Preprint Cambridge Centre for Computational Chemical Engineering ISSN 1473 – 4273

Stochastic weighted particle methods for population balance equations with coagulation, fragmentation and spatial inhomogeneity

Kok Foong Lee [1], Robert I. A. Patterson [3], Wolfgang Wagner [3], Markus Kraft [1,2]

released: 22 May 2015

[1] Department of Chemical Engineering
and Biotechnology
University of Cambridge
New Museums Site
Pembroke Street
Cambridge, CB2 3RA
United Kingdom
E-mail: mk306@cam.ac.uk

[2] School of Chemical
and Biomedical Engineering
Nanyang Technological University
62 Nanyang Drive
Singapore, 637459

[3] Weierstrass Institute for
Applied Analysis and Stochastics
Mohrenstraße 39
10117 Berlin
Germany
Email: wagner@wias-berlin.de

Preprint No. 154

Keywords: stochastic weighted particle methods, fragmentation, coagulation, weight transfer functions, compartmental model

Edited by
Computational Modelling Group
Department of Chemical Engineering and Biotechnology
University of Cambridge
New Museums Site
Pembroke Street
Cambridge CB2 3RA
United Kingdom

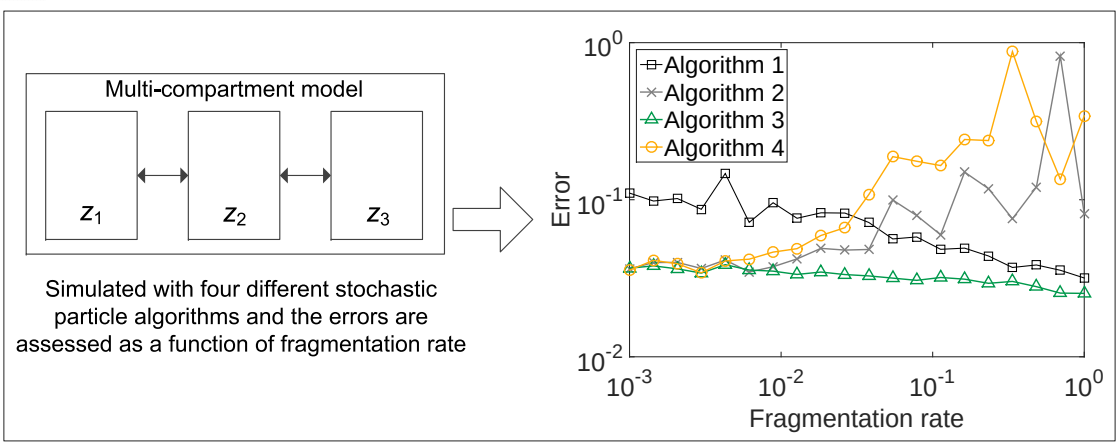
Fax: + 44 (0)1223 334796
E-Mail: c4e@cam.ac.uk
World Wide Web: http://como.cheng.cam.ac.uk/

Highlights

- Problems concerning multi-compartment population balance equations are studied
- A class of fragmentation weight transfer functions is presented
- Three stochastic weighted algorithms are compared against the direct simulation algorithm
- The numerical errors of the stochastic solutions are assessed as a function of fragmentation rate
- The algorithms are applied to a multi-dimensional granulation model

Abstract

This paper introduces stochastic weighted particle algorithms for the solution of multi-compartment population balance equations. In particular, it presents a class of fragmentation weight transfer functions which are constructed such that the number of computational particles stays constant during fragmentation events. The weight transfer functions are constructed based on systems of weighted computational particles and each of it leads to a stochastic particle algorithm for the numerical treatment of population balance equations. Besides fragmentation, the algorithms also consider physical processes such as coagulation and the exchange of mass with the surroundings. The numerical properties of the algorithms are compared to the direct simulation algorithm and an existing method for the fragmentation of weighted particles. It is found that the new algorithms show better numerical performance over the two existing methods especially for systems with significant amount of large particles and high fragmentation rates.



Contents

1 Introduction

Population balance equations track the change in the particle population with time through different particle mechanisms like coagulation, nucleation, and fragmentation. Traditionally, population balance models are one-dimensional with particle size as the focus. However, when much more detail is required, sectional methods [14] which usually formulate a population balance equation for each size class are not computationally feasible.

Stochastic particle methods have received increasing interest because they can easily incorporate multiple internal coordinates and avoid the 'curse of dimensionality' which affects other numerical methods [17]. Stochastic simulation of high-dimensional particle models has been applied successfully in a wide range of areas including soot [28], granulation [2], snowflake structure [20], and formation of precipitation particles in clouds [30].

Stochastic particle methods can be loosely separated into two categories: direct simulation algorithm (DSA) and stochastic weighted algorithms (SWAs). In the DSA, the physical processes are directly simulated, e.g. a coagulation event deletes the two original particles to form a new larger particle, and every computational particle represents the same number of physical particles. Although it is easy to implement, this method may allocate large amounts of computational resources to simulate the unimportant smaller particles and this leads to noisy estimates in the larger particle regions that dominate mass concentration [8, 27, 29].

To solve this issue, weighted particle methods, or SWAs may be used to utilise computational resources more efficiently. In SWAs, each particle is given a statistical weight which is proportional to the number of physical particles represented by the computational particle. Another key difference between the DSA and SWAs is that the coagulation and fragmentation processes do not require the number of computational particles to vary. Instead, the statistical weights are adjusted. In some way, SWAs are similar to the constant-N Monte Carlo method [18] which keeps the number of computational particles constant by resampling the ensemble after coagulation and fragmentation events, but this method is susceptible to noise especially for longer simulations.

The main challenge in the implementation of SWAs is the execution of processes which alter the number of particles, especially coagulation and fragmentation. In [9, 10, 31], a method called the Mass Flow Algorithm (MFA) is given for coagulation and fragmentation processes. The key difference between the MFA with the SWAs developed in this work is that the particles are not explicitly tagged with weights and the weights are implicitly determined as the inverse of the particle mass in the MFA. In systems with inception and frequent exchange of mass with the surroundings, frequent resampling of the particle ensemble is required and may be computationally inconvenient to implement. Nonetheless, successful attempts in extending the MFA to such systems have been reported [7, 23, 24].

The majority of previous studies involving SWAs with explicitly tagged weights focus on simulating coagulation accurately [8, 21, 25–27, 30, 32–34] and less attention is given to fragmentation. The fragmentation algorithm presented in [33] does not increase the number of computational particles where the weights of the resulting fragment particles are

added onto similar particles which are already in the ensemble. However, this algorithm is not suitable for high-dimensional models because it is impossible to search for particles that are identical to the fragment particles [19]. On the other hand, a fragmentation algorithm suitable for high-dimensional models is presented in [19], but the algorithm is not suitable for applications with high fragmentation rates because it increases the number of computational particles. Thus, frequent resampling of the particle ensemble may be required to accommodate the new fragment particles and this may cause severe numerical errors.

The main purpose of this paper is to extend the work done in [26] and develop efficient algorithms for the fragmentation of weighted particles which do not increase the number of computational particles. By having particles tagged with statistical weights, a myriad of possible weight transfer functions can be constructed for coagulation and fragmentation. The flexibility of the weight transfer functions is mainly suited for multidimensional models where any quantity can be chosen to be conserved across the jump processes. In granulation applications for example, the binder content may be chosen to be the quantity to conserve when the statistical weight is adjusted in a coagulation process. Work has been done in [26] where a large family of coagulation weight transfer functions is presented and this paper focuses on a similar aspect for fragmentation.

Besides fragmentation, another aspect of this work is the application of stochastic particle methods in spatially inhomogeneous systems. Stochastic particle methods are traditionally used to simulate spatially homogeneous problems but the inhomogeneous behaviour of mixing processes should not be neglected in systems of practical interest, such as soot formation during combustion [25] and wet granulation [19]. Hence, this leads to a number of attempts to extend stochastic particle methods to spatially inhomogeneous systems [13, 15, 19, 22, 25]. One of the common findings among these studies is that the usage of SWAs significantly reduces the stochastic noise due to the spatial inhomogeneity, which is highlighted in this work as well.

This paper is structured as follows: In Section 2, a general population balance model with coagulation, fragmentation and spatial inhomogeneity is formulated. The stochastic particle methods used in this work are described in detail in Section 3. Then, a test system is constructed in Section 4 to investigate the numerical properties of the stochastic particle methods before applying the methods to a realistic multidimensional granulation model in Section 5.

2 Population balance model

In this work, population balance problems with particles undergoing inception, coagulation, breakage, and transport are considered. Particles take positions in a bounded domain of compartments, $\mathbb{Z}$, and the particles are elements of a type space $\mathbb{X}$ which may be multidimensional and either discrete or continuous. Particles are allowed to move from one compartment to another, where the direction and rate of flow are specified by the connections between the compartments. However, only particles within the same compartment are allowed to interact with each other and undergo coagulation. Lastly, the rates of the mechanisms in each compartment are allowed to be different in order to capture the

inhomogeneous nature of most particle processes. The population balance problem is formulated for concentration measures $P(t,z,\mathrm{d}x)$, where $z \in \mathbb{Z}$ and $x \in \mathbb{X}$. So, $\int_U P(t,z,\mathrm{d}x)$ gives the number of particles with types $U \subset \mathbb{X}$ located in compartment z, and P is said to be a solution of the population balance problem if, for all $z \in \mathbb{Z}$:

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathbb{X}} \varphi(z,x)P(t,z,\mathrm{d}x) = \\
&\quad \frac{1}{2}\int_{\mathbb{X}}\int_{\mathbb{X}} \big[\varphi(z,x+y)-\varphi(z,x)-\varphi(z,y)\big]K(z,x,y)P(t,z,\mathrm{d}x)P(t,z,\mathrm{d}y) \\
&\quad + \frac{1}{2}\int_{\mathbb{X}}\int_{\mathbb{X}} \big[\varphi(z,y)+\varphi(z,x-y)-\varphi(z,x)\big]F(z,x,\mathrm{d}y)\,P(t,z,\mathrm{d}x) \\
&\quad + \int_{\mathbb{X}} \varphi(z,x)I(z,\mathrm{d}x) \\
&\quad + \sum_{j=1}^{N_{\mathrm{in}}(z)} \int_{\mathbb{X}} \varphi(z^{[j]},x) f_{\mathrm{flow}}(z^{[j]}\to z)\frac{1}{\tau(z^{[j]})}P(t,z^{[j]},\mathrm{d}x) \\
&\quad - \int_{\mathbb{X}} \varphi(z,x)\frac{1}{\tau(z)}P(t,z,\mathrm{d}x),
\end{aligned} \tag{1}$$

where φ is some suitable test function. Throughout this work, $\mathbb{Z}$ is finite, i.e. $\mathbb{Z} = \{z_1, z_2, \ldots, z_i\}$ where i is the number of compartments.

2.1 Particle processes

Each term in Equation (1) represents a particle process which is described in detail in this section.

