Weierstraß-Institut
für Angewandte Analysis und Stochastik
Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 0946 – 8633

A stochastic weighted particle method for coagulation–advection problems

Robert I. A. Patterson[1], Wolfgang Wagner[1]

submitted: 5. September 2011

[1] Weierstraß–Institut
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: Robert.Patterson@wias-berlin.de
Wolfgang.Wagner@wias-berlin.de

No. 1641
Berlin 2011

2010 *Mathematics Subject Classification.* 60J25, 65C05, 65C35, 82C22.

Key words and phrases. coagulation, advection, stochastic particle, instability.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 2044975
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

Abstract

A spatially resolved stochastic weighted particle method for inception–coagulation–advection problems is presented. Convergence to a deterministic limit is briefly studied. Numerical experiments are carried out for two problems with very different coagulation kernels. These tests show the method to be robust and confirm the convergence properties. The robustness of the weighted particle method is shown to contrast with two Direct Simulation Algorithms which develop instabilities.

1 Introduction

Stochastic simulation of particle formation and growth has been successfully used in a wide range of detailed particle modelling in fields including soot [25], metal oxide particles [7], snowflakes and their structure [18]. It can also be used for more abstract investigations such as the study of mixture distributions in multicomponent systems [20].

In this work population balance problems representing particles undergoing inception, coagulation and advection are considered. Particles take positions in a bounded domain, $\mathcal{X}$ of a finite dimensional Euclidean space and the physical properties of the particles are described by their 'type'. A very simple type is the particle mass, but in applications more detailed particle properties such as chemical composition, charge and aggregate structure can be of interest [25, 15, 17, 27] and so the type space $\mathcal{Z}$ may be multi-dimensional and either discrete or continuous. The problem balance problem is formulated for the density of the population distribution c, where

$$\int_{U\times A} c(t,x,z)\mathrm{d}x\mathrm{d}z \tag{1}$$

is the number of particles with types in $A \subset \mathcal{Z}$ that are located in $U \subset \mathcal{X}$ at time t. Throughout this work, differential operators are taken to operate only on the spatial position argument of c and integration $\mathrm{d}z$ over $\mathcal{Z}$ is taken to be Lebesgue measure or counting measure (when $\mathcal{Z}$ is discrete). The population balance problem with advection is then

$$\begin{aligned}\frac{\partial}{\partial t}c(t,x,z) + u(x)\nabla_x c(t,x,z) = I(x,z)&\\ + \frac{1}{2}\int_{\substack{z_1,z_2\in\mathcal{Z}\\ z_1+z_2=z}} K(z_1,z_2)c(t,x,z_1)c(t,x,z_2)\mathrm{d}z_1\mathrm{d}z_2&\\ - \int_{z_1\in\mathcal{Z}} K(z_1,z)c(t,x,z_1)c(t,x,z)\mathrm{d}z_1,&\end{aligned} \tag{2}$$

where K is the coagulation kernel, u is the advection velocity and I is the inception rate. Because (2) is a first order equation, it is sufficient to specify just one boundary condition for the region of the boundary, Γ_{in} where the flow crosses into the domain ($u \cdot \nu < 0$, where ν is the outward normal)

$$c(t, x, z) = c_{\text{in}}(t, x, z) \quad x \in \Gamma_{\text{in}}. \tag{3}$$

An initial condition, which for simplicity is taken as 0 is also required, but should not have any influence on the solution behaviour for large t.

Systems of practical interest such as soot formation during combustion, often lead to spatially inhomogeneous problems that have to be coupled to spatially resolved flow calculations. However, existing stochastic particle methods for population balance problems are mostly limited to to problems that can be formulated in a spatially homogeneous way, possibly with time varying coefficients [25, 7, 20, 18]. One way to bridge the gap between homogeneous and inhomogeneous methods is to precalculate the flow, possibly including a less sophisticated method to approximate the effects of the particles and then to carry out stochastic simulations inside a small volume moving along a stream line as a form of post-processing to obtain full particle distributions from spatially homogeneous simulations [1, 11, 16]. However, such a post-processing approach means that the detailed particle properties, which would only be studied if they were important, are not included in the underlying fluid dynamics and chemical reaction calculations, on which they can have very significant effects [5].

The present work appears to be new in using stochastic particle methods to numerically treat inhomogeneous coagulation–advection problems. A similar problem was studied by Rudnicki and Wieczorek [24] using direct simulation methods, but with a 'coagulation' rate that was linear in the particle concentration and so fundamentally different to the Smoluchowski coagulation equation considered here, in which coagulation is quadratic in particle concentration. Some previous work with theoretical and computational aspects has been done on Markov processes to simulate coagulation–diffusion problems [10, 12, 13]. Such problems have a rather different structure and the work focused on closed systems where mass could neither enter or leave the system. Many engineering problems on the other hand tend to involve continuous inflow and outflow, otherwise product or pollutant never interacts with the rest of the world. Simulations of particle formation in flames, which were the trigger for the present work show both of these features—mass transfer from the gas to the solid phase and outflow at the end of the flame.

Stochastic simulation methods for the Boltzmann equations for gas molecules also progressed from homogeneous problems [3, 28, 19] to very complex inhomogeneous domains [4] where a time independent boundary value problem solution is found by simulating the time dependent system for a period of time that is large enough for the initial transients to become negligible. The same strategy will be used in the present work, however there is one very significant difference between the Boltzmann and Smoluchowski (coagulation) situations: Boltzmann collisions exchange momentum but preserve the number of particles, whereas coagulations reduce the number of particles. This leads to some interesting numerical effects.

Applied computations that attempt to capture over a spatial domain the full particle distribution rather than just a few integral quantities, have been carried out in a small number of cases. In addition to [11] referred to above a finite element method was used in a post-processing step after 3-d flow equations had been solved [14] for a crystallisation problem with one internal coordinate. A simple sectional approximation with one internal coordinate was used by Sun et al. [26] to simulate soot formation and a more complex sectional approach was later used by Zhang et al. [29]. Additional examples, including other application areas, will become more widespread as the computational requirements become more modest in relation to the available hardware.

The remainder of the paper is structured as follows: In §2 the simulation processes are defined and their convergence is discussed. The numerical properties of the algorithms are explored in §3.1 for a problem that uses the constant coagulation kernel. A much more challenging problem is used for further tests in §3.2 and provides examples of how direct simulation methods can become unstable. Finally, conclusions and pointers for future work are set out in §4.

2 Stochastic Particle Methods

The basic idea is shared with the long established Bird algorithm for the Boltzmann equation [2]: use an operator splitting to simulated the free streaming (advection) part of (2) separately from the source terms on the right hand size (coagulation and inception). These source terms are then stochastically simulated with a population of particles divided into isolated spatial cells. Within such cells coagulation is treated as being independent of the distances between pairs of particles. Particle positions are not restricted to a discrete lattice (such as the cell centres), but are continuous variables, in order to avoid advection becoming numerically diffusive.

The algorithm is valid for position space of any dimension so for simplicity it is presented here (and later tested) for the case where $\mathcal{X} = [0, L]$; this domain $[0, L]$ is divided into n_{cell} cells

$$\mathcal{X}_i = [iL/n_{\mathrm{cell}}, (i+1)L/n_{\mathrm{cell}}), \quad i = 0, \ldots, n_{\mathrm{cell}} - 1, \tag{4}$$

with the understanding that the final cell includes its right hand end point. Write $\Delta x_{\mathrm{cell}} = L/n_{\mathrm{cell}}$ for the length of each cell and $|\mathcal{X}_i|$ for the cell volume, which in this 1-dimensional case is the same as its length. Let $\bar{x}_i$ $i = 0, \ldots, n_{\mathrm{cell}} - 1$ be the cell mid-points.

Time splitting is carried out over intervals $[j\Delta t_{\mathrm{split}}, (j+1)\Delta t_{\mathrm{split}}]$ so that one first simulates the source terms from the right hand side of (2) for each cell separately and secondly updates the position of each particle with its advective displacement over the same time interval. This implies that coagulation is delocalised in that particles that have different positions have a positive probability of coagulating with each other, but the delocalisation is restricted to be within one cell.

Throughout this work it is assumed that c exists with adequate regularity properties and is unique. Integrating against continuous compactly supported test functions ϕ gives the following weak form, which is used in studying the source terms

$$\frac{\partial}{\partial t}\int_{\mathcal{X}\times\mathcal{Z}}\phi(x,z)c(t,x,z)\mathrm{d}x\mathrm{d}z+u\int_{\mathcal{X}\times\mathcal{Z}}\phi(x,z)\nabla_x c(t,x,z)\mathrm{d}x\mathrm{d}z=$$
$$\int_{\mathcal{X}\times\mathcal{Z}}\phi(x,z)I(x,z)\mathrm{d}x\mathrm{d}z$$
$$+\frac{1}{2}\int_{\mathcal{X}\times\mathcal{Z}^2}\left[\phi(x,z_1+z_2)-\phi(x,z_1)-\phi(x,z_2)\right]$$
$$K(z_1,z_2)c(t,x,z_1)c(t,x,z_2)\mathrm{d}x\mathrm{d}z_1\mathrm{d}z_2. \quad (5)$$

For each splitting time interval, the physical particle population dynamics (2) are simulated with a system of stochastic particles parameterised by $n\in\mathbb{N}$:

$$(x_{j,n}(t),z_{j,n}(t),w_{j,n}(t)) \quad j=1,\dots,N_n(t) \quad (6)$$

where $x_{j,n}(t)\in[0,L]$, $z_{j,n}(t)\in\mathcal{Z}$ and $w_{j,n}(t)\in(0,w_{\max}]$. It will eventually be seen that $w_{j,n}/nV_1$ (the choice of $V_1>0$, which simply rescales n is discussed below) can be interpreted as the number of physical particles per unit volume represented by the j-th computational particle. During each splitting step particles remain within the same $\mathcal{X}_i$ and can only interact with other particles also in $\mathcal{X}_i$. Conditional on the initial conditions at the start of the splitting step, the particle processes in each cell are independent.

