distribution $n(D)$ as a function of diameter by

$$n(D) = \frac{dN(D)}{d \log D}. \quad (13)$$

Table 1
Initial aerosol distribution parameters.

Mode name	N/m^{-3}	$D_{\text{gn}}/\mu\text{m}$	σ_g
Small	1.8×10^9	0.01	1.334
Intermediate	1.8×10^9	1.00	1.334
Large	1.8×10^9	10.0	1.334

The initial distribution used for each mode was given by a log-normal distribution, defined by

$$n(D) = \frac{N}{\sqrt{2\pi} \log \sigma_g} \exp\left(-\frac{(\log D - \log D_{\text{gn}})^2}{2(\log \sigma_g)^2}\right), \quad (14)$$

where N is the total number concentration, D_{gn} is the geometric mean diameter and σ_g is the geometric standard deviation. The initial set of particles $\pi(0)$ is Poisson-sampled from $n(D)$ using the procedure in [10] so that the expected number of initial particles $\mathbb{E}[|\Pi(0)|]$ is a prescribed value N_p .

In order to display the number distribution $n(D)$, we post-process particle resolved results into a one-dimensional histogram with n_{bin} bins for a given range of diameters d_{min} to d_{max} . To display the results and compute the error in the solution in Sect. 4.3, we used 150 logarithmically spaced bins from 0.001 μm to 100 μm .

4.2. Model setup

For this study, we considered an idealized scenario for particle removal due to dry deposition where simulations proceeded in the absence of all other processes that affect the aerosol size distribution, such as coagulation, condensation, emission and dilution with the background air. The purpose of the simulations was to isolate and clearly demonstrate the capabilities of the two removal algorithms. The isolation of dry deposition also permits the calculation of the extant analytical solution for comparison with numerical results, as detailed in Appendix B.

We prescribed conditions representative of the atmosphere. The meteorological conditions were constant, with temperature $T = 288$ K, pressure $p = 100$ kPa and relative humidity $\text{RH} = 99\%$. The value of relative humidity was chosen to minimize the ability of particles to rebound from the surface. We set a mean wind of 5 m s^{-1} at a reference height of 20 m and neutral atmospheric stability, corresponding to Class D of the Pasquill Stability Class [39]. A box height of 50 m was selected to represent a typical lowest model layer.

We prescribed the land use category (LUC) to be crops/mixed farming and the seasonal category (SC) to be midsummer with lush vegetation, corresponding to LUC 7 and SC 1 respectively from Table 2 of [38]. Given this choice of the LUC and SC parameters, the roughness height $Z_0 = 0.1$ m, the characteristic radius of collectors $A = 2$ mm and specified constants $\alpha = 1.2$ and $\gamma = 0.54$ were applied to all simulations as determined from Table 3 of [38].

Particle size is a major factor determining dry deposition velocity as shown in Fig. 1. To explore this dependence on particle size, we selected three initial aerosol modes where only the geometric mean diameter was varied as given in Table 1. Particles in the Small mode, with $D_{\text{gn}} = 0.01$ μm , are efficiently removed due to the contribution of Brownian diffusion. Particles in the Large mode, with $D_{\text{gn}} = 10$ μm , experience enhanced dry deposition velocities due to impaction and gravitational settling. Particles within the Intermediate mode, with $D_{\text{gn}} = 1.0$ μm , experience a minimum in the dry deposition velocity as no mechanism is particularly effective in this size range.

To quantify the performance of the Naive Algorithm and the Binned Algorithm, we performed an ensemble of runs varying the expected initial number of computational particles N_p as 10^3 , 10^4 , 10^5 , and 10^6 , where the number of the ensemble runs n_{run} was 100.

4.3. Numerical results

Fig. 2 shows the ensemble average of the number distribution $n(D)$ for $N_p = 10^4$ at $t = 0, 3,$ and 6 h for each of the three modes given in Table 1 for the Naive Algorithm, the Binned Algorithm and the exact solution. The dry deposition velocity V_d is also shown to visualize the expected removal rates for each of the modes. The Small mode experienced the highest deposition velocities and decreased in number the most over time. Due to low dry deposition velocities for particles in the Intermediate mode, the number concentration decayed slowly. The particles in Large mode varied over an order of magnitude in terms of dry deposition velocities, which resulted primarily in loss of the larger particles of that mode. We see that the PartMC-simulated number distribution for both algorithms agrees well with the analytical solution.