2.1.1 Coagulation

The first term of (1) represents the coagulation process and is a slightly modified form of the Smoluchowski coagulation equation. Each pair of particles x and y coagulate at the rate specified by the kernel $K(z,x,y)$ to form a larger particle $(x+y)$. It is supposed that an operation "+" is defined in $\mathbb{X}$ for the coagulation process. Only particles within the same compartment are allowed to coagulate.

2.1.2 Fragmentation

The second term in Equation (1) represents the fragmentation process. The fragmentation model considered here is a binary fragmentation model where a particle x splits into y and $x-y$ according to the fragmentation kernel F. It is supposed that an operation "-" is defined in $\mathbb{X}$ to calculate one of the fragments deterministically from y.

The fragmentation kernel F is defined as:

$$F(x,\mathrm{d}y) = \mathbb{1}\{m(y) < m(x)\}\; g(z,x)\beta(x,y)\mathrm{d}y, \tag{2}$$

where $\mathbb{1}\{A\}$ is the indicator function on the set A, $m(x)$ is the mass of particle x, $g(z,x)$ is the breakage frequency of particle x in compartment z, and $\beta(x,y)$ is the probability measure on the space of fragments y for each particle x. The indicator function states that the mass of fragment y cannot be larger than the parent particle x. The probability measure β fulfils the symmetry property:

$$\beta(x,y) = \beta(x,x-y), \tag{3}$$

and $\beta(x,y)\mathrm{d}y$ gives the probability that the type of a particle formed from the fragmentation of a particle of type x is within the space of $[y, y+\mathrm{d}y]$.

Therefore, the fragmentation measure F satisfies the symmetry assumption

$$\int_{\mathbb{X}} \varphi(z,y)F(x,\mathrm{d}y) = \int_{\mathbb{X}} \varphi(z,x-y)F(x,\mathrm{d}y) \tag{4}$$

for measurable functions φ bounded on each interval $\{(x,y) \in \mathbb{X} : m(y) < m(x)\}$ (for each pair of (x,y), the mass of y has to be less than the mass of x).

The rate $g(z,x)$ is dependent on both the type of the particle and its spatial position but the probability measure of the fragments $\beta(x,y)$ is only dependent on the type of the particle and independent of the particle's spatial position.

2.1.3 Inception

The third term in Equation (1) represents the inception process. Particles of type x are incepted in compartment z at rate $I(z,\mathrm{d}x)$.

2.1.4 Flow of particles

The last two terms represent the movement of particles within the domain $\mathbb{Z}$. Each compartment is given a residence time $\tau(z)$ which controls the flowrate of the particles within the system. The flow of particles is independent of the particles' type and each particle in compartment z leaves at the rate $1/\tau(z)$.

The fourth term in Equation (1) represents the inflow process of compartment z. In this term, $N_{\text{in}}(z)$ is the number of inflows connected to compartment z and $z^{[j]}$ are the compartments which have particles flowing into z. The term $f_{\text{flow}}(z^{[j]} \to z)$ is defined as the fraction of flow from compartment $z^{[j]}$ which is connected to z. The superscript $[j]$ is a counting index for the compartments connected to z.

The fifth term in Equation (1) simply represents the disappearance of particles through outflow.

3 Stochastic particle methods

Each compartment z is simulated with a system of stochastic particles which describes the population dynamics (1):

$$(z, x_j, w_j), \quad j = 1, \ldots, N(t), \tag{5}$$

where $z \in \mathbb{Z}$, $x_j \in \mathbb{X}$, and $w_j \in (0, w_{\max}]$ is the statistical weight. The measure valued solution P for each compartment z is approximated by a sequence of particle systems parameterised by $n(z)$ which works in the sense that

$$\frac{1}{n(z)} \sum_{j=1}^{N(t)} w_j \, \varphi(z, \, x_j) \xrightarrow[n(z)\to\infty]{} \int_{\mathbb{X}} \varphi(z,x) P(t,z,\mathrm{d}x). \tag{6}$$

The normalisation parameter $n(z)$ is initialised such that

$$\frac{1}{n(z)} \sum_{j=1}^{N(0)} w_j \approx \int_{\mathbb{X}} P(0,z,\mathrm{d}x). \tag{7}$$

The approximation (6) may be interpreted as saying that the physical concentration of particles of type x_j in compartment z is approximately $w_j/n(z)$ and the total concentration of particles in compartment z is approximately $\sum w_j/n(z)$.

The number of particles, N, in each compartment is controlled by a parameter $N_{\max}$ which controls the maximum number of particles. Throughout this work, $N_{\max}$ is set to be equal in each compartment. When the number of computational particles in a compartment falls below 37.5%, each particle in the compartment is duplicated and the corresponding normalisation parameter $n(z)$ is doubled. On the other hand, when $N = N_{\max}$, a uniformly selected particle is deleted and $n(z)$ is rescaled by:

$$n_{\text{new}} = n_{\text{old}} \frac{N_{\max} - 1}{N_{\max}}. \tag{8}$$

As the systems investigated in this work feature frequent exchange of mass and inception, the simulations are initialised with $N(0) = 0.75 N_{\max}$ particles in each compartment to minimise the number of particle deletions.

3.1 Direct simulation algorithm

The Direct Simulation Algorithm (DSA) is so named because it can be interpreted as directly reproducing the behaviour of the particle system. Direct simulation is the case when $w_j(t) = 1$.

3.1.1 Inception

A new computational particle

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

is added to compartment z at the rate

$$n(z)I(z, \mathrm{d}x). \tag{10}$$

3.1.2 Coagulation

Coagulation jump processes are of the form:

$$(z, x, 1), (z, y, 1) \to (z, x+y, 1). \tag{11}$$

Each of these jumps occur at the rate

$$\frac{K(z, x, y)}{n(z)}. \tag{12}$$

3.1.3 Flow of particles

The simulation algorithm for the flow of particles considers the inflow and outflow terms in Equation (1) simultaneously, but can be better understood by focusing on the inflow term:

$$\sum_{j=1}^{N_{\text{in}}(z)} \int_{\mathbb{X}} \varphi(z^{[j]}, x) f_{\text{flow}}(z^{[j]} \to z) \frac{1}{\tau(z^{[j]})} P(t, z^{[j]}, \mathrm{d}x).$$

So, for every compartment $z^{[j]}$ with one or more outflow, each particle leaves compartment $z^{[j]}$ at the rate

$$\frac{1}{\tau(z^{[j]})}, \tag{13}$$

and this process is simulated by jump processes of the form

$$(z^{[j]}, x, 1) \to (z, x, 1)_{[1]}, (z, x, 1)_{[2]}, \ldots, (z, x, 1)_{[n_{\text{c}}]}. \tag{14}$$

For cases where $z^{[j]}$ has multiple outflows, the outflow compartment z is chosen with probability $f_{\text{flow}}(z^{[j]} \to z)$. In this process, the location of particle x is changed from $z^{[j]}$ to z. Thus, it can be interpreted as the outflow of $z^{[j]}$ which contributes to the inflow of z. Hence, jump processes in the form of (14) consider the last two terms in Equation (1) simultaneously.

The number of copies to make, n_c, is determined by the ratio of the normalisation parameters F_c:

$$n_c = \begin{cases} \lfloor F_c \rfloor + 1 & \text{for } u < F_c - \lfloor F_c \rfloor \\ \lfloor F_c \rfloor & \text{otherwise,} \end{cases} \tag{15}$$

where

$$F_c = \frac{n(z)}{n(z^{[j]})}, \tag{16}$$

and u is a uniformly chosen random number such that $u \in [0,1]$.

3.1.4 Fragmentation

Each particle x undergoes the following jump process:

$$(z,x,1) \to (z,y,1),(z,x-y,1) \tag{17}$$

at rate $g(z,x)$ and the fragment y is selected according to the probability measure $\beta(x,y)$.

3.2 Stochastic weighted algorithms

In the SWAs, the particle weights are no longer identically one. The jump dynamics of each process are presented in this section and the convergence of the SWAs is proven in Section 3.3. Three different SWAs (SWA1, SWA2 and SWA3) are presented in this work, each of which leads to a different simulation trajectory. The SWAs are mostly similar and only differ in the way fragmentation is executed. Despite that, it is found that the numerical properties of the algorithms are significantly different and this is shown in Section 4.

3.2.1 Inception

The inception process is similar to the DSA where a new computational particle

$$(z,x,1) \tag{18}$$

is added to compartment z at the rate

$$n(z)I(z,\mathrm{d}x). \tag{19}$$

A more complex inception process that depends on the weights is possible, but is not pursued here.

3.2.2 Coagulation

Coagulation jumps are of the form:

$$(z,x,w_x),(z,y,w_y) \to (z,x+y,w_{x+y}),(z,y,w_y), \tag{20}$$

and each of these jumps occur with the rate

$$\frac{K(z,x,y)w_y}{n(z)}. \tag{21}$$

Hence, the collision kernel for weighted particles $\widehat{K}$ is defined as

$$\widehat{K}(z,x,w_x,y,w_y) = K(z,x,y)w_y \tag{22}$$

and is no longer symmetric.

The weight w_{x+y} is calculated as:

$$\begin{aligned} w_{x+y} &= w_x \gamma_{\text{coag}}(x,w_x,y,w_y) \\ &= w_x \frac{m(x)}{m(x+y)}, \end{aligned} \tag{23}$$

where γ_{coag} represents the coagulation weight transfer function. Note that only the first particle x is modified and the second particle y is left unchanged for both cases. The definition of γ_{coag} used here is one of many possibilities and other mathematically valid weight transfer functions are studied in [26].

3.2.3 Flow of particles

Similarly as the DSA, each particle leaves compartment $z^{[j]}$ at the rate

$$\frac{1}{\tau(z^{[j]})}, \tag{24}$$

but the process is simulated with a slightly different jump process:

$$(z^{[j]},x,w_x) \to (z,x,w_x \times F_{\text{c}}), \tag{25}$$

where F_{c} is defined similarly as (16). The presence of statistical weights allows the number of particles to be scaled accurately and it is not necessary to determine the number of copies randomly (cf. (15)) like the DSA. Similarly as the DSA again, the outflow compartment z is chosen with probability $f_{\text{flow}}(z^{[j]} \to z)$.

3.2.4 Fragmentation

Fragmentation jump processes for the SWAs occur in the form

$$(z,x,w_x) \to (z,y,w_y) \tag{26}$$

at the rate

$$g(z,x). \tag{27}$$

The fragment particle (y, w_y) is selected according to the probability measure

$$\frac{w_x}{w_y}\beta(x,y) \tag{28}$$

for each pair of (x, w_x). Hence,

$$\frac{w_x}{w_y}\beta(x,y)\mathrm{d}y\mathrm{d}w_y \tag{29}$$

gives the probability that the type a particle formed from the fragmentation of a particle (x, w_x) is within the space of $[y, y+\mathrm{d}y]$ and its statistical weight is in the range $[w_y, w_y + \mathrm{d}w_y]$. Although $\beta(x,y)$ is symmetric (cf. (3)), this probability measure is no longer symmetric with the multiplication of w_x/w_y.