The density c is approximated by a sum of delta functions, the locations of which (in $[0,L]\times\mathcal{Z}$) are the stochastic particles [9, 23]. The approximation works in the sense that

$$\lim_{n\to\infty}\mathbb{E}\left[\frac{1}{nV_1}\sum_{j=1}^{N_n(t)}w_{j,n}(t)\phi\left(x_{j,n}(t),z_{j,n}(t)\right)\mathbb{1}\left\{x_{j,n}(t)\in\mathcal{X}_i\right\}\right]$$
$$=\int_{\mathcal{X}_i\times\mathcal{Z}}\phi(x,z)c(t,x,z)\mathrm{d}x\mathrm{d}z+\mathrm{Err}\left(\Delta x_{\mathrm{cell}},\Delta t_{\mathrm{split}}\right)$$
$$\forall t,\ i=1,\dots,n_{\mathrm{cell}}, \quad (7)$$

where $\mathrm{Err}\left(\Delta x_{\mathrm{cell}},\Delta t_{\mathrm{split}}\right)$ is the discretisation and splitting error, which is a deterministic quantity.

The role of n is to control the computational quality and cost of the stochastic particle method. In the limit $n\to\infty$ one expects the particle system to converge to a deterministic limit so that (7) holds even without taking the expectation, although it must be noted that the n limit does not remove the errors due to Δx_{cell} and $\Delta t_{\mathrm{split}}$. Computations necessarily use finite values of n, but larger values reduce systematic and statistical error, for a computational price. Examination of the equations below shows that n multiplies the rate at which computational particles are created (10) and divides the rate at which they are destroyed (12); it can thus be used to approximately select the number of computational particles and in this way to control computational quality and cost. An additional scaling factor for n, $V_1>0$ is introduced for each algorithm below so that n can be restricted to $\mathbb{N}$. For both problems considered V_1 was chosen so that

$$\mathbb{E}\left[\sum_{j=1}^{N_n(t)} \mathbb{1}\left\{x_{j,n}(t) \in \mathcal{X}_i\right\}\right] < \frac{3}{4}n \quad i = 1, \ldots, n_{\text{cell}}. \tag{8}$$

This is a bound on the mean number of computational particles per cell; the factor of $\frac{3}{4}$ was convenient for software reasons.

There are two main methods for the simulation of the stochastic particle processes within homogeneous cells: the Direct Simulation Algorithm (DSA) and the Stochastic Weighted Algorithms (SWA).

2.1 Direct Simulation Algorithms

Direct simulation is so named because it can helpfully be interpreted as directly reproducing the behaviour of small, but representative portion of the physical system. An additional approximation is inherent in direct simulation: correlations between particles are ignored. Direct simulation is the case when $w_{j,n}(t) \equiv 1$.

2.1.1 Inception

A new computational particle

$$(x, z, 1) \tag{9}$$

is added to cell i (midpoint $\bar{x}_i$) at rate

$$nV_1 \int_{\mathcal{Z}} I\left(\bar{x}_i, z\right) \mathrm{d}z \approx \frac{nV_1}{|\mathcal{X}_i|} \int_{\mathcal{X}_i \times \mathcal{Z}} I(x, z) \mathrm{d}x \mathrm{d}z \tag{10}$$

with x chosen uniformly in $\mathcal{X}_i$ and z chosen according to the density $I\left(\bar{x}_i, z\right)$. Recall that $\mathrm{d}z$ should be understood as Lebesgue measure unless $\mathcal{Z}$ is discrete, in which case it is counting measure.

2.1.2 Coagulation

Coagulation jumps reduce the number of computational particles by 1 and are of the form

$$\left(x_{j_1,n}, z_{j_1,n}, 1\right), \ \left(x_{j_2,n}, z_{j_2,n}, 1\right) \to \left(\tilde{x}, z_{j_1,n} + z_{j_2,n}, 1\right) \tag{11}$$

where $\tilde{x}$, similarly to [24], is distributed according to $S(x_{j_1,n}, z_{j_1,n}, x_{j_2,n}, z_{j_2,n}, \mathrm{d}\tilde{x})$ independently of all other parts of the Markov process and each jump of the form (11) occurs within the cell $\mathcal{X}_i$ at rate

$$\frac{K\left(z_{j_1,n}, z_{j_2,n}\right)}{nV_1} \mathbb{1}\left\{x_{j_1,n} \in \mathcal{X}_i\right\} \mathbb{1}\left\{x_{j_2,n} \in \mathcal{X}_i\right\}. \tag{12}$$

A number of definitions of $S(x_1, z_1, x_2, z_2, \mathrm{d}\tilde{x})$ may seem plausible: One can try to use the midpoint

$$S(x_1, z_1, x_2, z_2, \mathrm{d}\tilde{x}) := \delta_{(x_1+x_2)/2}(\mathrm{d}\tilde{x}), \tag{13}$$

but for numerical purposes ($\Delta x_{\text{cell}} > 0$) this gradually concentrates the particle distribution at the cell centre. A second systematic error arises since the simulation of the streaming step moves particles towards the right hand end of a cell and then coagulation with newly incepted particles in the simulation of the source terms moves particles back towards the centre of the cell so that their apparent velocity is less than u. A more reasonable definition takes the form

$$S(x_1, z_1, x_2, z_2, \mathrm{d}\tilde{x}) := \\ p(x_1, z_1, x_2, x_2)\delta_{x_1}(\mathrm{d}\tilde{x}) + (1 - p(x_1, z_1, x_2, x_2))\,\delta_{x_2}(\mathrm{d}\tilde{x}). \quad (14)$$

The definitions $p \equiv 1$ and $p \equiv \frac{1}{2}$ are equivalent, because all ordered pairs of indices have to be considered for coagulation and K is symmetric. In this work the form $p \equiv \frac{1}{2}$ is preferred, because it explicitly preserves the symmetry of the coagulation events. This definition of S and p the resulting algorithm is denoted “DSA1”.

One can choose to preserve, in mean, the centre of mass of the two particles by setting

$$p(x_1, z_1, x_2, x_2) = \frac{\bar{m}(z_1)}{\bar{m}(z_1) + \bar{m}(z_2)}, \quad (15)$$

where $\bar{m}(z)$ is the mass of a particle of type $z \in \mathcal{Z}$. This choice will be denoted “DSA2”.

2.1.3 Limiting Equation

The convergence of the DSA methods and a weak differential equation for their limit within each splitting step is discussed in §2.3.1. Using the additional assumption that the limiting distribution has a density $\tilde{c}$ on $\mathcal{X} \times \mathcal{Z}$ (with respect to Lebesgue measure on $\mathcal{X}$ and Lebesgue or counting measure on $\mathcal{Z}$ according to whether $\mathcal{Z}$ is discrete or not) then the convergence results imply the following equations for $\tilde{c}$ for $i = 1, \ldots, n_{\text{cell}}$:

For DSA1 one has

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathcal{X}_i \times \mathcal{Z}} \phi(x, z)\tilde{c}(t, x, z)\mathrm{d}x\mathrm{d}z = \int_{\mathcal{X}_i \times \mathcal{Z}} \phi(x, z) I(\bar{x}_i, z)\mathrm{d}x\mathrm{d}z \\ + \frac{1}{2\,|\mathcal{X}_i|} \int_{(\mathcal{X}_i \times \mathcal{Z})^2} [\phi(x_1, z_1 + z_2) - \phi(x_1, z_1) - \phi(x_2, z_2)] \\ K(z_1, z_2)\tilde{c}(t, x_1, z_1)\tilde{c}(t, x_2, z_2)\mathrm{d}x_1\mathrm{d}z_1\mathrm{d}x_2\mathrm{d}z_2 \quad (16)$$

and for DSA2

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathcal{X}_i \times \mathcal{Z}} \phi(x, z)\tilde{c}(t, x, z)\mathrm{d}x\mathrm{d}z = \int_{\mathcal{X}_i \times \mathcal{Z}} \phi(x, z) I(\bar{x}_i, z)\mathrm{d}x\mathrm{d}z \\ + \frac{1}{|\mathcal{X}_i|} \int_{(\mathcal{X}_i \times \mathcal{Z})^2} \left[\frac{\bar{m}(z_1)}{\bar{m}(z_1) + \bar{m}(z_2)} \phi(x_1, z_1 + z_2) - \phi(x_1, z_1)\right] \\ K(z_1, z_2)\tilde{c}(t, x_1, z_1)\tilde{c}(t, x_2, z_2)\mathrm{d}x_1\mathrm{d}z_1\mathrm{d}x_2\mathrm{d}z_2. \quad (17)$$

These equations deal with the particle population during the stochastic simulation of the source terms from the right hand side of (5) during one splitting step. Advection does not feature because this is the other part of the splitting.