We quantified the error in number distribution for ensemble member j by the L^2 error:

$$\|n(D) - \Phi(D)\|_2 = \sqrt{\int_{-\infty}^{\infty} (n(D) - \Phi(D))^2 d \log D}, \quad (15)$$

which was approximated by

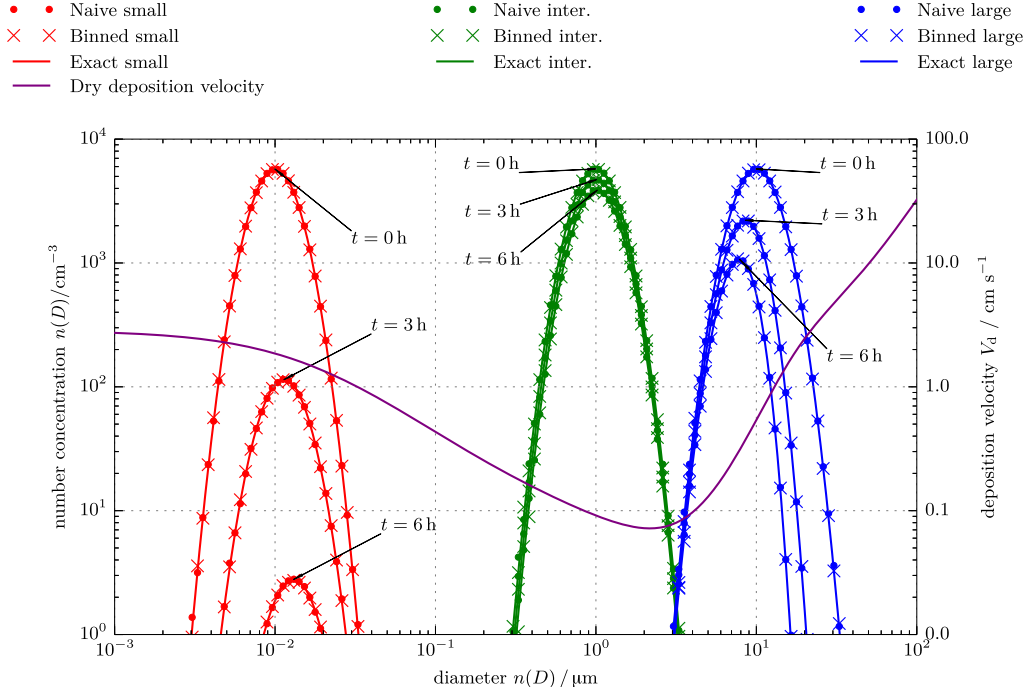


Fig. 2. Comparison of analytical solution to ensemble averaged number distribution $n(D)$ for three different aerosol size modes at $t = 0, 3,$ and 6 h for Naive and Binned Algorithms.

$$\|n(D) - \Phi(D)\|_2 \approx \sqrt{\sum_i^{N_{\text{bin}}} (n_i^j - \Phi_i)^2 \log\left(\frac{D_{i+1}}{D_i}\right)}, \quad (16)$$

where n_i^j is the number concentration density of bin i for run j , n_{bin} is the number of bins, Φ_i is the analytical solution of bin i , and D_i and D_{i+1} are diameter bin edges for bin i . The relative error for run j is computed by

$$e^j = \frac{\|n(D) - \Phi(D)\|_2}{\|\Phi(D)\|_2}, \quad (17)$$

where

$$\|\Phi(D)\|_2 = \sqrt{\int_{-\infty}^{\infty} (\Phi(D))^2 d \log D}. \quad (18)$$

We then calculate the root mean square error (RMSE) of n_{run} independent simulations as

$$\bar{e} = \sqrt{\frac{1}{n_{\text{run}}} \sum_{j=1}^{n_{\text{run}}} e^{j2}}, \quad (19)$$

and the standard deviation in the error σ_e given by

$$\sigma_e = \sqrt{\frac{1}{n_{\text{run}}} \sum_{j=1}^{n_{\text{run}}} (e^j - \bar{e})^2}. \quad (20)$$