So, the fragmentation kernel for weighted particles, $\widehat{F}$, is

$$\begin{aligned}\widehat{F}(z,x,w_x,\mathrm{d}y,\mathrm{d}w_y) &= \mathbb{1}\{m(y) < m(x)\}\; g(z,x)\frac{w_x}{w_y}\beta(x,y)\mathrm{d}y\mathrm{d}w_y \\ &= \frac{w_x}{w_y}F(z,x,\mathrm{d}y)\mathrm{d}w_y.\end{aligned} \tag{30}$$

Here, w_y is defined by

$$\begin{aligned}w_y &= \gamma_{\text{frag}}(x,w_x,y) \\ &= w_x\alpha(x,y)\end{aligned} \tag{31}$$

where γ_{frag} is the fragmentation weight transfer function and its main function is to keep the number of computational particles constant while maintaining the approximation (6).

The definition of α is restricted by the following equation:

$$\frac{1}{\alpha(x,y)} + \frac{1}{\alpha(x,x-y)} = 1. \tag{32}$$

The justification of this apparently arbitrary ansatz is that it recovers the symmetry nature of the binary breakage process (3) from the asymmetric simulation process (cf. (28)) and leads to the correct limiting behaviour (6).

Many weight transfer functions γ_{frag} satisfying (32) can be constructed. From a physical point of view, a particle splits into two fragments in a binary fragmentation process. Since the jump (26) does not increase the number of particles, one expects the statistical weight of the resulting fragment to be doubled:

$$\gamma_{\text{frag}} = 2w_x, \tag{33}$$

which corresponds to $\alpha(x,y) = 2$ and this is the simplest possible solution.

It is also possible to define γ_{frag} to conserve any quantity during fragmentation. Let $M > 0$ be any quantity conserved during fragmentation, that is,

$$M(x) = M(y) + M(x-y), \quad \forall x, y \in \mathbb{X}. \tag{34}$$

The change in the amount of M as a result of the jump (26) is

$$w_x M(x) - \gamma_{\text{frag}}(x, w_x, y) M(y), \tag{35}$$

which is 0 if and only if

$$\gamma_{\text{frag}}(x, w_x, y) = w_x \frac{M(x)}{M(y)}. \tag{36}$$

Hence, any quantity $M > 0$ can be conserved across fragmentation jumps by defining γ_{frag} according to (36) which corresponds to $\alpha(x,y) = M(x)/M(y)$.

Table 1 shows two definitions of γ_{frag} which are studied in this work. The algorithm using the first definition shall be referred to as "SWA1" and the algorithm using the second definition shall be referred to as "SWA2".

Table 1: *Definitions of the numerically tested fragmentation weight transfer functions.*

Algorithm	$\gamma_{\text{frag}}(x, w_x, y)$
SWA1	$w_x \times 2$
SWA2	$w_x \times [m(x)/m(y)]$

3.2.5 Alternative method for fragmentation

It is also possible to perform the fragmentation of weighted particles using the "DSA" method and this method was used in [19]. Here, each particle x undergoes the following jump process:

$$(z, x, w_x) \to (z, y, w_x), (z, x-y, w_x) \tag{37}$$

at rate $g(z,x)$ and the fragment y is selected according to the probability measure $\beta(x,y)$. The fragments are given the same weight as the original particle x. This method is very easy to implement but it does not have the advantage of maintaining the number of computational particles constant and it is not suitable for simulations with high fragmentation rates.

The main problem with this method arises when there is no empty space left in the ensemble during a fragmentation jump. As a result, a particle has to be removed from the ensemble in order to accommodate the new fragment particle and this will contribute to some errors in the solution.

The algorithm using this approach shall be referred as "SWA3".

3.3 Convergence

For the DSA, general convergence results [10, 11] imply that, for continuous functions φ with compact support,

$$\frac{1}{n(z)}\sum_{j=1}^{N(t)}\varphi(z,x_j)\xrightarrow[n(z)\to\infty]{}\int_{\mathbb{X}}\varphi(z,x)P(t,z,\mathrm{d}x). \tag{38}$$

Similarly, one sees that

$$\frac{1}{n(z)}\sum_{j=1}^{N(t)}\psi(z,x_j,w_{x_j})\xrightarrow[n(z)\to\infty]{}\int_{E}\psi(z,x,w_x)Q(t,z,\mathrm{d}x,\mathrm{d}w_x), \tag{39}$$

where $E=\mathbb{X}\times(0,w_{\max}]$ is an extended state space for the particles and ψ is continuous with compact support. For the SWAs, the solution $Q(t,z,\mathrm{d}x,\mathrm{d}w)$ is defined through the following equation which is based on the jump dynamics presented in Section 3.2.

For every $z\in\mathbb{Z}$:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\int_E\psi(z,x,w)Q(t,z,\mathrm{d}x,\mathrm{d}w_x) = \\
&\int_E\int_E\left[\psi(z,x+y,w_{x+y})-\psi(z,x,w_x)\right]\widehat{K}(z,x,w_x,y,w_y)Q(t,z,\mathrm{d}x,\mathrm{d}w_x)Q(t,z,\mathrm{d}y,\mathrm{d}w_y)\\
&+\int_E\int_E\Big[\psi(z,y,w_y)-\psi(z,x,w_x)\Big]\widehat{F}(z,x,w_x,\mathrm{d}y,\mathrm{d}w_y)Q(t,z,\mathrm{d}x,\mathrm{d}w_x)\\
&+\int_{\mathbb{X}}\varphi(z,x)I(z,\mathrm{d}x)\\
&+\sum_{j=1}^{N_{\mathrm{in}}(z)}\int_E\psi(z^{[j]},x,w_x)f_{\mathrm{flow}}(z^{[j]}\to z)\frac{1}{\tau(z^{[j]})}Q(t,z^{[j]},\mathrm{d}x,\mathrm{d}w_x)\\
&-\int_E\psi(z,x,w_x)\frac{1}{\tau(z)}Q(t,z,\mathrm{d}x,\mathrm{d}w_x).
\end{aligned} \tag{40}$$

The collision and fragmentation terms are no longer halved because the kernels are not symmetric in this case. The fragmentation term in Equation (40) is formulated for the approach described in Section 3.2.4 (SWA1 & SWA2). Since the approach described in Section 3.2.5 (SWA3) follows the DSA method, it can be assumed that it has the correct approximation properties.

The following definitions:

$$P(t,z,\mathrm{d}x)=\int_0^{w_{\max}}w_xQ(t,z,\mathrm{d}x,\mathrm{d}w_x), \tag{41}$$

$$\psi(z,x,w_x)=w_x\varphi(z,x), \tag{42}$$

are used to show that (40) will eventually recover the population dynamics (1). Using these definitions, it is relatively straightforward to show that the last two terms in Equation (40) match the last two terms of Equation (1).

The collision term can be shown to have the correct approximation property by exploiting the symmetry of K, and the definitions of w_{x+y}, Q, and ψ from Equations (23, 41 - 42):

$$\begin{aligned}
&\int_E \int_E \left[\psi(z,x+y,w_{x+y}) - \psi(z,x,w_x)\right] \widehat{K}(z,x,w_x,y,w_y) Q(t,z,\mathrm{d}x,\mathrm{d}w_x) Q(t,z,\mathrm{d}y,\mathrm{d}w_y) \\
&= \int_E \int_E \left[\frac{m(x)}{m(x+y)}\varphi(z,x+y) - \varphi(z,x)\right] w_x w_y K(z,x,y) Q(t,z,\mathrm{d}x,\mathrm{d}w_x) Q(t,z,\mathrm{d}y,\mathrm{d}w_y) \\
&= \int_E \int_E \left[\frac{m(y)}{m(x+y)}\varphi(z,x+y) - \varphi(z,y)\right] w_x w_y K(z,x,y) Q(t,z,\mathrm{d}x,\mathrm{d}w_x) Q(t,z,\mathrm{d}y,\mathrm{d}w_y) \\
&= \frac{1}{2}\int_E \int_E \left[\varphi(z,x+y) - \varphi(z,x) - \varphi(z,y)\right] w_x w_y K(z,x,y) Q(t,z,\mathrm{d}x,\mathrm{d}w_x) Q(t,z,\mathrm{d}y,\mathrm{d}w_y) \\
&= \frac{1}{2}\int_{\mathbb{X}} \int_{\mathbb{X}} \left[\varphi(z,x+y) - \varphi(z,x) - \varphi(z,y)\right] K(z,x,y) P(t,z,\mathrm{d}x) P(t,z,\mathrm{d}y)
\end{aligned}$$

Similarly for the fragmentation term, it can be shown to have the correct approximation property by using the symmetry of F (4), the restriction of α (32), and the definitions of α, Q and ψ from Equations (31, 41 - 42):

$$\begin{aligned}
&\int_E \int_E \left[\psi(z,y,w_y) - \psi(z,x,w_x)\right] \widehat{F}(z,x,w_x,\mathrm{d}y,\mathrm{d}w_y) Q(t,z,\mathrm{d}x,\mathrm{d}w_x) \\
&= \int_E \int_E \left[\varphi(z,y) - \frac{1}{\alpha(x,y)}\varphi(z,x)\right] w_x F(z,x,\mathrm{d}y) \mathrm{d}w_y Q(t,z,\mathrm{d}x,\mathrm{d}w_x) \\
&= \int_E \int_E \left[\varphi(z,x-y) - \frac{1}{\alpha(x,x-y)}\varphi(z,x)\right] w_x F(z,x,\mathrm{d}y) \mathrm{d}w_y Q(t,z,\mathrm{d}x,\mathrm{d}w_x) \\
&= \frac{1}{2}\int_E \int_E \left[\varphi(z,y) + \varphi(z,x-y) - \varphi(z,x)\right] w_x F(z,x,\mathrm{d}y) \mathrm{d}w_y Q(t,z,\mathrm{d}x,\mathrm{d}w_x) \\
&= \frac{1}{2}\int_{\mathbb{X}} \int_{\mathbb{X}} \left[\varphi(z,y) + \varphi(z,x-y) - \varphi(z,x)\right] F(z,x,\mathrm{d}y) P(t,z,\mathrm{d}x).
\end{aligned}$$

Therefore, one sees that $P(t,z,\mathrm{d}x)$ as defined by (41) solves the population balance problem (1) with the appropriate weight transfer functions γ_{coag} and γ_{frag}.

4 Numerical studies

In stochastic simulations of particle population balance models, numerical error can occur in two forms: systematic error and statistical error. These errors can usually be reduced by increasing the maximum number of computational particles, $N_{\max}$, and the number of stochastic runs, L.

For the stochastic simulations in this work, the temporal evolution of a functional $M(t)$ is

averaged over the number of runs L:

$$\eta_1(t) = \frac{1}{L}\sum_{l=1}^{L} M(t), \tag{43}$$

where a functional refers to a property of the particle ensemble such as the moments defined in Table 3.