For finite Δx_{cell} (16) and (17) are distinct, but formally letting $\Delta x_{\text{cell}} \to 0$ causes the right hand side of both equations to collapse onto the right hand side of (5).

2.2 Stochastic Weighted Algorithm

Stochastic weighted particle algorithms for spatially homogeneous coagulation problems are presented in [23] based on the Mass Flow Algorithm of [8]. Weights are no longer identically 1, but the algorithms are constructed such that

$$w_{j,n}(t) \in (0, w_{\max}] \quad \forall j \leq N_n(t). \tag{18}$$

2.2.1 Inception

A new computational particle

$$(x, z, w_{\text{incep}}) \tag{19}$$

is added to cell i (midpoint $\bar{x}_i$) at rate

$$\frac{nV_1}{w_{\text{incep}}} \int_{\mathcal{Z}} I\left(\bar{x}_i, z\right) \mathrm{d}z \approx \frac{nV_1}{w_{\text{incep}} \Delta x_{\text{cell}}} \int_{\mathcal{X}_i \times \mathcal{Z}} I(x, z) \mathrm{d}x\mathrm{d}z \tag{20}$$

with x chosen uniformly in $\mathcal{X}_i$ and z chosen according to the density $I\left(\bar{x}_i, z\right)$.

This weighted particle algorithm is presented and used with w_{incep} as a constant. In this setting, this weighted algorithm is equivalent to the Mass Flow Algorithm (MFA) [8]. The weighted algorithm generalises the MFA in that w_{incep} may be a random variable with distribution depending on t, x and even z. This would lead to a more complex form for (20), but all other details of the algorithm would be unchanged.

In the case of constant w_{incep}, there is a redundancy between w_{incep} and V_1; only one of the two parameters is necessary to fully control the algorithm. To simplify the comparison with DSA methods, this work takes $w_{\text{incep}} \equiv 1$ and uses V_1 to control the algorithm as discussed in §2. The redundancy is not immediately obvious, but it arises because coagulation changes statistical weights multiplicatively (see the definition of $\tilde{w}_1$ before (32)) and so all weights scale with w_{incep}. The definition of the coagulation rate for the weighted particle system shows that it also scales linearly with weights and hence with w_{incep}. Therefore changing w_{incep} and V_1 while keeping their ratio constant has no effect on the algorithm properties.

2.2.2 Coagulation

Coagulation jumps change only one particle and are of the form

$$\begin{aligned}
&\left(x_{j_1,n}, z_{j_1,n}, w_{j_1,n}\right), \left(x_{j_2,n}, z_{j_2,n}, w_{j_2,n}\right) \to \\
&\qquad \left(x_{j_1,n}, z_{j_1,n} + z_{j_2,n}, w_{j_1,n} \frac{\bar{m}(z_{j_1,n})}{\bar{m}(z_{j_1,n}) + \bar{m}(z_{j_2,n})}\right), \left(x_{j_2,n}, z_{j_2,n}, w_{j_2,n}\right).
\end{aligned} \tag{21}$$

Other definitions of this jump and the conservation properties that result from different definitions of the jump are studied in [23]. The jump 21 occurs in cell $\mathcal{X}_i$ with rate

$$\frac{\tilde{K}\left(z_{j_1,n}, w_{j_1,n}, z_{j_2,n}, w_{j_2,n}\right)}{nV_1} \mathbb{1}\left\{x_{j_1,n} \in \mathcal{X}_i\right\} \mathbb{1}\left\{x_{j_2,n} \in \mathcal{X}_i\right\}, \tag{22}$$

where $\tilde{K}\left(z_{j_1,n}, w_{j_1,n}, z_{j_2,n}, w_{j_2,n}\right) = K\left(z_{j_1,n}, z_{j_2,n}\right) w_{j_2,n}$.

2.2.3 Limiting Equation

The convergence of the SWA and a weak differential equation for the limit within each splitting step after integrating out the weights is discussed in §2.3.2. Using the same additional assumption as in §2.1.3 one has the following weak equation for the density of the limiting population:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{X}\times\mathcal{Z}} \phi(x,z)\tilde{c}(t,x,z)\mathrm{d}x\mathrm{d}z &= \sum_{i=1}^{n_{\text{cell}}}\int_{\mathcal{X}_i\times\mathcal{Z}} \phi(x,z) I(\bar{x}_i, z)\mathrm{d}x\mathrm{d}z \\ &+ \frac{1}{2}\sum_{i=1}^{n_{\text{cell}}}\int_{(\mathcal{X}_i\times\mathcal{Z})^2}\left[\frac{\bar{m}(z_1)}{\bar{m}(z_1)+\bar{m}(z_2)}\phi(x_1, z_1+z_2) - \phi(x_1,z_1)\right] \\ &\qquad K(z_1,z_2)\tilde{c}(t,x_1,z_1)\tilde{c}(t,x_2,z_2)\mathrm{d}x_1\mathrm{d}z_1\mathrm{d}x_2\mathrm{d}z_2. \end{aligned} \tag{23}$$

This is identical to (17) for the limiting density of DSA2 and one therefore expects that for given Δx_{cell} and Δt_{split} that the SWA and the DSA2 converge to the same limit as $n \to \infty$.

2.3 Convergence

In the first instance, these results apply for the simulation of the particle processes for one splitting step. However, because the streaming step is deterministic, convergence at the end of the simulation for one splitting step implies that the initial conditions for the simulation in the next splitting step converge and hence one has a global in time convergence of the stochastic particle simulation. A differential equation for the limit is however only available within the stochastic part of each splitting step, excluding the streaming part of the splitting.

2.3.1 Direct Simulation Algorithms

General convergence results [8, 9, 6] imply that, for continuous functions ϕ with compact support

$$\lim_{n\to\infty}\frac{1}{nV_1}\sum_{j=1}^{N_n(t)}\phi\left(x_{j,n}(t), z_{j,n}(t)\right) = \int_{\mathcal{X}\times\mathcal{Z}}\phi(x,z)\nu(t,\mathrm{d}x,\mathrm{d}z) \tag{24}$$

where ν is characterised by

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{X}_i\times\mathcal{Z}}\phi(x,z)\nu(t,\mathrm{d}x,\mathrm{d}z)=\int_{\mathcal{X}_i\times\mathcal{Z}}\phi(x,z)I(\bar{x}_i,z)\mathrm{d}x\mathrm{d}z$$
$$+\frac{1}{2\left|\mathcal{X}_i\right|}\int_{(\mathcal{X}_i\times\mathcal{Z})^2}\int_{\mathcal{X}_i}\left[\phi(x,z_1+z_2)-\phi(x_1,z_1)-\phi(x_2,z_2)\right]$$
$$S(x_1,z_1,x_2,z_2,\mathrm{d}x)K(z_1,z_2)\nu(t,\mathrm{d}x_1,\mathrm{d}z_1)\nu(t,\mathrm{d}x_2,\mathrm{d}z_2)$$
$$i=1,\dots,n_{\text{cell}}. \quad (25)$$

In the case that p is symmetric, in particular when $p\equiv\frac{1}{2}$ (DSA1), substituting the definition of S from (14) into (25) and exploiting the symmetry leads to

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{X}_i\times\mathcal{Z}}\phi(x,z)\nu(t,\mathrm{d}x,\mathrm{d}z)=\int_{\mathcal{X}_i\times\mathcal{Z}}\phi(x,z)I(\bar{x}_i,z)\mathrm{d}x\mathrm{d}z$$
$$+\frac{1}{2\left|\mathcal{X}_i\right|}\int_{(\mathcal{X}_i\times\mathcal{Z})^2}\left[\phi(x_1,z_1+z_2)-\phi(x_1,z_1)-\phi(x_2,z_2)\right]$$
$$K(z_1,z_2)\nu(t,\mathrm{d}x_1,\mathrm{d}z_1)\nu(t,\mathrm{d}x_2,\mathrm{d}z_2). \quad (26)$$

In the case that p is proportional to the mass of the first particle (DSA2), that is, when (15) holds then $1-p(x_1,z_1,x_2,x_2)=p(x_2,z_2,x_1,x_1)$ and substituting into (25) and exploiting the symmetry between (x_1,z_1) and (x_2,z_2) yields

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{X}_i\times\mathcal{Z}}\phi(x,z)\nu(t,\mathrm{d}x,\mathrm{d}z)=\int_{\mathcal{X}_i\times\mathcal{Z}}\phi(x,z)I(\bar{x}_i,z)\mathrm{d}x\mathrm{d}z$$
$$+\frac{1}{\left|\mathcal{X}_i\right|}\int_{(\mathcal{X}_i\times\mathcal{Z})^2}\left[p(x_1,z_1,x_2,x_2)\phi(x_1,z_1+z_2)-\phi(x_1,z_1)\right]$$
$$K(z_1,z_2)\nu(t,\mathrm{d}x_1,\mathrm{d}z_1)\nu(t,\mathrm{d}x_2,\mathrm{d}z_2). \quad (27)$$