Fig. 3 shows the relative mean error \bar{e} and the 95% confidence intervals from 100 ensemble members for each algorithm and mode with varying number of computational particles N_p . For all particle methods, we expect the error to decrease as the number of computational particles N_p increases, with a slope of -0.5 , as the ensemble members are i.i.d. Fig. 3 shows that no additional mean relative error was incurred by selecting the Binned Algorithm over the Naive Algorithm, in keeping with Theorem 2.2.

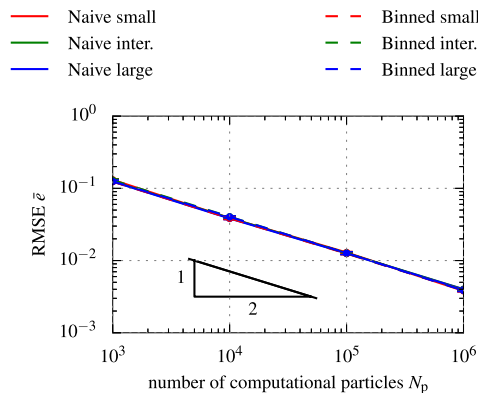


Fig. 3. Root mean square error \bar{e} as expected number of computational particles N_p increases, based on ensemble of 100 runs, for Naive and Binned Algorithms for each of three aerosol modes. Error bars represent 95% confidence intervals for mean error. Expected rate of convergence is shown as slope of -0.5 .

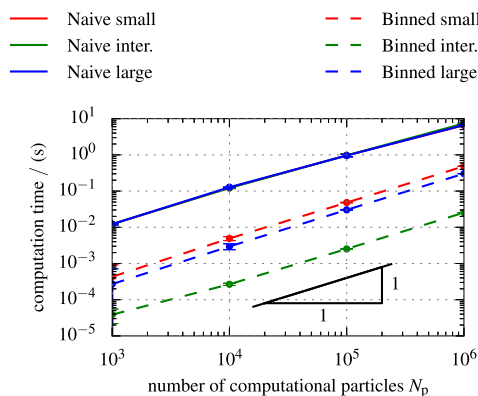


Fig. 4. Comparison of computation time of each of three aerosol modes as number of computational particles N_p increases from 10^3 to 10^6 for Naive Algorithm and Binned Algorithm.

Fig. 4 compares the computation time of the Naive Algorithm and Binned Algorithm for each of the three different particle size modes as a function of increasing number of computational particles N_p from 10^3 to 10^6 for a single averaged time step. As the number of computational particles N_p increases, the computational time increases linearly, as expected. The Naive Algorithm requires that all particles in a simulation are tested for possible removal. As a result, all modes require the same computational time of $O(|\pi|)$. This occurs regardless of how loss rates vary within each of the three modes shown in Fig. 2. By contrast, the Binned Algorithm achieves greater efficiency gains for some modes than for others due to the size dependence of particle loss rates. Here, compared to the Naive Algorithm, the Intermediate mode required nearly three orders of magnitude less computation time. Despite experiencing higher removal rates, both Small and Large mode efficiency still improved by more than an order of magnitude compared to the Naive Algorithm.

Fig. 5 shows the computational time of a time step for the two removal algorithms based on total percentage of particles removed. Again, the Naive Algorithm results in constant computational time regardless of the fraction of particles removed, because the computational cost is proportional to the number of particles in the simulation. In contrast, the Binned Algorithm has a cost that is linear in the fraction of particle removals, as the computational time is designed to be proportional to the number of particle removals. Therefore, the Binned Algorithm becomes particularly desirable for the very low removal rates encountered by most particles in the atmosphere. We observe that even for removal fractions approaching unity the Binned Algorithm is still cheaper than the Naive Algorithm, and numerical extrapolation shows that parity in cost will occur only when almost all particles are removed in a single time step (a physically unrealistic situation).