The empirical variance is:

$$\eta_2(t) = \frac{1}{L}\sum_{l=1}^{L} M(t)^2 - \eta_1(t)^2, \tag{44}$$

and the half-widths of the confidence intervals are estimated by:

$$c_P(t) = a_P\sqrt{\frac{\eta_2(t)}{L}}, \tag{45}$$

where P is the confidence level. In this work, the value $a_P = 1.64$ which corresponds to $P = 0.9$ is used.

4.1 Test system

The performances of the algorithms presented in Section 3 are first investigated by comparing the stochastic solutions with a test system which can be written as a series of ordinary differential equations (ODEs). The method adopted here is similar to the Discrete Model in [12, 22]. In the Discrete Model, each particle size is represented by an ODE. By choosing a sufficiently large maximum size which corresponds to the number of ODEs, relatively high-precision solutions can be obtained. Thus making it a reliable reference to validate the stochastic methods.

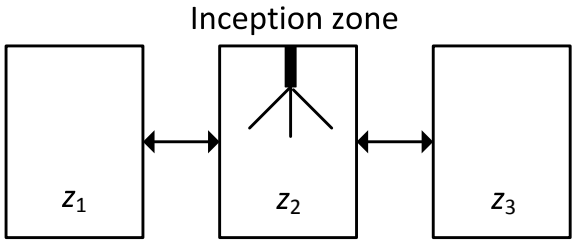


Figure 1: *Test system.*

The main purpose of the current work is to test the new fragmentation algorithms (SWA1 & SWA2) on a multi-compartment population balance model for high shear granulation [19]. As such, a test system is constructed to mimic the simulation conditions of the granulation model as closely as possible. The test system for this study is illustrated in Figure 1. It is a three-compartment model ($\mathbb{Z} = \{z_1, z_2, z_3\}$) with simultaneous inception, coagulation, and transport processes. It is solved with a simple particle model, $\mathbb{X} = \{1, 2, 3, \ldots\}$, and the particle mechanisms are described below:

- Coagulation:

$$K(z,x,y) = k_{\text{coag}}(z).$$

- Fragmentation:

$$g(z,x) = \begin{cases} x\, k_{\text{frag}}(z), & x \geq 5, \\ 0, & x < 5. \end{cases}$$
$$\beta(x,y) = \frac{1}{x-1}, \qquad y < x.$$

- Inception:

$$I(z,x) = \begin{cases} k_{\text{inc}}(z), & \text{if } x = 1, \\ 0, & \text{if } x \neq 1. \end{cases}$$

The population balance equation for the test system can be described with the following ODEs:

- For $x = 1$,

$$\begin{aligned} \frac{\mathrm{d}P(t,z,x)}{\mathrm{d}t} &= I(z,x) \\ &\quad - K(z,x,y)\, P(t,z,x) \sum_{y=1}^{N_{\text{ODE}}} P(t,z,y) \\ &\quad - g(z,x)P(t,z,x) + 2 \sum_{y=x+1}^{N_{\text{ODE}}} g(z,y)\beta(y,x)P(t,z,y) \\ &\quad + \sum_{j=1}^{N_{\text{in}}(z)} f_{\text{flow}}(z^{[j]} \to z) \frac{1}{\tau(z^{[j]})} P(t,z^{[j]},x) - \frac{1}{\tau(z)} P(t,z,x). \end{aligned}$$

- For $1 < x < N_{\text{ODE}}$,

$$\begin{aligned} \frac{\mathrm{d}P(t,z,x)}{\mathrm{d}t} &= I(z,x) \\ &\quad + \frac{1}{2} K(z,x,y) \sum_{y=1}^{x-1} P(t,z,y) P(t,z,x-y) - K(z,x,y) P(t,z,x) \sum_{y=1}^{N_{\text{ODE}}} P(t,z,y) \\ &\quad - g(z,x)P(t,z,x) + 2 \sum_{y=x+1}^{N_{\text{ODE}}} g(z,y)\beta(y,x)P(t,z,y) \\ &\quad + \sum_{j=1}^{N_{\text{in}}(z)} f_{\text{flow}}(z^{[j]} \to z) \frac{1}{\tau(z^{[j]})} P(t,z^{[j]},x) - \frac{1}{\tau(z)} P(t,z,x). \end{aligned}$$

- For $x = N_{\text{ODE}}$,

$$\begin{aligned}\frac{\mathrm{d}P(t,z,x)}{\mathrm{d}t} &= I(z,x) \\ &+ \frac{1}{2}K(z,x,y)\sum_{y=1}^{x-1}P(t,z,y)P(t,z,x-y) - K(z,x,y)P(t,z,x)\sum_{y=1}^{N_{\text{ODE}}}P(t,z,y) \\ &- g(z,x)P(t,z,x) \\ &+ \sum_{j=1}^{N_{\text{in}}(z)} f_{\text{flow}}(z^{[j]} \to z)\frac{1}{\tau(z^{[j]})}P(t,z^{[j]},x) - \frac{1}{\tau(z)}P(t,z,x).\end{aligned}$$

N_{ODE} is the number of ODEs for each compartment and it corresponds to the maximum particle size. So, the total number of ODEs solved simultaneously is equivalent to $3 \times N_{\text{ODE}}$ for this three-compartment system. For this test system, the appropriate value of N_{ODE} is primarily determined through trial and error, e.g. values around $N_{\text{ODE}} \approx 100$ are used at high fragmentation rates and values around $N_{\text{ODE}} \approx 1000$ are used at low fragmentation rates.

The test system is initialised with the following particle concentrations:

$$P(t,z_1,1) = P(t,z_2,1) = P(t,z_3,1) = 1, \tag{46}$$

and simulated for 10 seconds.

The connections between the compartments are defined by the following parameters:

$$\begin{aligned}&f_{\text{flow}}(z_1 \to z_2) = 1, \\ &f_{\text{flow}}(z_2 \to z_1) = f_{\text{flow}}(z_2 \to z_3) = 0.5, \\ &f_{\text{flow}}(z_3 \to z_2) = 1,\end{aligned} \tag{47}$$

and the rate parameters used for this test system are summarised in Table 2.

By conservation of mass:

$$\sum_{j=1}^{N_{\text{out}}(z)} f_{\text{flow}}(z \to z^{[j]}) = 1, \tag{48}$$

where $N_{\text{out}}(z)$ is the total number of outflows from compartment z. This condition is fulfilled by the defined connections (47).

From the generated solutions for P, the moments of the particle ensemble are calculated according to the definitions given in Table 3.

In typical simulations using the SWAs, computational particles are initialised with $w = 1$. However, in this study, the computational particles are given non-uniform weights: 50% of the particles are initialised with $w = 0.5$ and the other 50% are initialised with $w = 1$. This is to ensure that the simulations of the SWA2 algorithm do not follow the same simulation trajectories of the Mass Flow Algorithm [9, 10, 31] because the mass at each position is no longer equal with non-uniform weights.

Table 2: *Rate parameters for the test system.* k_{frag} *is set to be equal in each compartment but varied over the range* $0.001 - 1.0$.

Rate constants	Units	z_1	z_2	z_3
$k_{\text{coag}}(z)$	s^{-1}	0.1	1.0	10.0
$k_{\text{inc}}(z)$	s^{-1}	0.0	1.0	0.0
$\tau(z)$	s	5.0	2.5	5.0
$k_{\text{frag}}(z)$	s^{-1}	$0.001 - 1.0$	$0.001 - 1.0$	$0.001 - 1.0$

Table 3: *Definition of the moments for the test system.*

Moments	Formula
$M_0(t)$	$\sum P(t,z,x)$
$M_1(t)$	$\sum xP(t,z,x)$
$M_2(t)$	$\sum x^2P(t,z,x)$
$M_3(t)$	$\sum x^3P(t,z,x)$

4.2 Performances of the stochastic methods at different fragmentation rates

Initial tests of the fragmentation algorithms reveal that the accuracy is dependent on the number of fragmentation jumps performed in a simulation, especially with the SWA3 algorithm which is reported in [19] for a granulation population balance model.

A similar situation can be observed with the test system, which is shown in Figure 2. In this set of graphs, the stochastic solutions generated by SWA3 are compared against the ODE solutions at different fragmentation rates. It can be observed that the solutions generated with SWA3 become less accurate and increasingly noisy as the fragmentation rate increases which is consistent with the results in [19].

In order to compare the accuracies of the algorithms efficiently, the following representative errors of a simulation for a certain measured property (e.g. moments) are defined:

- Relative average systematic error,

$$\bar{e}_{\text{sys}} = \frac{1}{I}\sum_{i=1}^{I}\frac{|\mu(t_i) - M^*(t_i)|}{M^*(t_i)}, \tag{49}$$

- and relative average statistical error,

$$\bar{e}_{\text{stat}} = \frac{1}{I}\sum_{i=1}^{I}\frac{c_P(t_i)}{M^*(t_i)}, \tag{50}$$

where M^* represents the ODE solution and I is the number of time intervals. In the current numerical study, the errors are averaged over $I = 50$ time intervals. This means that the

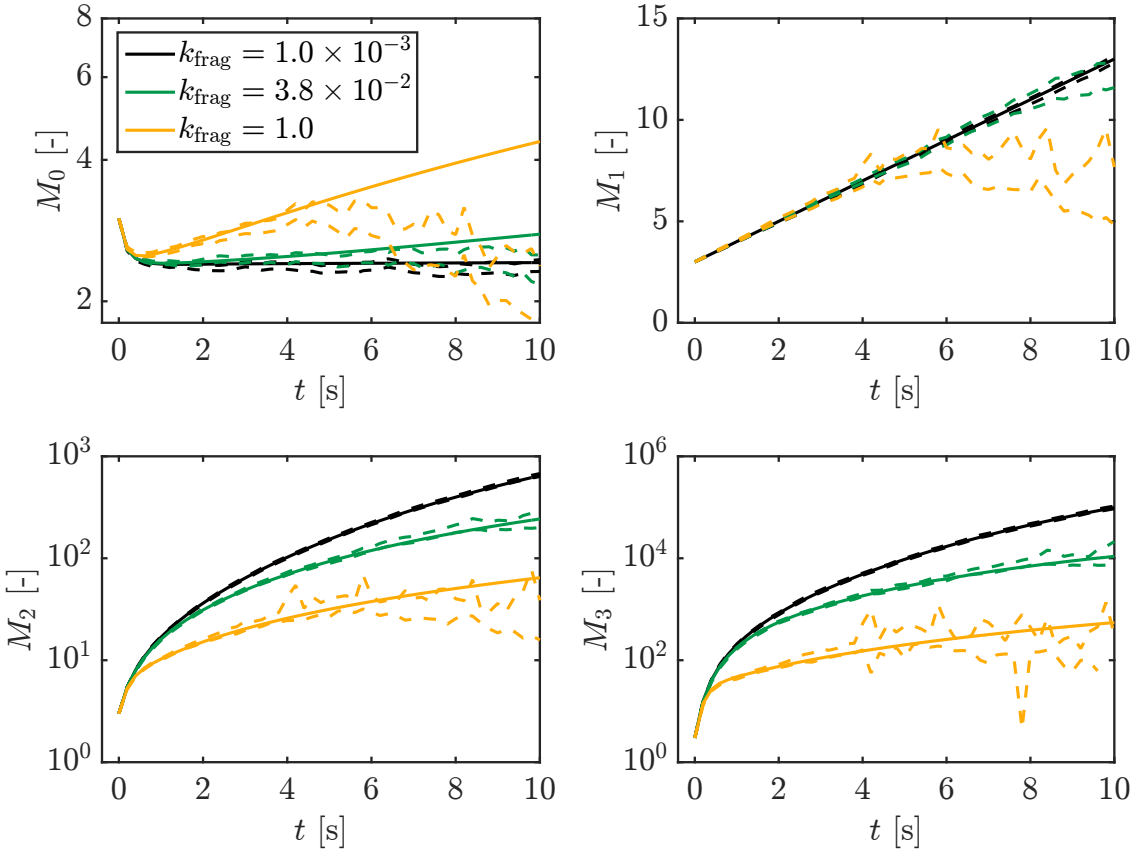


Figure 2: *Evolution of moments for different fragmentation rates. The solid lines are the ODE solutions and the dashed lines are solutions generated by SWA3 with* $N_{\text{max}} = 32$, $L = 100$. *The upper and lower dashed lines represent the confidence intervals of the stochastic solutions.*

values of the moments are calculated at every 0.2 seconds (the test system is simulated for 10 seconds) and compared to the ODE solutions at the corresponding time steps.