2.3.2 Stochastic Weighted Algorithm

In the same way, one sees that

$$\lim_{n\to\infty}\frac{1}{nV_1}\sum_{j=1}^{N_n(t)}\psi\left(x_{j,n}(t),z_{j,n}(t),w_{j,n}(t)\right)=\int_E\psi(x,z,w)\,f(t,\mathrm{d}x,\mathrm{d}z,\mathrm{d}w)\,, \quad (28)$$

where $E=\mathcal{X}\times\mathcal{Z}\times[0,w_{\max}]$ is an extended state space and ψ is continuous with compact support. The kernel f is characterised by the equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_E\psi(x,z,w)\,f(t,\mathrm{d}x,\mathrm{d}z,\mathrm{d}w)=$$
$$\int_{E^2}\left[\psi\left(x_1,z_1+z_2,\frac{\bar{m}(z_1)}{\bar{m}(z_1)+\bar{m}(z_2)}w_1\right)-\psi(x_1,z_1,w_1)\right]\times$$
$$\tilde{K}(z_1,w_1,z_2,w_2)\,f(\mathrm{d}x_1,\mathrm{d}z_1,\mathrm{d}w_1)\,f(\mathrm{d}x_2,\mathrm{d}z_2,\mathrm{d}w_2). \quad (29)$$

If one considers test functions of the form $\psi(x, z, u) = \phi(x, z)$ for ϕ continuous and compactly supported on $\mathcal{X} \times \mathcal{Z}$ then

$$\lim_{n\to\infty} \frac{1}{nV_1} \sum_{j=1}^{N_n(t)} \phi\left(x_{j,n}(t), z_{j,n}(t)\right) = \int_E \phi(x, z)\, \nu(t, \mathrm{d}x, \mathrm{d}z)\,, \tag{30}$$

where ν is defined as follows to be comparable to the same symbol in (25)

$$\nu(t, \mathrm{d}x, \mathrm{d}z) = \int_0^{w_{\max}} u\, f(t, \mathrm{d}x, \mathrm{d}z, \mathrm{d}u). \tag{31}$$

With ν as defined in (31), (29) becomes

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathcal{X}_i \times \mathcal{Z}} \phi(x, z)\, \nu(t, \mathrm{d}x, \mathrm{d}z) = \\ \int_{(\mathcal{X}_i \times \mathcal{Z})^2} \left[\frac{\bar{m}(z_1)}{\bar{m}(z_1) + \bar{m}(z_2)} \phi(x_1, z_1 + z_2) + -\phi(x_1, z_1) \right] \\ K(z_1, z_2)\, \nu(t, dx_1, dz_1)\, \nu(t, dx_2, dz_2) \\ i = 1, \ldots, n_{\text{cell}}. \end{aligned} \tag{32}$$

Equation (32) is identical to (27) showing that DSA2 and the SWA described in this section have the same limiting process as $n \to \infty$ and in particular that limits of the SWA are solutions to (25) with the natural interpretation of the weights given by (31).

2.4 Mass Moments

Define the unsteady local mass moments for $k \in \mathbb{N}$ by

$$m_k(t, x) = \int_{\mathcal{Z}} \bar{m}(z)^k c(t, x, z) \mathrm{d}z \tag{33}$$

where c is the solution to (2).

For large t, c approaches the solution of the time independent version of (2)

$$\begin{aligned} u \nabla_x c(x, z) = I(x, z) \\ + \frac{1}{2} \int_{\substack{z_1, z_2 \in \mathcal{Z} \\ z_1 + z_2 = z}} K(z_1, z_2) c(x, z_1) c(x, z_2) \mathrm{d}z_1 \mathrm{d}z_2 \\ - \int_{z_1 \in \mathcal{Z}} K(z_1, z) c(x, z_1) c(x, z) \mathrm{d}z_1 \end{aligned} \tag{34}$$

for $x \in [0, L]$, $z \in \mathcal{Z}$ and with boundary condition

$$c(0, z) = c_{\text{in}}(z). \tag{35}$$

In the following, the moments of this steady state c

$$m_k(x) = \int_{\mathcal{Z}} \bar{m}(z)^k c(x, z) \mathrm{d}z \tag{36}$$

are studied, because time averaging can be used to estimate the expectation from (7). In practice one observes simulated values of

$$M_{n,k}(t, x) := \frac{1}{nV_1} \sum_{j=1}^{N_{i,n}(t)} \bar{m}\left(z_{j,n}(t)\right)^k w_{j,n}(t) \tag{37}$$

where $x \in \mathcal{X}_i$ for many values of t in the same simulation. Samples are observed at times $j\Delta t_{\text{sample}},\ j \in \mathbb{N}$ with $\Delta t_{\text{sample}} = n\Delta t_{\text{split}}$ for some $n \in \mathbb{N}$. For t large enough the system approaches a steady state and provided Δt_{sample} is not too small one can view successive samples of $M_{n,k}(t, x)$ for $t > t_{\text{relax}}$ as approximately independent realisations of a time independent random variable $M_{n,k}(x)$. This is computationally very valuable, because it reduces the number of realisations of the Markov process that have to be generated.

3 Numerical Results

3.1 Constant Coagulation Kernel

In this section a system of length $L = 2\times 10^{-1}$, $\mathcal{Z} = \mathbb{R}^+$ is taken to be mass, $K(z_i, z_j) \equiv K_0$, $u \equiv 1$, with $K_0 = 10^{-5}$ and $I(x, z) = I_0 \times \mathbb{1}\{z = 1\}$ with $I_0 = 3 \times 10^8$. The left (inflow) boundary condition is $c_{\text{in}} \equiv 0$.

Numerical parameters were $V_1 = 10^{-7}$ for DSA1&2 and $V_1 = 1.25 \times 10^{-8}$ for the SWA. Further parameters were $\Delta t_{\text{split}} = 10^{-4}$, $\Delta x_{\text{cell}} = 4 \times 10^{-3}$ and $\Delta t_{\text{sample}} = 5 \times 10^{-3}$. For inception in the SWA, $w_{\text{incep}} = 1$ was used. Averaging of moments was carried out over the time interval $[0.4, 4.0]$ so that $t_{\text{relax}} = 0.4$.

Using simple fluid mechanics style reasoning for small control volumes one derives from (34) the following equations and solutions for the total mass moments:

$$u\nabla m_0(x) = 3I_0 - \frac{K_0}{2} m_0(x)^2, \quad m_0(0) = 0 \tag{38}$$

$$m_0(x) = \sqrt{\frac{6I_0}{K_0}} \tanh\left(\sqrt{\frac{3I_0K_0}{2}} \frac{x}{u}\right) \tag{39}$$

$$u\nabla m_1(x) = 3I_0, \quad m_1(0) = 0 \tag{40}$$

$$m_1(x) = \frac{3I_0}{u} x \tag{41}$$

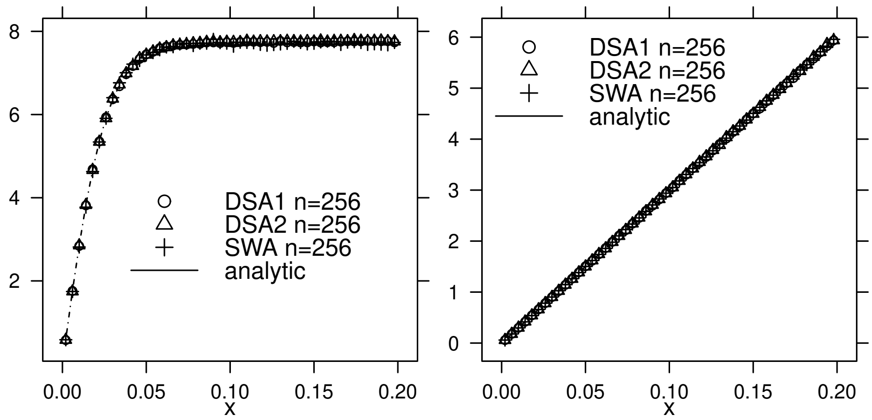
$$u\nabla m_2(x) = 3I_0 + K_0 m_1(x)^2, \quad m_2(0) = 0 \tag{42}$$

$$m_2(x) = \frac{3I_0}{u} x + \frac{3I_0^2 K_0}{u^3} x^3 \tag{43}$$

$$u\nabla m_3(x) = 3I_0 + 3K_0 m_1(x) m_2(x) \quad m_3(0) = 0 \tag{44}$$

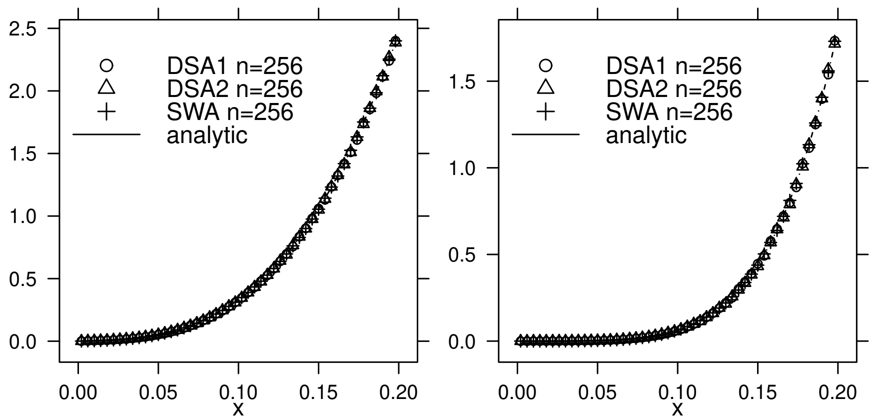
$$m_3(x) = \frac{3I_0}{u}x + \frac{9I_0^2K_0}{u^3}x^3 + \frac{27I_0^3K_0^2}{5u^5}x^5 \tag{45}$$

Figures 1&2 show that all three methods considered successfully estimate $m_k(x)$ $k = 0, 1, 2, 3$ with averaged values of $M_{n,k}(t, x)$ with $n = 256$. The results for $n = 1024, 4096, 16384$ were similar, showing that the method converges for rather small values of n.