Fig. 6 shows the root mean square error as a function of computational time for an average single time step for the Binned Algorithm over the Naive Algorithm. These results are based on the three-mode scenario presented in Fig. 2. The Binned Algorithm resulted in a decrease in computational time and contributed no additional error.

To examine the benefits in practice, the Binned Algorithm for removal by dry deposition was applied to a typical scenario with 10^5 computational particles. While gas-aerosol phase chemistry remains the dominant cost, the percentage of computational time determining particle loss decreased from 0.8% to 0.004%. When chemistry was disabled, the particle removal represents a more substantial portion of the total computation time and is reduced from 35% to 0.2%.

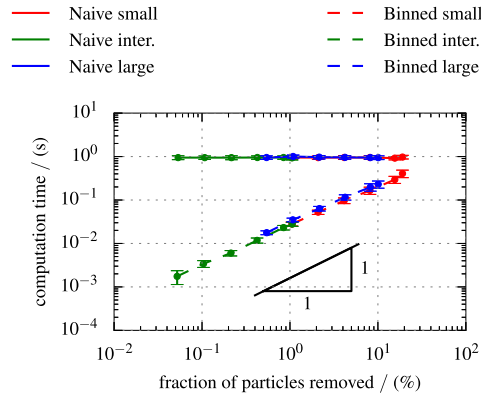


Fig. 5. Comparison of computation time of a single time step of Binned Algorithm and Naive Algorithm for each of three aerosol modes with varied fraction of particle removals. Ensemble size was 100 and computational particles $N_p = 100\,000$ were used for all data points.

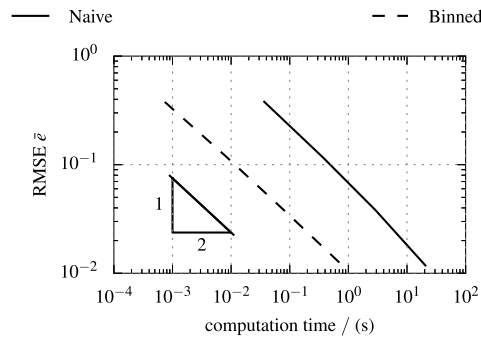


Fig. 6. Root mean square error as a function of computation time for Binned and Naive Algorithms. Ensemble size was 100. We observe speedup of about 50 times with no increase in error.

5. Code availability

The source code for the PartMC model is available from <http://lagrange.mechse.illinois.edu/mwest/partmc/>. The Naive Algorithm and the Binned Algorithm described in this manuscript were implemented in PartMC version 2.3.0.

6. Conclusions

We introduced the Binned Algorithm (Algorithm 2) for efficiently sampling single-particle events in particle simulations. We proved that this algorithm always generates exact samples of the stochastic process (Corollary 2.3) and that it achieves perfect sampling efficiency in the limit of perfect kernel estimates (Theorem 2.4).

As shown in Fig. 6, the Binned Algorithm for particle removals shows about a 50-fold reduction in computational time on atmospherically relevant applications while contributing no additional error. When applied to dry deposition, the benefits are most substantial for particles in the size range of 50 nm to 1 μm , since these particles have the lowest removal probabilities. While efficiency gains will be most noticeable for studies in the absence of gas-particle chemistry, benefits apply for all model runs given that the Binned Algorithm presented here provides pure efficiency gains where no additional error is incurred.

The Binned Algorithm may be also applied to other single-particle removal processes. For example, laboratory experiments with aerosols in a chamber environment include two loss processes, namely wall loss due to particle diffusion and sedimentation, where both processes have a dependence on size [40]. Within particle-resolved 1D and 3D atmospheric models, gravitational settling may be included in situations where particles are large enough to have non-negligible settling velocities [41]. The simulation of the fragmentation of soot particles may be represented with a dependence on particle size [42]. For processes that depend on additional variables that have only a weak influence on probabilities, this can be accounted for by adjusted upper bounds on probabilities. For example, density is a weakly dependent variable for dry deposition and is accounted for by overestimating particle density for computation of upper bounds, as described in Section 3. For processes depending strongly on multiple variables, multidimensional binning may be required and remains computationally feasible if the number of dimensions remains small. For example, 2D binning for coagulation was used in [22] where the process depends strongly on the masses of the two coagulating particles. By applying the Binned Algorithm, the computationally costly iteration over all particles can be avoided in all of these per-particle processes.