Figure 3 shows the statistical errors of the moments at different fragmentation rates (plotted as a function of the fragmentation rate constant, k_{frag}). Similar trends can be observed for most of the values in the plots except the errors of the DSA for M_0. It can be observed that the DSA is better at predicting M_0 compared to the other algorithms but it is not the case for the higher moments. For the higher moments (M_1, M_2 and M_3), the errors for the DSA are relatively higher at low fragmentation rates but they tend to decrease as the fragmentation rate increases. On the other hand, the errors for the SWA1 and SWA3 algorithms start off low but become higher as the fragmentation rate increases. SWA2 is perhaps the best performing algorithm based on this figure because the errors are almost independent of the fragmentation rate.

Though not as obvious as the statistical errors, similar trends can also be observed for the systematic errors, which is shown in Figure 4.

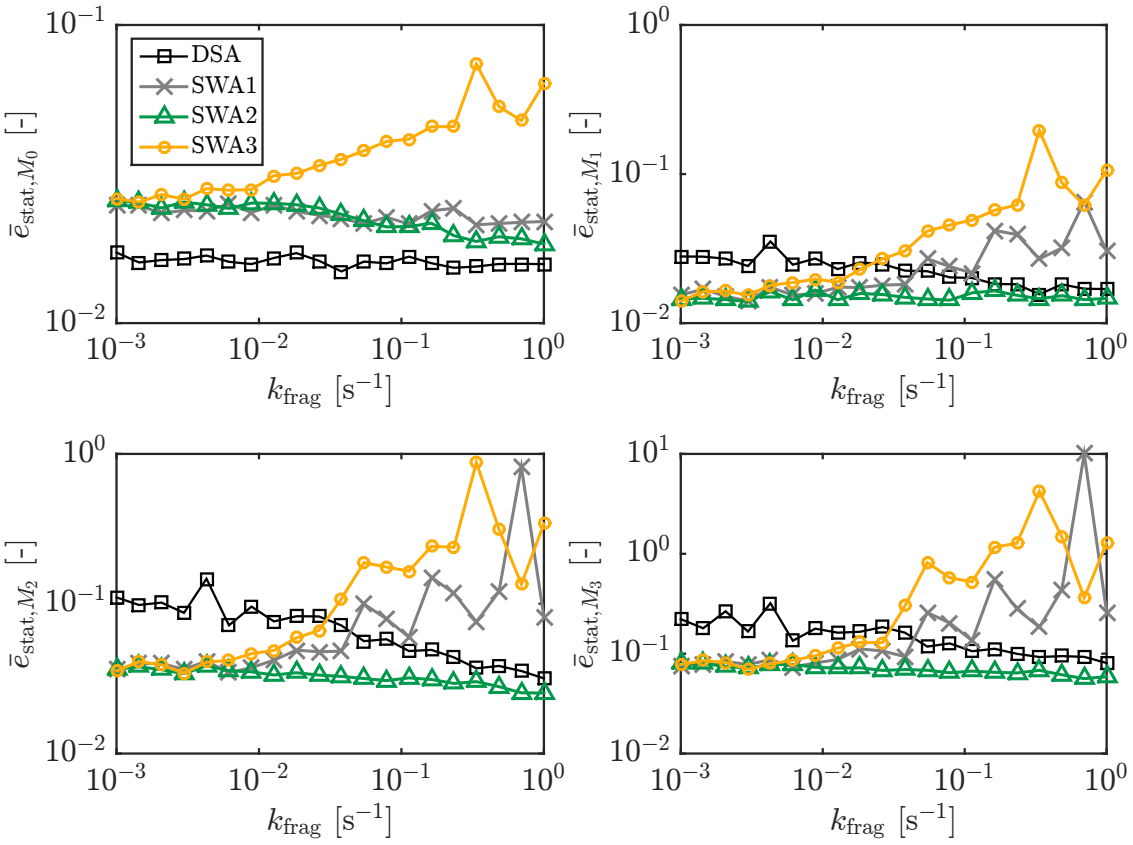


Figure 3: *Statistical errors as a function of* k_{frag}. *Simulations are performed with* $N_{\max} = 32$ *and* $L = 100$.

4.3 Computational efficiency

In Section 4.2, the errors of the stochastic solutions are investigated as a function of fragmentation rate. Other than showing that they are accurate, it is also important to ensure that they can achieve a reasonable accuracy using the shortest possible amount of CPU time (t_{CPU}).

In this section, the CPU times required by each algorithm to estimate M_2 with average systematic and statistical errors of $\bar{e} = 0.01$ (1%) are compared. As it is highlighted in Section 4.2 that the algorithms perform differently at different fragmentation rates, a detailed analysis of the efficiencies of the algorithms is carried out for $k_{\text{frag}} = 0.001$ (low fragmentation rate) and $k_{\text{frag}} = 1$ (high fragmentation rate).

The systematic and statistical errors of a stochastic simulation can be reduced by increasing the values of $N_{\max}$ and L but with a computational price. Here, the errors are evaluated as a function of $N_{\max}$ and the number of runs is fixed at $L = 100$.

Figure 5 presents the errors of the stochastic solutions as a function of $N_{\max}$. From these plots, the values of $N_{\max}$ required to achieve $\bar{e}_{\text{sys}} = 0.01$ and $\bar{e}_{\text{stat}} = 0.01$ for each algorithm are recorded in Table 4. Judging by the CPU times required to reduce the errors to about 1%, it is clear that the SWAs are much more efficient than the equivalent DSA simulation

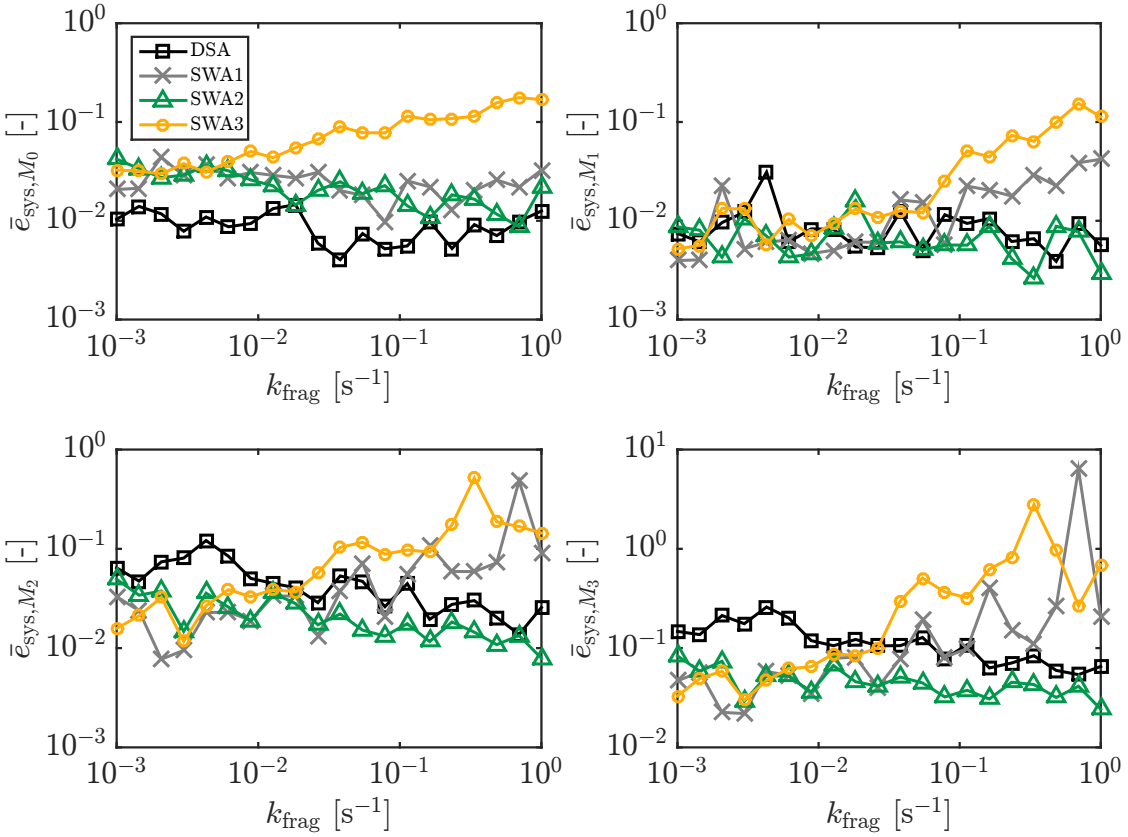


Figure 4: *Systematic errors as a function of* k_{frag}. *Simulations are performed with* $N_{\max} = 32$ *and* $L = 100$.

at this fragmentation rate.

The poor performance of the DSA at low fragmentation rate is mainly caused by the movement of some isolated large particles coupled with the way particles are moved across different compartments (cf. (14)). As the number of copies n_c is determined stochastically, some large particles might be deleted unintentionally during transport. On the other hand, the SWAs place more emphasis on the larger particles and the number of particles to transfer can be controlled accurately (cf. (25)), hence greatly reducing the errors from transport.

Similar results for $k_{\mathrm{frag}} = 1$ are recorded in Figure 6 and Table 5. In contrast with the results obtained at $k_{\mathrm{frag}} = 0.001$, it can be observed that the algorithms show different behaviours. First of all, the DSA and SWA2 algorithms are significantly more efficient compared to the SWA1 and SWA3 algorithms. The efficiency of the DSA at this rate is much higher because the system is mainly composed of small particles and the error caused by the movements of isolated large particles is reduced. The difference in performance between the SWA1 and SWA2 algorithms indicates that the choice of the fragmentation weight transfer function affects the quality of the simulation and careful consideration should be given to the choice of functions for different systems. The poor performance of the SWA3 algorithm is clearly due to the fact that the particle ensemble has to be rescaled many times to accommodate the new particles from fragmentation.