(a) $M_{0,n}(x)/10^6$ (b) $M_{1,n}(x)/10^7$

Figure 1: First two moments for constant kernel test problem



(a) $M_{2,n}(x)/10^9$ (b) $M_{3,n}(x)/10^{11}$

Figure 2: Additional moments for the constant kernel test problem

The variance of the $M_{n,k}(t, x)$ samples closely approaches the $1/n$ scaling that is usual for Monte Carlo methods. Detailed results are shown in Figures 3–6. DSA1&2 show very similar behaviour, but are qualitatively different from SWA for $M_{n,k}$ when $k = 1, 2, 3$, that is, for all except the zeroth moment. For the three higher moments the relative variance for small x (where the system is relatively empty) is much lower for the DSA methods than for the SWA. However, the DSA methods show relative variance that increases with x whereas the SWA has relative variance that decreases with x. The relative variances intersect close to $x = 0.1$, but at a point which appears to move left with increasing k. The difference between the DSA methods and

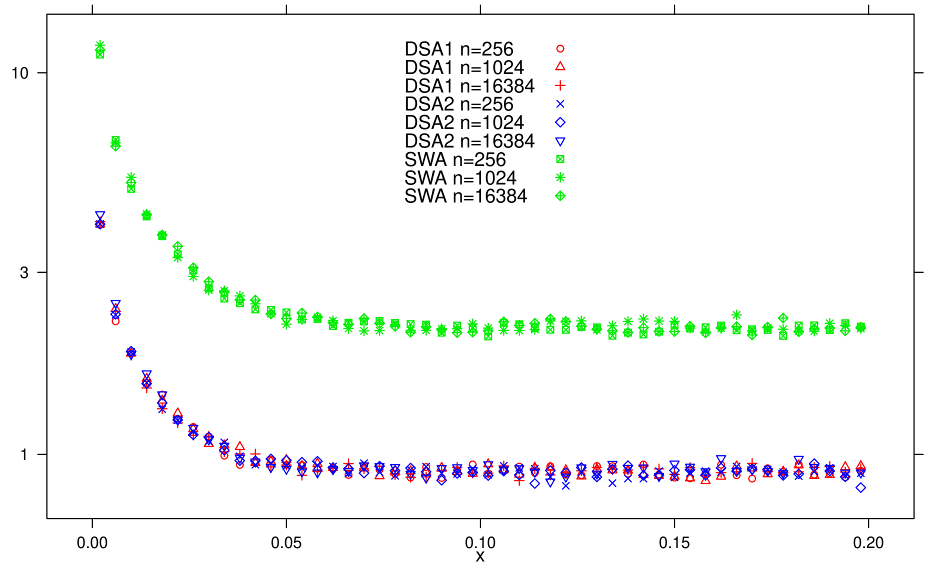


Figure 3: Normalised standard deviation for simulated zeroth moment of the constant kernel test problem $\sqrt{n \times \text{var}\left(M_{n,0}(x)\right)}/m_0(x)$.

the SWA may partly be explained by the way in which the SWA weights the simulation particles so that the larger, rarer particles are better resolved than with DSA. The price for this is poorer resolution of small particles; smaller particles are more common (since there is continuous inception) and so the DSA methods simulate $M_{n,0}$ better than the SWA [23]. However, while the outperformance of the SWA for $M_{n,k},\ k = 2, 3$ might be explained by the weighting, the case of $M_{n,1}$ is more complex. From a physical standpoint, coagulation should have no effect on how mass moves through the system, and in the absence of coagulation, one finds that, for $x \in \mathcal{X}_i$ and upto approximations due to the finite spatial discretisation,

$$nV_1 M_{n,1}(t, x) \sim \text{Poi}\left(\frac{3I_0 \Delta x_{\text{cell}}(2i+1)}{2u} nV_1\right) \tag{46}$$

so that (note $x \approx \Delta x_{\text{cell}} \frac{2i+1}{2}$, the cell mid-point)

$$n \frac{\text{var}\left(M_{n,1}(t, x)\right)}{m_1 \left(\Delta x_{\text{cell}} \frac{2i+1}{2}\right)^2} = \frac{2u \Delta x_{\text{cell}}}{3I_0 V_1 (2i+1)} \approx \frac{u}{3I_0 V_1 x}. \tag{47}$$

Equation (47) should apply to the DSA results and to the SWA results with the appropriate values of V_1. These values implied by (47) are plotted as lines in Figure 4, where the SWA shows a good match but the DSA methods are seen to quickly start generating extra variance.

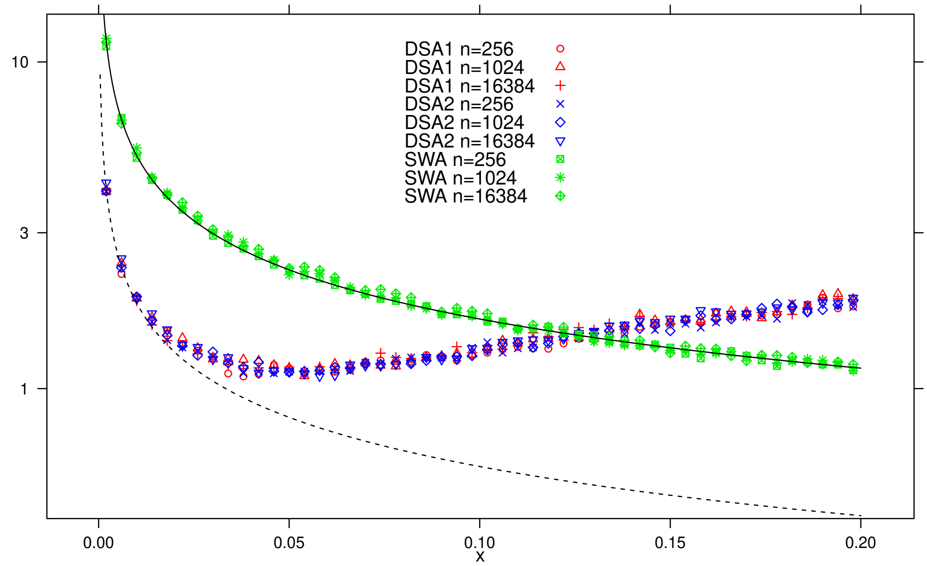


Figure 4: Normalised standard deviation for simulated first moment of the constant kernel test problem $\sqrt{n \times \mathrm{var}\left(M_{n,1}(x)\right)}/m_1(x)$ (symbols) and theoretical limit from inception process (SWA solid line, DSA dashed line).

This indicates a fundamental difference between DSA and SWA coagulation, even though both are exactly mass conserving on the grid scale. The key difference is that DSA coagulation events increase the physical mass associated with one computational particle and remove a second, whereas SWA coagulation events, because of their use of statistical weights, avoid the removal of the second computational particle.

For the problem considered in this section, systematic errors are small, even for $n = 256$. The optimal choice of method within the range of n studied, reduces to a question about which can achieve a specified confidence interval for the mean value of a chosen functional with the smallest amount of computer time. Figures 3 to 6 show that for $M_{n,k}(x)$ the answer to this question will depend on x and k and that the differences are small enough that the method could reasonably be chosen arbitrarily rather than spending time on a detailed comparison. However the increasing nature of the relative standard deviation for the DSA algorithms is a cause for anxiety. A second test problem, with a strongly size dependent coagulation kernel is now considered and it will be seen that this issue with the DSA methods becomes much more serious.