Acknowledgements

NR and JHC acknowledge funding from the U.S. Department of Energy, Office of Science, Biological and Environmental Research as part of the Atmospheric System Research program under grant DE-SC0011771. MDM acknowledges support from EPA RD-83504201. MW acknowledges support from NSF CMMI 11-50490. The contents of this work are solely the responsibility of the grantees and do not necessarily represent the official views of the US EPA. Further, US EPA does not endorse the purchase of any commercial products or services mentioned in this publication.

Appendix A. Computing particle dry deposition velocity

The PartMC determination of dry deposition velocity and dry removal rate follows closely from the size-segregated particle dry deposition scheme presented in [38]. Dry deposition velocity of a particle V_d can be expressed as

$$V_d = V_s + \frac{1}{r_a + r_s}, \quad (\text{A.1})$$

where V_s is the gravitational settling velocity, r_a is the aerodynamic resistance and r_s is the surface resistance. Gravitational settling velocity is calculated as

$$V_s = \frac{\rho d_p^2 g C_c}{18\mu}, \quad (\text{A.2})$$

where ρ is the density of the particle, d_p is the particle diameter (used in place of D in this appendix to differentiate from Brownian diffusion), g is the acceleration due to gravity, C_c is the Cunningham slip correction factor, and μ is the dynamic viscosity of air. The Cunningham slip-flow correction factor to account for non-continuum

$$C_c = 1 + \frac{2\lambda}{d_p} \left(1.257 + 0.4 \exp\left(-0.55 \frac{d_p}{\lambda}\right) \right), \quad (\text{A.3})$$

where λ is the mean free path of air molecules given by

$$\lambda = \frac{2\mu}{\rho_a \bar{v}_a}, \quad (\text{A.4})$$

and ρ_a is the density of air, μ is the dynamic viscosity and \bar{v}_a is the mean thermal speed of an air molecule. From [41], the dynamic viscosity μ is calculated as

$$\mu = 1.8325 \times 10^{-5} \left(\frac{416.16}{T + 120} \right) \left(\frac{T}{296.16} \right)^{1.5}, \quad (\text{A.5})$$

where T is temperature.

To determine the aerodynamic resistance r_a , we specify a mean wind speed \bar{u} at a reference height z_{ref} and a stability class. For a neutral stability atmosphere, aerodynamic resistance r_a is given by

$$r_a = \frac{1}{\kappa u_*} \ln\left(\frac{z}{z_0}\right), \quad (\text{A.6})$$

where $\kappa = 0.4$ is the von Karman constant and u_* is the frictional velocity. Frictional velocity, under neutral stability conditions, can be expressed as

$$u_* = \frac{\kappa \bar{u}(z_{\text{ref}})}{\ln(z_{\text{ref}}/z_0)}. \quad (\text{A.7})$$

The surface resistance r_s is computed as

$$r_s = \frac{1}{\epsilon_0 u_* (E_B + E_{\text{IM}} + E_{\text{IN}}) R_1}, \quad (\text{A.8})$$

where E_B is collection efficiency from Brownian diffusion, E_{IM} is collection efficiency from impaction and E_{IN} is collection efficiency from interception. ϵ_0 is an empirical constant, taken to have a value of 3. R_1 is a correction factor for particle for large particles and is parameterized as a function of particle size.