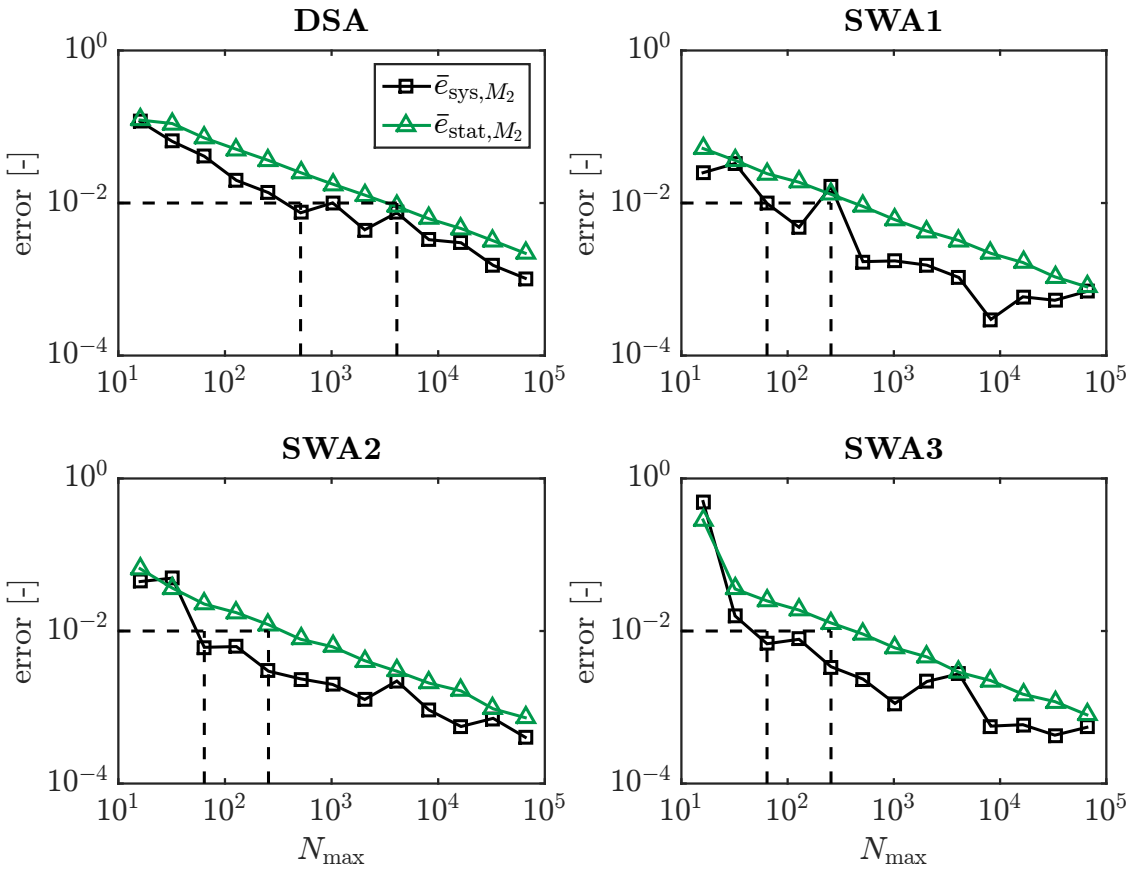


Figure 5: *Errors plotted against* $N_{\max}$ *for* $k_{\mathrm{frag}} = 1 \times 10^{-3}$. *The number of runs is fixed at* $L = 100$.

Table 4: $N_{\max}$ *and* t_{CPU} *required to achieve* $\bar{e} = 0.01$ *at* $k_{\mathrm{frag}} = 0.001$. *The results are obtained from Figure 5 and the number of runs is fixed at* $L = 100$.

k_{frag} [s^{-1}]	error [-]	Algorithm	$N_{\max}$ required [-]	t_{CPU} required [s]
1.0×10^{-3}	$\bar{e}_{\mathrm{sys},M_2} = 0.01$	DSA	512	5.06
		SWA1	64	0.95
		SWA2	64	0.95
		SWA3	64	0.92
1.0×10^{-3}	$\bar{e}_{\mathrm{stat},M_2} = 0.01$	DSA	4096	41.17
		SWA1	256	3.33
		SWA2	256	3.30
		SWA3	256	3.39

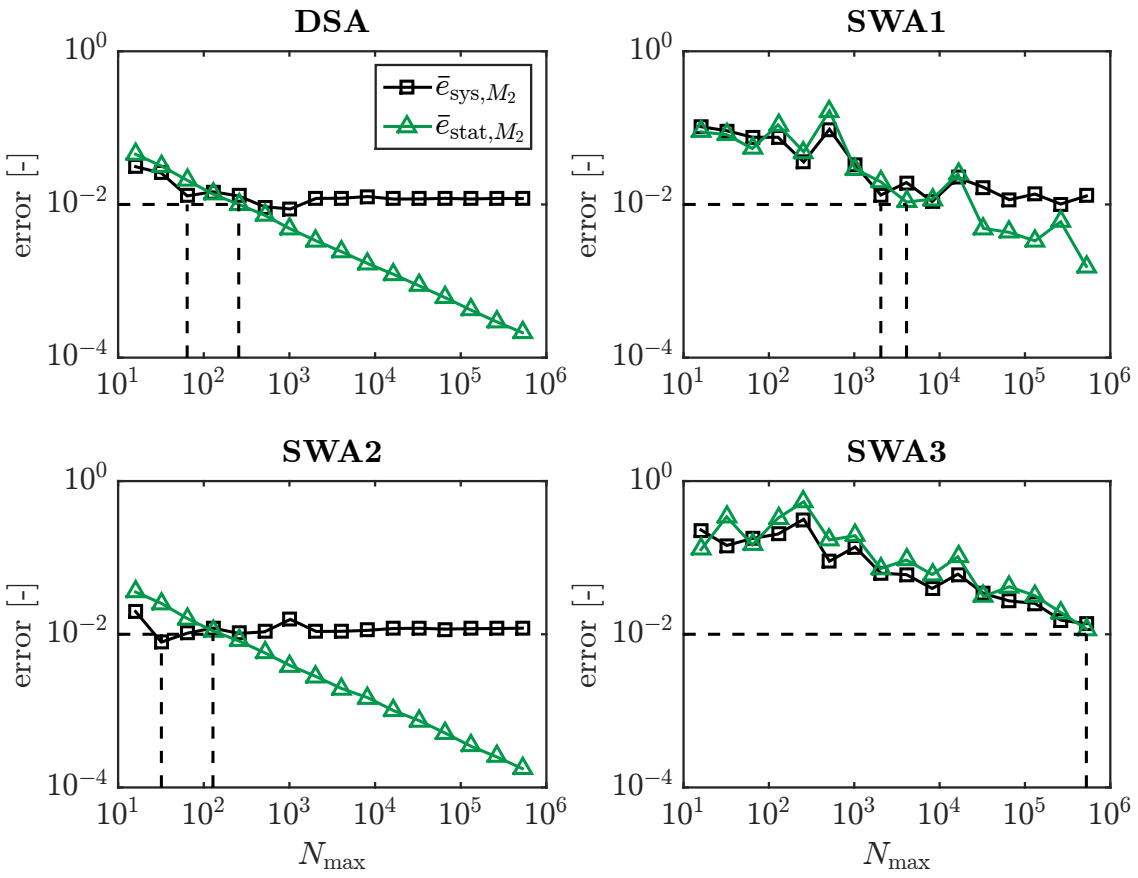


Figure 6: *Errors plotted against* $N_{\max}$ *for* $k_{\mathrm{frag}} = 1.0$. *The number of runs is fixed at* $L = 100$.

Table 5: $N_{\max}$ *and* t_{CPU} *required to achieve* $\bar{e} = 0.01$ *at* $k_{\mathrm{frag}} = 1$. *The results are obtained from Figure 6 and the number of runs is fixed at* $L = 100$.

k_{frag} [s^{-1}]	error [-]	Algorithm	$N_{\max}$ required [-]	t_{CPU} required [s]
1.0	$\bar{e}_{\mathrm{sys},M_2} = 0.01$	DSA	64	1.08
		SWA1	2048	40.91
		SWA2	32	0.64
		SWA3	524288	2.55×10^4
1.0	$\bar{e}_{\mathrm{stat},M_2} = 0.01$	DSA	256	2.85
		SWA1	4096	88.58
		SWA2	128	1.92
		SWA3	524288	2.55×10^4

5 Application to a realistic multidimensional model

In this section, a multidimensional population balance model for wet granulation is simulated with the algorithms described in Section 3. A typical granulation simulation is ini-

tialised with particles which represent the powder granules before adding liquid droplets (binder) so that the particles can undergo coagulation. After the desired amount of liquid is achieved, the simulation is allowed to proceed for a period of time and this phase is usually called the massing phase. For more information regarding the simulation conditions, readers can refer to [19].

The granulation model used here was initially developed as a single-compartment model [1–6] and extended to a multi-compartment model [19]. The development of the multi-compartment model was initially motivated by several unsuccessful attempts at predicting the experimentally measured size distributions accurately with the single-compartment configuration [16]. It was deduced that the main reason for the poor fit is that the single-compartment system assumes perfect mixing and ignores the effects of heterogeneity in powder mixing processes.

Here, the multi-compartment model described in [19] is modelled using the configuration shown in Figure 1 (three compartments) and the model is fitted to a set of experimental results from [16, Experiment B1]. As the focus of this work is the development of efficient fragmentation algorithms for weighted particles, the methodology used to optimise the model is not explained here and readers can refer to [16] for more details. The results are shown in Figure 7, which compares the best fit size distributions generated with the single-compartment and three-compartment systems. In this plot, f_{m} is defined as the mass fraction of particles with size d. It is clear that the multi-compartment system is significantly better in modelling this wet granulation system where the assumption of perfect mixing is poor.

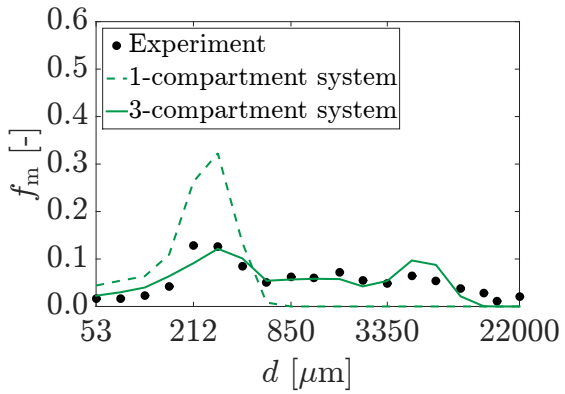


Figure 7: *Multi-compartment granulation model [19] fitted to a set of experimental data [16, Experiment B1]. Both systems are simulated with* $N_{\max} = 1024$ *and* $L = 100$ *using SWA2.*

The multi-compartment model was previously simulated using SWA3 in [19] which requires particles to be uniformly deleted when the ensemble is full. Although the error from transporting particles was reduced by using weighted particles (cf. (25)), significant numerical error can still be observed when the rate of fragmentation is high. Thus, this leads to the development of the SWA1 and SWA2 algorithms which do not increase the number of particles in a fragmentation jump.