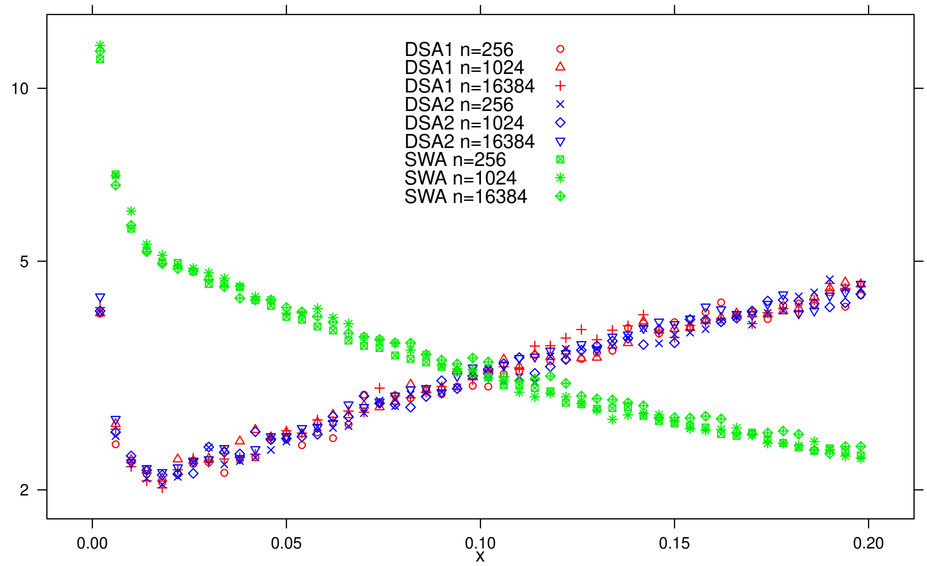


Figure 5: Normalised standard deviation for simulated second moment of the constant kernel test problem $\sqrt{n \times \mathrm{var}\left(M_{n,2}(x)\right)}/m_2(x)$.

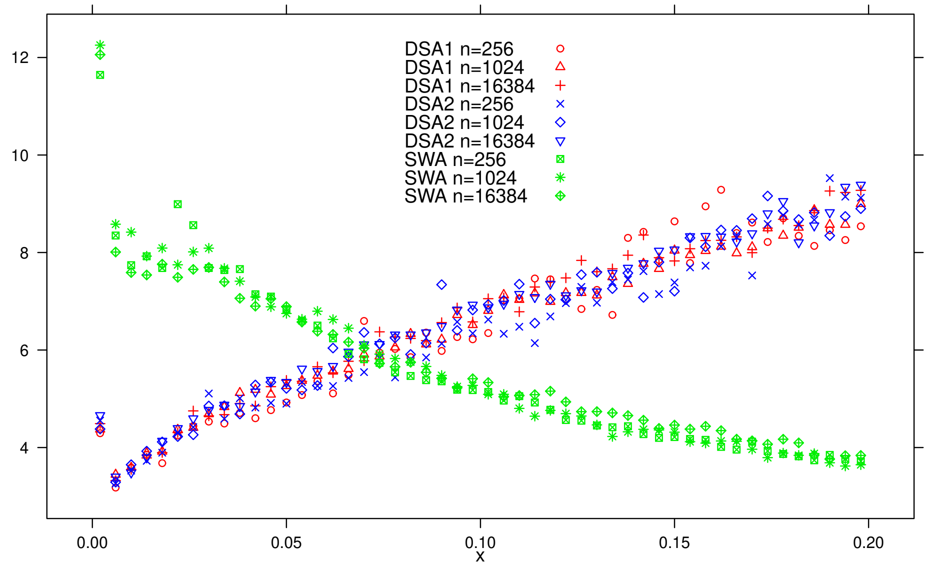


Figure 6: Normalised standard deviation for simulated third moment of the constant kernel test problem $\sqrt{n \times \mathrm{var}\left(M_{n,3}(x)\right)}/m_3(x)$.

3.2 Additive Coagulation Kernel

In this section a system of length $L = 10^{-1}$, $\mathcal{Z} = \mathbb{N}$ is taken to be particle mass, $K(z_i, z_j) = K_0 \times (z_i + z_j)$, $u = 1$, with $K_0 = 1.06 \times 10^{-2}$ (the parameters were originally chosen on physical grounds) and $I(x, z) = 10^5 \mathbb{1}\{z = 1\}$.

The additive kernel is the most strongly size dependent kernel that does not, in the spatially homogenous case, lead to the formation of infinite particles in finite time (gelling).

Since coagulation does not affect the mass of particles it is easy to see that

$$\frac{\partial}{\partial t} m_1(t, x) + u \nabla_x m_1(t, x) = I(1) \tag{48}$$

and that

$$m_1(t, x) = I(1) \left(\frac{x}{u} \wedge t \right). \tag{49}$$

The steady state limit is well behaved with

$$m_1(x) = I(1) \frac{x}{u} = \lim_{t \to \infty} m_1(t, x). \tag{50}$$

The numerical parameters were as follows: $\Delta x_{\text{cell}} = 2 \times 10^{-3}$, $\Delta t_{\text{split}} = 4 \times 10^{-4}$, $\Delta t_{\text{sample}} = 4 \times 10^{-3}$, $V_1 = \frac{1}{3} \times 10^{-3}$ for DSA1&2 and $V_1 = 2^{-7} \times 10^{-2} = 7.8125 \times 10^{-5}$ for the SWA. These values of V_1 mean that the long run average number of computational particles in the entire domain was just under $28n$ for the DSA cases and just under $20n$ for the SWA.

DSA moments were averaged over $[0.2, 4.0]$, SWA over $[0.2, 0.4]$, that is, $t_{\text{relax}} = 0.2$ in both cases but the time averaging of the DSA samples of $M_{i,n,k}$ was over a much longer interval. The extended sampling interval was chosen to make the DSA and SWA runtimes approximately equal for the same value of n in the initial implementation of the algorithms.

The Markov chains were realised multiple times with different seeds for the pseudo random number generator (the Mersenne twister [21]). The number of realisations was chosen to be $262144/n$ in order to achieve similar sized confidence intervals for all values of n.

The mean of the simulated particle concentration $M_{n,0}(x)$ is plotted in Figure 7 and shows little dependence on algorithm or n. However, time averaged values of $M_{n,1}(x)$ show a surprising lack of smoothness (contrast Equation (50)) for large x. Plots of the averaged $M_{n,1}(x)$ are given in Figure 8 and both sets of DSA data show oscillations for $n = 2048, 16384$, but the SWA has only the slightest deviation from the theoretical straight line, even for $n = 2048$. A quantitative description of this phenomenon is provided in Table 1, which shows the mean and standard deviation of the sampled $M_{n,1}$ values and an estimated of the 95% confidence interval half width for the sample mean calculated with the usual central limit theorem approximation. The data in Table 1 show that the DSA2 and SWA closely approach the true value $10^5 \times 0.099 = 9.90 \times 10^3$ (calculating at the cell centre). However, the DSA1 shows a small, but statistically significant, difference.

Equation (47) also applies to the rescaled $M_{n,1}(x)$ standard deviation for the additive coagulation kernel test problem under consideration here and is compared to the simulated values in Figure 9. This figure closely resembles the corresponding Figure 4 for the constant coagulation

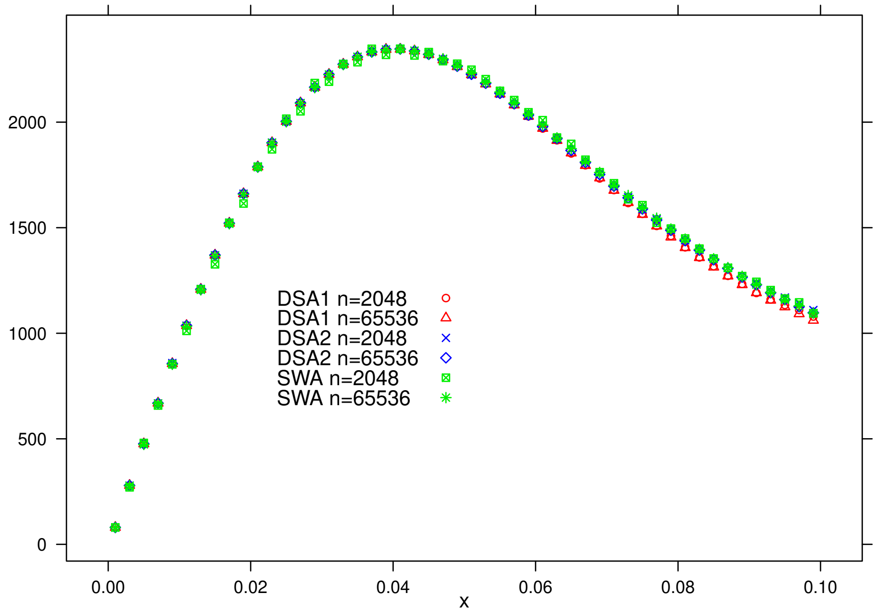


Figure 7: Mean of simulated particle concentration $M_{n,0}(x)$ for the additive coagulation kernel test problem

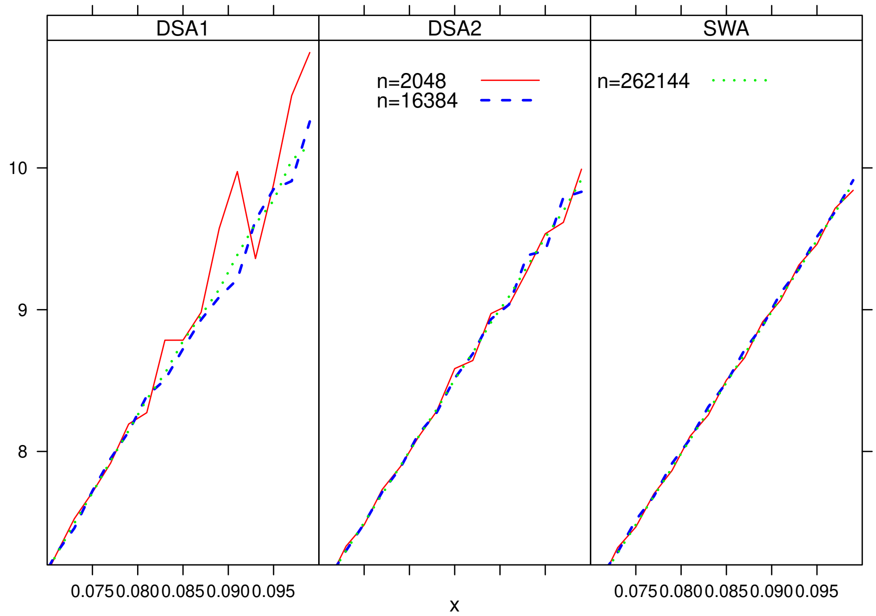


Figure 8: Zoomed comparison of $M_{n,1}(x)/10^3$ for the additive coagulation kernel test problem