The collection efficiency due to Brownian diffusion E_B is given by

$$E_B = Sc^{-\gamma}, \quad (\text{A.9})$$

where Sc is the Schmidt number and γ is a constant that depends on surface type. The Schmidt number Sc is defined as

$$Sc = \frac{\nu}{D}, \quad (\text{A.10})$$

where ν is the kinematic viscosity of air given by

$$\nu = \frac{\mu}{\rho_a}, \quad (\text{A.11})$$

for density of air ρ_a and D is Brownian diffusivity of the particle given by

$$D = \frac{kTC_c}{3\pi\mu d_p}, \quad (\text{A.12})$$

where k is Boltzmann's constant. The collection efficiency due to particle impaction with the surface E_{IM} is given by

$$E_{IM} = \left(\frac{St}{\alpha + St} \right)^\beta, \quad (\text{A.13})$$

where α and β are constants and St is the Stokes number, which depends on surface type. The Stokes number for smooth and vegetative surfaces, respectively, is given by

$$St = \frac{V_s u_*^2}{g\nu} \quad \text{for smooth} \quad (\text{A.14})$$

$$St = \frac{V_s u_*^2}{gA} \quad \text{for vegetative,} \quad (\text{A.15})$$

where A is the characteristic radius for different land use and seasonal categories (given by Table 3 in [38]). The collection efficiency due to particle interception with the surface E_{IN} is given by

$$E_{IN} = \frac{1}{2} \left(\frac{d_p}{A} \right)^2. \quad (\text{A.16})$$

Large particles may rebound after hitting the surface and this is accounted for by modifying the collection efficiency by the factor R_1 which has the form

$$R_1 = \exp(-St^{1/2}). \quad (\text{A.17})$$

Per-particle dry deposition velocities may be calculated by application of these equations. Given a dry deposition velocity, the particle loss rate K is given by

$$K = \frac{V_d}{\Delta z}, \quad (\text{A.18})$$

where Δz is a reference or lowest grid cell height.

Appendix B. Computing exact solution

The exact solution of the number concentration density $n(D, t)$ at time t can be represented as an exponential decay proportional to the dry deposition rate at diameter D and is given by

$$n(D, t) = n_0(D) \exp\left(-\frac{V_d(D)}{\Delta z} t\right), \quad (\text{B.1})$$

where $n_0(D)$ is the initial number distribution and $V_d(D)$ is the dry deposition velocity at diameter D .

To replicate the post-process binning of particle-resolved simulations, the analytical solution of Equation (B.1) is evaluated on a high-resolution diameter grid \hat{D} with $n_{\text{bin}} = 1500$ logarithmically-spaced from $0.001 \mu\text{m}$ to $100 \mu\text{m}$. The high-resolution number concentration density $n(\hat{D})$ is then averaged to the particle-resolved post-processing grid of 150 bins. As a result, the analytical value of the bin i from D_i to D_{i+1} is

$$\Phi_i = \frac{1}{10} \sum_{j=10(i-1)}^{10i} n(\hat{D}_j). \quad (\text{B.2})$$