As the advantage of using weighted particles is not significant with the single-compartment

system, only the numerical studies for the three-compartment system are shown here. Throughout this section, the three-compartment system used to generate the size distribution in Figure 7 is used for the following numerical studies.

The numerical performances of the DSA and SWAs in predicting the size distributions are compared in Figure 8. Each graph in this figure presents two sets of results: the solid lines are the high-precision solutions which require enormous CPU times ($t_{\text{CPU}} \approx 12-24$ hours), and the points are solutions generated with reasonable CPU times ($t_{\text{CPU}} \approx 5-10$ minutes). The high-precision solutions are generated with $N_{\text{max}} = 1,048,576$ and the SWAs are simulated with $N_{\text{max}} = 1024$. On the other hand, the DSA simulation is performed at $N_{\text{max}} = 2048$ so that its CPU time is roughly equal to the equivalent SWA simulations.

The first thing to observe from these plots is that the solutions of the different methods converge to the same solution as $N \to \infty$. However, since the points are solutions from simulations with equal computational effort, it can be concluded that the SWA1 and SWA2 algorithms are more efficient compared to the DSA and SWA3 algorithms. For the DSA and SWA3 results, significant systematic errors and statistical noise can be noticed in the solutions. The main causes for these errors are explained later with Figures 9 and 10.

One of the main problems with the DSA is the allocation of computational resources within the simulations. Previous studies involving the DSA often report that this method allocates huge amount of resources to simulate the unimportant smaller particles and this leads to noisy estimates in the larger particle regions that dominate mass concentration [8, 27, 29]. The same situation is also observed in this system and is shown in Figure 9. Here, the number fraction of computational particles for each size, f_{n}, is compared against the total mass fraction of the corresponding size which is the same as f_{m} shown in Figure 8, but plotted on logarithmic scale in Figure 9.

Looking at the plot for the DSA in Figure 9, it can be observed that most of the computational resources are wasted on the physically insignificant small particles - more than 50% of the computational particles are in the smallest size class which forms less than 5% by mass in the system. On the other hand, the number of computational particles for each size is directly proportional to the mass for the SWA1 and SWA2 simulations and this greatly reduces the noise in the larger particle regions. The distribution of particles in the SWA3 simulation is slightly peculiar due to the huge amount of particle deletions from the fragmentation jumps. The marginally higher number of particles in the mid-high size range is possibly caused by the fragmentation of the larger particles (large particles have higher fragmentation rates than the smaller particles [19]).

Figure 10 highlights a serious issue found in the DSA and SWA3 simulations. As a reference, a test run is performed by only adding liquid particles into the system without simulating any other processes, which is shown as the solid lines in the plots (a brief description of the simulation process is given at the beginning of this section). Each plot in this figure compares the total mass per volume, m_{total}, in the actual simulations with the values obtained with the test run. These plots show the time evolution of the total mass throughout the simulations. On the other hand, the size distributions in Figures 7 and 8 are evaluated at the end of the simulations, which correspond to $t = 420$ seconds in Figure 10.

Significant fluctuation in mass can be observed in the DSA simulation and the main con-

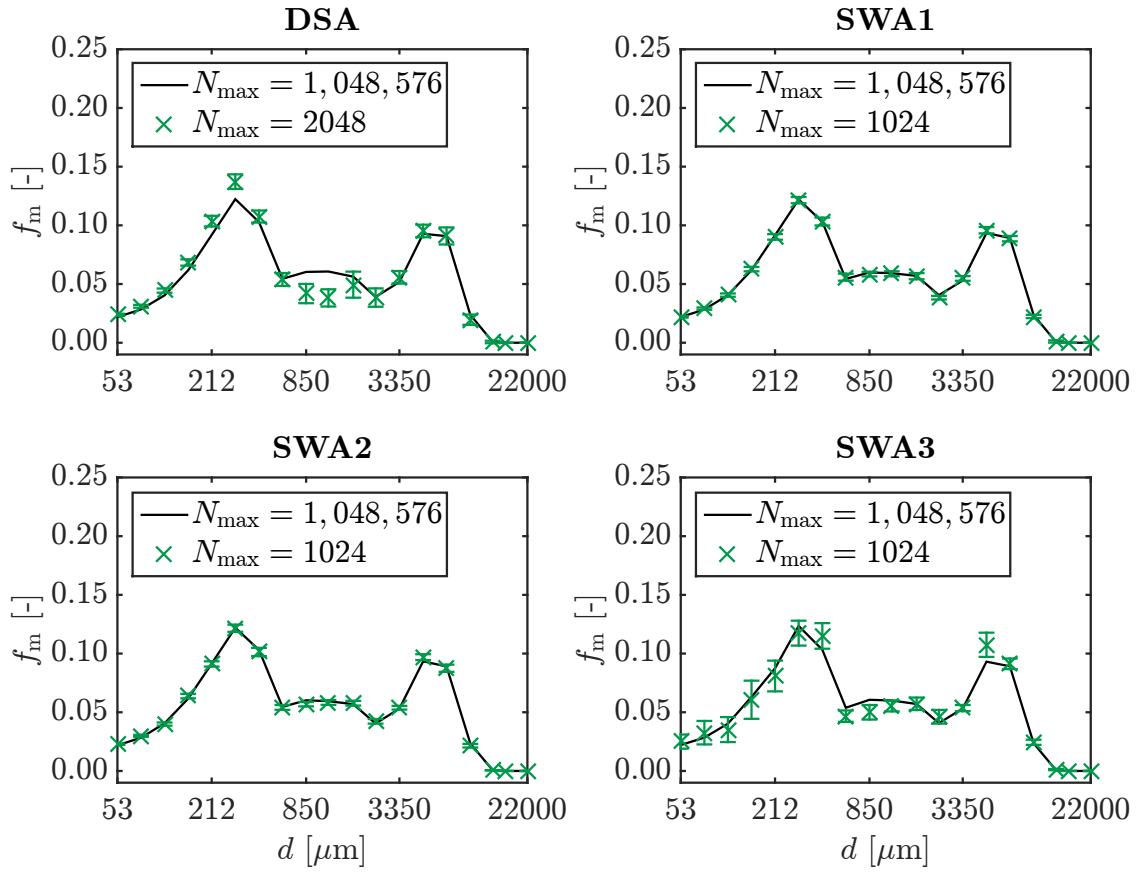


Figure 8: *Standard simulations ($N_{\max} = 1024 - 2048$, $L = 100$) compared against the high-precision solutions ($N_{\max} = 1,048,576$, $L = 10$) for the three-compartment system used to generate the size distribution in Figure 7. The DSA simulation is performed with $N_{\max} = 2048$ so that its CPU time is equivalent to the SWA simulations.*

tribution of these errors is from the way particles are transferred in the DSA as described in Section 4.3. Although the numerical errors are not that severe for the SWA3 algorithm, noticeable deviation from the reference mass can still be observed towards the end of the simulation when significant deletions of particles have been done. This problem is solved by using the new fragmentation algorithms which do not increase the number of computational particles (SWA1 & SWA2).

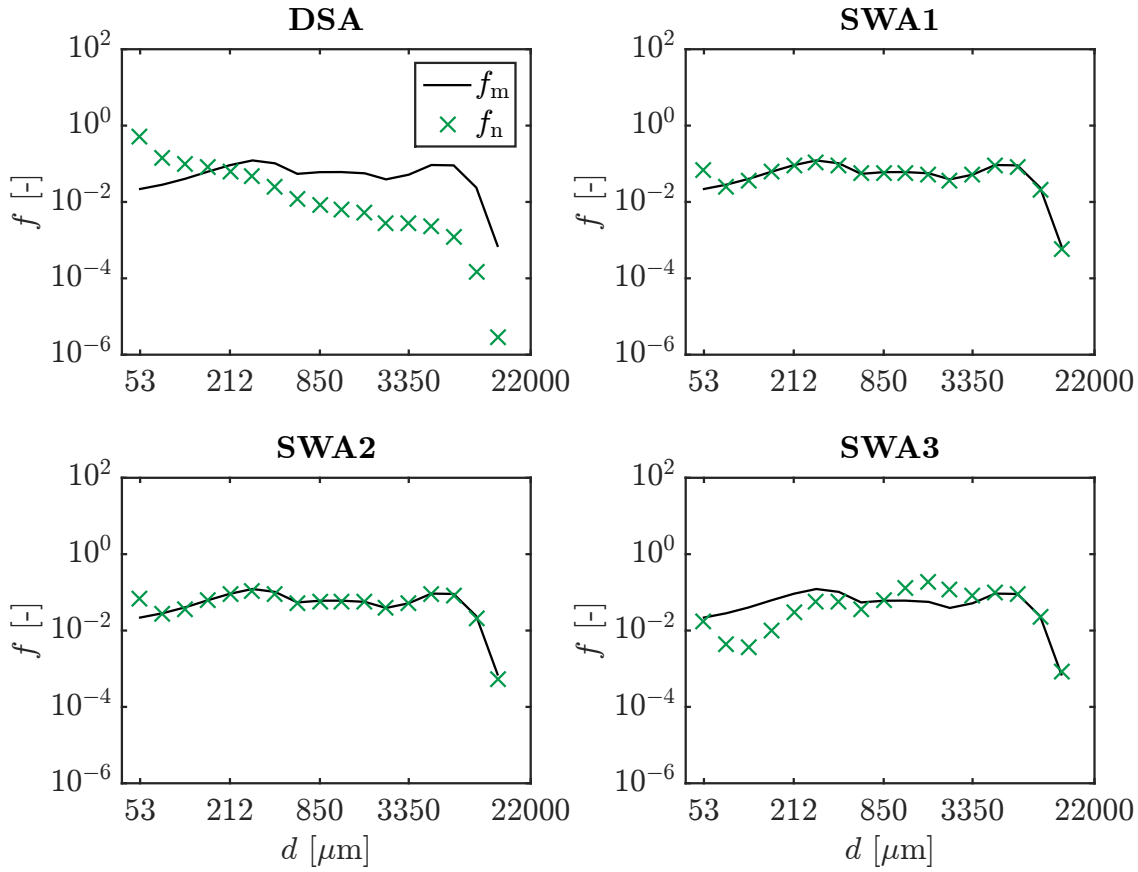


Figure 9: *Number fraction of computational particles f_{n} compared against the mass fraction f_{m}. Both f_{n} and f_{m} are obtained from the high precision solutions ($N_{\max} = 1,048,576$, $L = 10$).*