Table 1: Confidence intervals for the mean of $M_{n,1}$ estimated from the cell $[0.098, 0.1]$.

	n	sample mean $/10^3$	sample std. dev. $/10^3$	95% conf. int. $/10^3$
DSA1	2048	10.570	11.09	0.062
DSA1	4096	10.514	9.99	0.079
DSA1	8192	10.347	7.64	0.086
DSA1	16384	10.097	5.09	0.081
DSA1	32768	10.112	4.64	0.104
DSA1	65536	10.319	3.66	0.116
DSA1	131072	10.147	2.56	0.114
DSA1	262144	10.132	1.99	0.126
DSA2	2048	9.894	9.923	0.0220
DSA2	4096	9.854	3.271	0.0260
DSA2	8192	9.874	2.645	0.0297
DSA2	16384	9.898	2.064	0.0293
DSA2	32768	9.880	1.534	0.0345
DSA2	65536	9.890	1.143	0.0363
DSA2	131072	9.871	0.811	0.0365
DSA2	262144	9.894	0.586	0.0372
SWA	2048	9.878	0.249	0.0060
SWA	4096	9.887	0.178	0.0061
SWA	8192	9.886	0.126	0.0061
SWA	16384	9.880	0.085	0.0059
SWA	32768	9.877	0.064	0.0062
SWA	65536	9.878	0.046	0.0063
SWA	131072	9.876	0.030	0.0059
SWA	262144	9.877	0.020	0.0055

kernel problem from the previous section. One again sees that the SWA adds little variance beyond that which is intrinsic to the inception process, whereas the two DSA variants introduce an ever increasing amount of noise, which in this case leads to a loss of solution quality. At

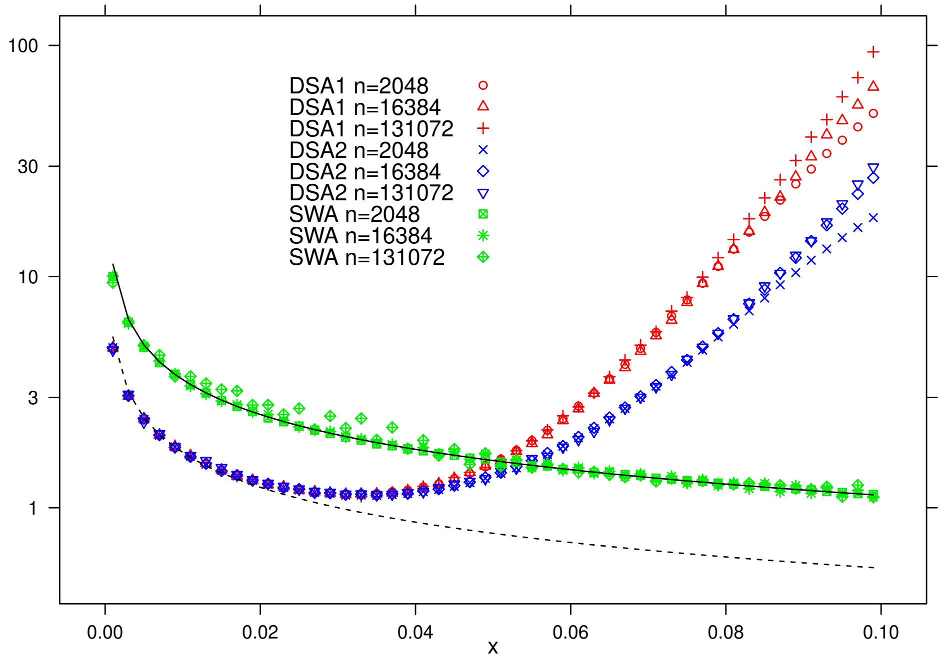


Figure 9: Normalised standard deviation for simulated first moment of the additive kernel test problem $\sqrt{n \times \mathrm{var}\left(M_{n,1}(x)\right)}/m_1(x)$ (symbols) and theoretical limit from inception process (SWA solid line, DSA dashed line).

$x = 0.1$ Figure 9 shows a factor of approximately 30 difference in the standard deviations of the $M_{n,1}(t,x)$ samples generated with the DSA2 and the SWA.

The problems with the DSA methods are more severe for the higher order mass moments—considerable dependence on n can be seen for the DSA generated values summarised in Table 2. In particular, one sees that the DSA1 means differ greatly from the DSA2 and SWA limiting values. However, the excessive fluctuations seen in the higher moments eventually feed back into $M_{n,0}$ as shown in Figure 10.

In Table 2 the SWA mean values appear to converge to approximately 9.1×10^7. All the SWA values are within 3% of this value. The DSA2 values are consistent with convergence to the same limit, as expected from §2.2.3, but it takes until $n = 131072$ just to reach the 3% tolerance and to bring the sample mean within one confidence interval half width of the supposed limit. A similar pattern is seen for $M_{n,3}$ reinforcing the impression that the SWA is generally a more precise method than both the variants of the DSA. It is interesting to note that the SWA showed a much more limited advantage for homogeneous problems where there was no question of where to place computational particles after computation in the DSA [23]. This observation, along with the differences between DSA1 and DSA2 support the conclusion that the key problem with the DSA methods in this setting of coagulation and advection is the numerical noise generated in

Table 2: Confidence intervals for the mean of $M_{n,2}$ estimated from the cell $[0.098, 0.1]$.

	n	sample mean $/10^7$	sample std. dev. $/10^7$	95% conf. int. $/10^7$
DSA1	2048	10.55	50.83	0.286
DSA1	4096	16.95	123.15	0.978
DSA1	8192	20.63	100.39	1.128
DSA1	16384	19.92	61.58	0.978
DSA1	32768	31.45	105.39	2.368
DSA1	65536	40.68	134.16	4.263
DSA1	131072	40.79	82.59	3.712
DSA1	262144	48.09	99.28	6.310
DSA2	2048	2.953	3.45	0.019
DSA2	4096	4.176	4.69	0.037
DSA2	8192	5.549	5.97	0.067
DSA2	16384	6.836	6.70	0.095
DSA2	32768	7.660	6.39	0.144
DSA2	65536	8.376	5.80	0.184
DSA2	131072	8.661	4.60	0.207
DSA2	262144	8.981	3.26	0.207
SWA	2048	9.354	2.09	0.051
SWA	4096	9.273	1.42	0.049
SWA	8192	9.192	1.02	0.050
SWA	16384	9.110	0.72	0.049
SWA	32768	9.049	0.50	0.049
SWA	65536	9.105	0.48	0.047
SWA	131072	9.145	0.26	0.050
SWA	262144	9.110	0.18	0.049

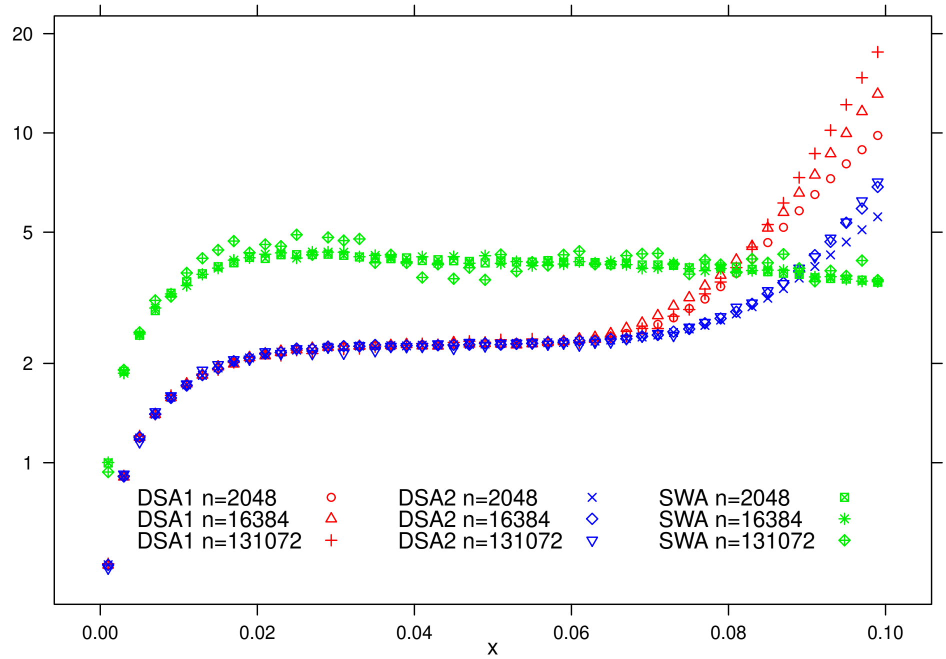


Figure 10: Normalised standard deviation for simulated zeroth moment of the additive kernel test problem $\sqrt{n \times \mathrm{var}\left(M_{n,0}(x)\right)}/10^3$.

the spatial distribution of mass during simulated computational events.