References

- [1] S.-I. Shima, K. Kusano, A. Kawano, T. Sugiyama, 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 (642) (2009) 1307–1320.
- [2] S. Arabas, S.-I. Shima, Large-eddy simulations of trade wind cumuli using particle-based microphysics with Monte Carlo coalescence, *J. Atmos. Sci.* 70 (9) (2013) 2768–2777.
- [3] I. Sölch, B. Kärcher, A large-eddy model for cirrus clouds with explicit aerosol and ice microphysics and Lagrangian ice particle tracking, *Q. J. R. Meteorol. Soc.* 136 (653) (2010) 2074–2093.
- [4] V. Saliakas, C. Kotoulas, D. Meimaroglou, C. Kiparissides, Dynamic evolution of the particle size distribution in suspension polymerization reactors: a comparative study on Monte Carlo and sectional grid methods, *Can. J. Chem. Eng.* 86 (5) (2008) 924–936.
- [5] J.R. van Peborgh Gooch, M.J. Hounslow, Monte Carlo simulation of size-enlargement mechanisms in crystallization, *AIChE J.* 42 (7) (1996) 1864–1874.
- [6] E.L. Haseltine, D.B. Patience, J.B. Rawlings, On the stochastic simulation of particulate systems, *Chem. Eng. Sci.* 60 (10) (2005) 2627–2641.
- [7] Y. Efendiev, M. Zachariah, Hybrid Monte Carlo method for simulation of two-component aerosol coagulation and phase segregation, *J. Colloid Interface Sci.* 249 (1) (2002) 30–43.
- [8] E. Debry, B. Sportisse, B. Jourdain, A stochastic approach for the numerical simulation of the general dynamics equation for aerosols, *J. Comput. Phys.* 184 (2) (2003) 649–669.
- [9] A. Maisels, F. Einar Kruijs, H. Fissan, Direct simulation Monte Carlo for simultaneous nucleation, coagulation, and surface growth in dispersed systems, *Chem. Eng. Sci.* 59 (11) (2004) 2231–2239.
- [10] N. Riemer, M. West, R.A. Zaveri, R.C. Easter, Simulating the evolution of soot mixing state with a particle-resolved aerosol model, *J. Geophys. Res.* 114 (2009) D09202, <http://dx.doi.org/10.1029/2008JD011073>.
- [11] Z. Xu, H. Zhao, C. Zheng, Fast Monte Carlo simulation for particle coagulation in population balance, *J. Aerosol Sci.* 74 (2014) 11–25.
- [12] M. Balthasar, M. Kraft, A stochastic approach to calculate the particle size distribution function of soot particles in laminar premixed flames, *Combust. Flame* 133 (3) (2003) 289–298.
- [13] K.F. Lee, R.I. Patterson, W. Wagner, M. Kraft, Stochastic weighted particle methods for population balance equations with coagulation, fragmentation and spatial inhomogeneity, *J. Comput. Phys.* 303 (2015) 1–18, <http://dx.doi.org/10.1016/j.jcp.2015.09.031>.
- [14] A. Violi, Modeling of soot particle inception in aromatic and aliphatic premixed flames, *Combust. Flame* 139 (4) (2004) 279–287.
- [15] M. Celnik, R. Patterson, M. Kraft, W. Wagner, Coupling a stochastic soot population balance to gas-phase chemistry using operator splitting, *Combust. Flame* 148 (3) (2007) 158–176.
- [16] D.M. Murphy, D.J. Cziczko, K.D. Froyd, P.K. Hudson, B.M. Matthew, A.M. Middlebrook, R.E. Peltier, A. Sullivan, D.S. Thomson, R.J. Weber, Single-particle mass spectrometry of tropospheric aerosol particles, *J. Geophys. Res.* 111 (2006) D23532, <http://dx.doi.org/10.1029/2006JD007340>.
- [17] K.A. Pratt, K.A. Prather, Real-time, single-particle volatility, size, and chemical composition measurements of aged urban aerosols, *Environ. Sci. Technol.* 43 (21) (2009) 8276–8282.
- [18] A. Zelenyuk, D. Imre, Beyond single particle mass spectrometry: multidimensional characterisation of individual aerosol particles, *Int. Rev. Phys. Chem.* 28 (2) (2009) 309–358, <http://dx.doi.org/10.1080/01442350903037458>.
- [19] J.L. Jimenez, J.T. Jayne, Q. Shi, C.E. Kolb, D.R. Worsnop, I. Yourshaw, J.H. Seinfeld, R.C. Flagan, X. Zhang, K.A. Smith, J.W. Morris, P. Davidovits, Ambient aerosol sampling using the aerodyne aerosol mass spectrometer, *J. Geophys. Res.* 108 (2003) 8425, <http://dx.doi.org/10.1029/2001JD001213>.
- [20] R.A. Zaveri, R.C. Easter, J.D. Fast, L.K. Peters, Model for simulating aerosol interactions and chemistry (MOSAIC), *J. Geophys. Res.* 113 (2008) D13204, <http://dx.doi.org/10.1029/2007JD008782>.
- [21] R.E.L. DeVilje, N. Riemer, M. West, The weighted flow algorithm (WFA) for stochastic particle coagulation, *J. Comput. Phys.* 230 (23) (2011) 8427–8451, <http://dx.doi.org/10.1016/j.jcp.2011.07.027>.
- [22] M.D. Michelotti, M.T. Heath, M. West, Binning for efficient stochastic multiscale particle simulations, *Multiscale Model. Simul.* 11 (2013) 1071–1096, <http://dx.doi.org/10.1137/130908038>.
- [23] R.I. Patterson, W. Wagner, M. Kraft, Stochastic weighted particle methods for population balance equations, *J. Comput. Phys.* 230 (19) (2011) 7456–7472, <http://dx.doi.org/10.1016/j.jcp.2011.06.011>.
- [24] D.T. Gillespie, An exact method for numerically simulating the stochastic coalescence process in a cloud, *J. Atmos. Sci.* 32 (10) (1975) 1977–1989.
- [25] D.T. Gillespie, Exact stochastic simulation of coupled chemical reactions, *J. Phys. Chem.* 81 (25) (1977) 2340–2361.
- [26] A. Eibeck, W. Wagner, An efficient stochastic algorithm for studying coagulation dynamics and gelation phenomena, *SIAM J. Sci. Comput.* 22 (3) (2000) 802–821.
- [27] M.A. Gibson, J. Bruck, Efficient exact stochastic simulation of chemical systems with many species and many channels, *J. Phys. Chem. A* 104 (9) (2000) 1876–1889.
- [28] H. Kuwahara, I. Mura, An efficient and exact stochastic simulation method to analyze rare events in biochemical systems, *J. Chem. Phys.* 129 (2008) 165101.
- [29] D.T. Gillespie, Approximate accelerated stochastic simulation of chemically reacting systems, *J. Chem. Phys.* 115 (4) (2001) 1716–1733.
- [30] M. Rathinam, L.R. Petzold, Y. Cao, D.T. Gillespie, Stiffness in stochastic chemically reacting systems: the implicit tau-leaping method, *J. Chem. Phys.* 119 (2003) 12784.
- [31] D.T. Gillespie, L.R. Petzold, Improved leap-size selection for accelerated stochastic simulation, *J. Chem. Phys.* 119 (2003) 8229.
- [32] D.T. Gillespie, M. Roh, L.R. Petzold, Refining the weighted stochastic simulation algorithm, *J. Chem. Phys.* 130 (2009) 174103.
- [33] S. Wu, J. Fu, L.R. Petzold, Adaptive deployment of model reductions for tau-leaping simulation, *J. Chem. Phys.* 142 (20) (2015) 204108, <http://dx.doi.org/10.1063/1.4921638>.
- [34] J. Fu, S. Wu, L.R. Petzold, Time dependent solution for acceleration of tau-leaping, *J. Comput. Phys.* 235 (2013) 446–457, <http://dx.doi.org/10.1016/j.jcp.2012.10.036>.
- [35] N. Riemer, M. West, R.A. Zaveri, R.C. Easter, Estimating black carbon aging time-scales with a particle-resolved aerosol model, *J. Aerosol Sci.* 41 (2010) 143–158, <http://dx.doi.org/10.1016/j.jaerosci.2009.08.009>.
- [36] R.A. Zaveri, J.C. Barnard, R.C. Easter, N. Riemer, M. West, Particle-resolved simulation of aerosol size, composition, mixing state, and the associated optical and cloud condensation nuclei activation properties in an evolving urban plume, *J. Geophys. Res.* 115 (2010) D17210, <http://dx.doi.org/10.1029/2009JD013616>.
- [37] D.E. Knuth, *The Art of Computer Programming*, vol. 2: Seminumerical Algorithms, 3rd edition, Addison–Wesley, 1998, p. 473.
- [38] L. Zhang, S. Gong, J. Padro, L. Barrie, A size-segregated particle dry deposition scheme for an atmospheric aerosol module, *Atmos. Environ.* 35 (3) (2001) 549–560, [http://dx.doi.org/10.1016/S1352-2310\(00\)00326-5](http://dx.doi.org/10.1016/S1352-2310(00)00326-5).
- [39] F. Pasquill, The estimation of the dispersion of windborne material, *Meteorol. Mag.* 90 (1063) (1961) 33–49.
- [40] K.-H. Naumann, COSIMA — a computer program simulating the dynamics of fractal aerosols, *J. Aerosol Sci.* 34 (10) (2003) 1371–1397.
- [41] M.Z. Jacobson, *Fundamentals of Atmospheric Modeling*, Cambridge University Press, 2005.
- [42] S.J. Harris, M.M. Maricq, The role of fragmentation in defining the signature size distribution of diesel soot, *J. Aerosol Sci.* 33 (6) (2002) 935–942.