Mean moment values for DSA2 and SWA do not change significantly when Δx_{cell} is reduced by a factor of 10 (increasing the number of cells and total number of computational particles in the domain by the same factor). The very large errors of DSA1 are reduced by decreasing Δx_{cell}, which again supports the theory that the disruption of the spatial mass during computational coagulation events is at the root of the fluctuations observed in the DSA methods.

4 Conclusion

The present work developed from an unsuccessful attempt to extend a spatially homogeneous Direct Simulation Algorithm for soot formation [22] to the spatially inhomogeneous case. Initial attempts showed large instabilities and systematic errors. Investigations eventually identified the same problems in the additive coagulation kernel test problem considered in §3.2. The key features of the initial difficulties could be reproduced without incorporating reactions on particle surfaces, which were accordingly excluded here in the interests of simplicity. The exclusion of diffusion is justified as a first approximation, because soot particles are large compared to surrounding gas molecules.

The present work shows that DSA methods are vulnerable to instabilities the manifest themselves first in high variance and then in systematic error when used for the simulation of coagulation–advection problems. The SWA avoids these problems, which are much more severe for the strongly size dependent additive coagulation kernel than they are for the size-independent constant coagulation kernel.

Despite these fundamental differences in numerical performance, it is also shown theoretically that both the DSA methods, as well as the SWA, converge to deterministic limits as the number of computational particles approaches infinity. This convergence behaviour was clearly observed for the SWA and DSA2 in the numerical examples.

The instability with the DSA methods appears to stem from the nature of the computational jumps that capture the effects of physical coagulation processes. During the DSA simulation of coagulation, one computational particle becomes larger and a second (the source of the mass to grow the first) is removed. This increases the amount of mass at the position of the first particle and reduces the amount of mass at the position of the second particle. Selecting the order of the coagulation partners uniformly at random makes this effect more severe than choosing the order such that the probability of a particle being the first partner is proportional to its mass.

Weighted particle methods exploit the freedom to choose statistical weights to keep the amount of mass constant at both points and simply alter the distribution of mass across the particle size or type space at the location of the first particle. In the weighted particle setting coagulation adds little noise to the first moment of the mass distribution beyond that which is intrinsic to the inception process.

Future work will address the incorporation of surface reactions, which are important for physical modelling purposes [22] and the refinement of the stochastic weighted particle algorithm.

References

[1] M Balthasar and M Kraft. A stochastic approach to solve the particle size distribution function of soot particles in laminar premixed flames. *Combust. Flame*, 133:289–298, 2003.

[2] G A Bird. *Molecular gas dynamics*. Clarendon press, 1976.

[3] G A Bird. Approach to translational equilibrium in a rigid sphere gas. *Phys. Fluids*, 6(10): 1518–1519, 1963. doi: 10.1063/1.1710976.

[4] G A Bird, M A Gallis, J R Torczynski, and D J Rader. Accuracy and efficiency of the sophisticated direct simulation Monte Carlo algorithm for simulating noncontinuum gas flows. *Phys. Fluids*, 21:017103, 2009. doi: 10.1063/1.3067865.

[5] M Celnik, R Patterson, M Kraft, and W Wagner. A predictor–corrector algorithm for the coupling of stiff ODEs to a particle population balance. *J. Comput. Phys.*, 228:2758–2769, 2009. doi: 10.1016/j.jcp.2008.12.030.

[6] M Deaconu, N Fournier, and E Tanré. A pure jump Markov process associated with Smoluchowski's coagulation equation. *Ann. Prob.*, 30(4):1763–1796, 2002.

[7] Y Efendiev and M R Zachariah. Hierarchical hybrid Monte-Carlo method for simulation of two-component aerosol nucleation, coagulation and phase segregation. *J. Aerosol Sci.*, 34:169–188, 2003. doi: 10.1016/S0021-8502(02)00156-8.

[8] A Eibeck and W Wagner. Stochastic particle approximations for Smoluchowski's coagulation equation. *Ann. Appl. Probab.*, 11:1137–1165, 2001.

[9] A Eibeck and W Wagner. Stochastic interacting particle systems and nonlinear kinetic equations. *Ann. Appl. Probab.*, 13:845–889, 2003.

[10] A L Garcia, C van den Broeck, M Aertsens, and R Serneels. A Monte Carlo simulation of coagulation. *Physica A.*, 143:535–546, 1987. doi: 10.1016/0378-4371(87)90164-6.

[11] D Grosschmidt, P Habisreuther, and H Bockhorn. Calculation of the size distribution function of soot particles in turbulent diffusion flames. *Proc. Combust. Inst.*, 31:657–668, 2007. doi: 10.1016/j.proci.2006.07.213.

[12] F Guiaş. Convergence properties of a stochastic model for coagulation-fragmentation processes with diffusion. *Stoch. Anal. Appl.*, 19(2):245–278, 2001. doi: 10.1081/SAP-100001188.

[13] F Guiaş. A stochastic approach for simulating spatially inhomogeneous coagulation dynamics in the gelation regime. *Commun. Nonlinear Sci. Numer. Simul.*, 14:204–222, 2009. doi: 10.1016/j.cnsns.2007.07.015.

[14] W Hackbusch, V John, A Khachatryan, and C Suciu. A numerical method for the simulation of an aggregation-driven population balance system. Technical Report 1621, WIAS, Berlin, 2011.

[15] P Lavvac, M Sander, M Kraft, and H Imanaka. Surface chemistry and particle shape: Processes for the evolution of aerosols in Titan's atmosphere. *Astrophys. J.*, 728:1–11, 2011. doi: 10.1088/0004-637X/728/2/80.

[16] N W Lee, J amd Sung and K Y Huh. Prediction of soot particle size distribution for turbulent reacting flow in a diesel engine. *Int. J. Engine Res.*, 12:181–189, 2011. doi: 10.1177/1468087410396286.

[17] D L Marchisio. On the use of bi-variate population balance equations for modelling barium titanate nanoparticle precipitation. *Chem. Eng. Sci.*, 64:697–708, 2009. doi: 10.1016/j.ces.2008.04.052.

[18] K Maruyama and Y Fujiyoshi. Monte Carlo simulation of the formation of snowflakes. *J. Atmos. Sci.*, 62:1529–1544, 2005. doi: 10.1175/JAS3416.1.

[19] I Mathies and W Wagner. Convergence of the stochastic weighted particle method for the Boltzmann equation. *SIAM J. Sci. Comput.*, 24:1589–1609, 2003. doi: 10.1137/S1064827502406014.

[20] T Matsoukas and C L Marshall, jr. Bicomponent aggregation in finite systems. *EPL*, 92:46007, 2010. doi: 10.1209/0295-5075/92/46007.

[21] M Matsumoto and T Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8:3–30, 1998. doi: 10.1145/272991.272995.

[22] R I A Patterson, J Singh, M Balthasar, M Kraft, and J R Norris. The linear process deferment algorithm: A new technique for solving population balance equations. *SIAM J. Sci. Comput.*, 28:303–320, 2006. doi: 10.1137/040618953.

[23] R I A Patterson, W Wagner, and M Kraft. Stochastic weighted particle methods for population balance equations. *J. Comput. Phys.*, 230:7456–7472, 2011. doi: 10.1016/j.jcp.2011.06.011.

[24] R Rudnicki and R Wieczorek. Phytoplankton dynamics: from the behavior of cells to a transport equation. *Math. Model. Nat. Phenom*, 1:81–97, 2006. doi: 10.1051/mmnp:2006005.

[25] M Sander, R I A Patterson, A Braumann, A Raj, and M Kraft. Developing the PAH-PP soot particle model using process informatics and uncertainty propagation. *Proc. Combust. Inst.*, 33:675–683, 2011. doi: 10.1016/j.proci.2010.06.156.

[26] Z Sun, R L Axelbaum, and R W Davis. A sectional model for investigating microcontamination in a rotating disk CVD reactor. *Aerosol Sci. Tech.*, 38:1161–1170, 2004. doi: 10.1080/027868290896799.

[27] S Tiwari, A Klar, and S Hardt. A particle-particle hybrid method for kinetic and continuum equations. *J. Comput. Phys.*, 228:7109–7124, 2009. doi: 10.1016/j.jcp.2009.06.019.

[28] W Wagner. A convergence proof for Bird's direct simulation Monte Carlo method for the Boltzmann equation. *J. Stat. Phys.*, 66:1011–1044, 1992. doi: 10.1007/BF01055714.

[29] Q Zhang, H Guo, F Liu, G J Smallwood, and M J Thomson. Modeling of soot aggregate formation and size distribution in a laminar ethylene/air coflow diffusion flame with detailed pah chemistry. *Proc. Combust. Inst.*, 32:761–768, 2009. doi: 10.1016/j.proci.2008.06.109.