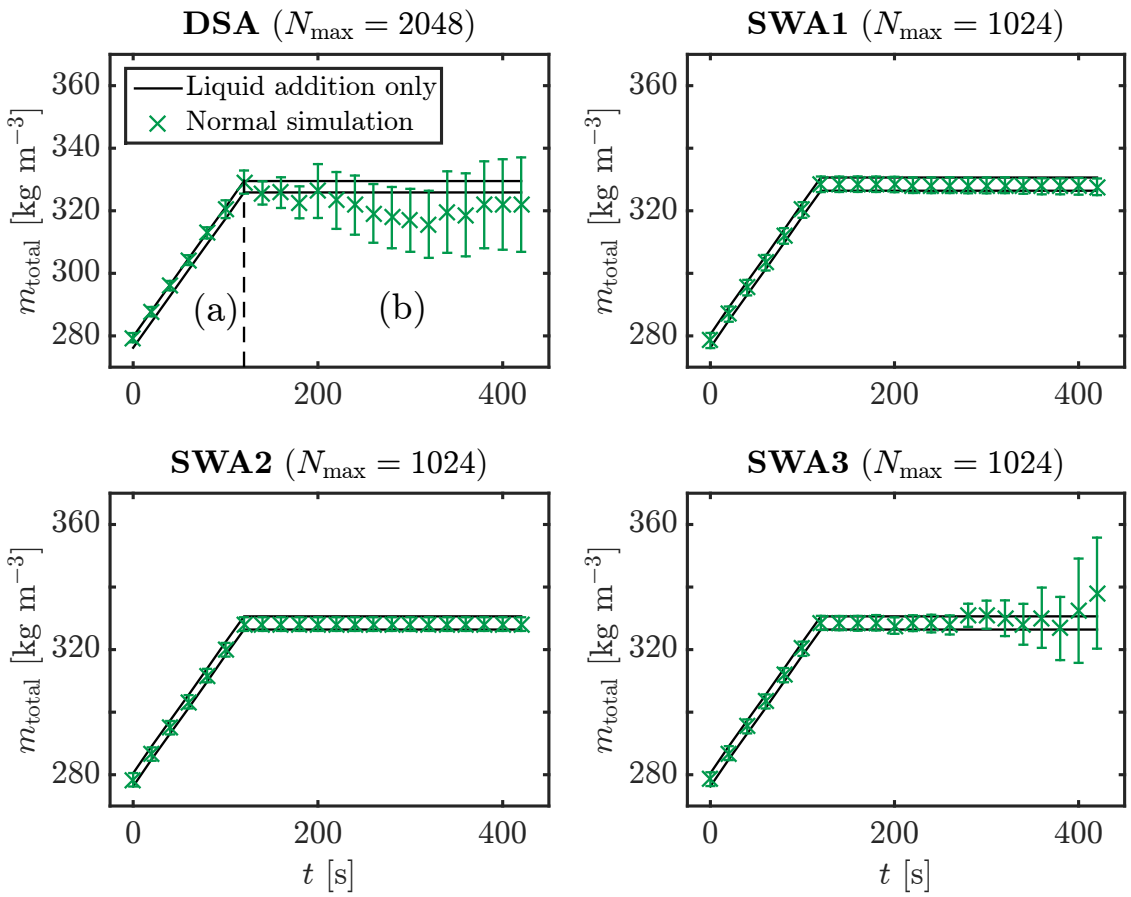


Figure 10: *Total mass per volume plotted against time in the three-compartment granulation model. The liquid addition and massing phases are indicated by '(a)' and '(b)' respectively in the DSA plot. The number of runs is fixed at* $L = 100$.

6 Conclusions

In this work, population balance problems with simultaneous coagulation, fragmentation, inception and spatial inhomogeneity are investigated.

Main focus is given to the development of efficient fragmentation algorithms for weighted particles. Three fragmentation algorithms are investigated, two of which are developed in this work (SWA1 & SWA2). Their performances are compared with an existing algorithm (SWA3) and the traditional direct simulation algorithm (DSA).

Overall, it is found that the new algorithms (SWA1 & SWA2) are superior compared to the old algorithm (SWA3), mainly because the new algorithms do not increase the number of computational particles in a fragmentation jump process. The SWA3 algorithm increases the number of particles in a fragmentation jump and the particle ensemble has to be rescaled frequently to accommodate the new fragment particles and this caused significant numerical errors in the solutions.

Another important finding in this work is that there is not a single superior algorithm which is suitable for any type of system. For example, the direct simulation algorithm performs poorly when the system contains a high number of large particles and its efficiency improves when the system mainly consists of small particles. Thus, the choice of a suitable algorithm should be made carefully for each individual system.

Lastly, the new fragmentation algorithms are applied to a realistic multidimensional model for wet granulation. The advantage of the new algorithms are shown by comparing the quality of the solutions generated at equal computational cost. It is shown that the new SWAs (SWA1 & SWA2) are able to produce more accurate results with less computational effort compared to the old SWA method (SWA3) and the direct simulation algorithm. It can be concluded that for systems which contain particles with a wide range of sizes such as the one investigated, stochastic weighted algorithms are preferable over the direct simulation algorithm.

Acknowledgements

This project is partly funded by the National Research Foundation (NRF), Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) Programme. MK gratefully acknowledges the support of the Weierstrass Institute for Applied Analysis and Stochastics (WIAS) in Berlin.

References

[1] A. Braumann and M. Kraft. Incorporating experimental uncertainties into multivariate granulation modelling. *Chemical Engineering Science*, 65(3):1088 – 1100, 2010. doi:10.1016/j.ces.2009.09.063.

[2] A. Braumann, M. J. Goodson, M. Kraft, and P. R. Mort. Modelling and validation of granulation with heterogeneous binder dispersion and chemical reaction. *Chemical Engineering Science*, 62(17):4717 – 4728, 2007. doi:10.1016/j.ces.2007.05.028.

[3] A. Braumann, P. L. Man, and M. Kraft. Statistical approximation of the inverse problem in multivariate population balance modeling. *Industrial & Engineering Chemistry Research*, 49(1):428–438, 2009. doi:10.1021/ie901230u.

[4] A. Braumann, M. Kraft, and P. R. Mort. Parameter estimation in a multidimensional granulation model. *Powder Technology*, 197(3):196 – 210, 2010. doi:10.1016/j.powtec.2009.09.014.

[5] A. Braumann, M. Kraft, and W. Wagner. Numerical study of a stochastic particle algorithm solving a multidimensional population balance model for high shear granulation. *Journal of Computational Physics*, 229(20):7672 – 7691, 2010. doi:10.1016/j.jcp.2010.06.021.

[6] A. Braumann, P. L. Man, and M. Kraft. The inverse problem in granulation modeling-two different statistical approaches. *AIChE Journal*, 57(11):3105–3121, 2011. doi:10.1002/aic.12526.

[7] E. Debry, B. Sportisse, and B. Jourdain. A stochastic approach for the numerical simulation of the general dynamics equation for aerosols. *Journal of Computational Physics*, 184(2):649 – 669, 2003. doi:10.1016/S0021-9991(02)00041-4.

[8] R. DeVille, N. Riemer, and M. West. Weighted Flow Algorithms (WFA) for stochastic particle coagulation. *Journal of Computational Physics*, 230(23):8427 – 8451, 2011. doi:10.1016/j.jcp.2011.07.027.

[9] A. Eibeck and W. Wagner. Approximative solution of the coagulation–fragmentation equation by stochastic particle systems. *Stochastic analysis and applications*, 18(6): 921–948, 2000. doi:10.1080/07362990008809704.

[10] A. Eibeck and W. Wagner. Stochastic particle approximations for Smoluchoski's coagulation equation. *Annals of Applied Probability*, pages 1137–1165, 2001. doi:10.1214/aoap/1015345398.

[11] A. Eibeck and W. Wagner. Stochastic interacting particle systems and nonlinear kinetic equations. *Annals of Applied Probability*, pages 845–889, 2003. doi:10.1214/aoap/1060202829.

[12] M. Frenklach and S. J. Harris. Aerosol dynamics modeling using the method of moments. *Journal of colloid and interface science*, 118(1):252–261, 1987. doi:10.1016/0021-9797(87)90454-1.

[13] F. Guiaş. A stochastic approach for simulating spatially inhomogeneous coagulation dynamics in the gelation regime. *Communications in Nonlinear Science and Numerical Simulation*, 14(1):204 – 222, 2009. doi:10.1016/j.cnsns.2007.07.015.

[14] M. J. Hounslow, R. L. Ryall, and V. R. Marshall. A discretized population balance for nucleation, growth, and aggregation. *AIChE Journal*, 34(11):1821 – 1832, 1988. doi:10.1002/aic.690341108.

[15] R. Irizarry. Fast compartmental Monte Carlo simulation of population balance models: Application to nanoparticle formation in nonhomogeneous conditions. *Industrial & Engineering Chemistry Research*, 51(47):15484 – 15496, 2012. doi:10.1021/ie3011116.

[16] C. A. Kastner, G. P. Brownbridge, S. Mosbach, and M. Kraft. Impact of powder characteristics on a particle granulation model. *Chemical Engineering Science*, 97 (0):282 – 295, 2013. doi:10.1016/j.ces.2013.04.032.

[17] M. Kraft. Modelling of particulate processes. *KONA Powder and Particle Journal*, 23:18–35, 2005. doi:10.14356/kona.2005007.

[18] K. Lee and T. Matsoukas. Simultaneous coagulation and break-up using constant-*N* monte carlo. *Powder Technology*, 110(1-2):82 – 89, 2000. doi:10.1016/S0032-5910(99)00270-3.

[19] K. F. Lee, S. Mosbach, M. Kraft, and W. Wagner. A multi-compartment population balance model for high shear granulation. *Computers & Chemical Engineering*, 75: 1–13, 2015. doi:10.1016/j.compchemeng.2015.01.009.

[20] K. Maruyama and Y. Fujiyoshi. Monte Carlo simulation of the formation of snowflakes. *Journal of the atmospheric sciences*, 62(5):1529–1544, 2005. doi:10.1175/JAS3416.1.

[21] W. J. Menz, R. I. Patterson, W. Wagner, and M. Kraft. Application of stochastic weighted algorithms to a multidimensional silica particle model. *Journal of Computational Physics*, 248(0):221 – 234, 2013. doi:10.1016/j.jcp.2013.04.010.

[22] W. J. Menz, J. Akroyd, and M. Kraft. Stochastic solution of population balance equations for reactor networks. *Journal of Computational Physics*, 256(0):615 – 629, 2014. doi:10.1016/j.jcp.2013.09.021.

[23] N. Morgan, C. Wells, M. Kraft, and W. Wagner. Modelling nanoparticle dynamics: coagulation, sintering, particle inception and surface growth. *Combustion Theory and Modelling*, 9(3):449–461, 2005. doi:10.1080/13647830500277183.

[24] N. Morgan, C. Wells, M. Goodson, M. Kraft, and W. Wagner. A new numerical approach for the simulation of the growth of inorganic nanoparticles. *Journal of Computational Physics*, 211(2):638 – 658, 2006. doi:10.1016/j.jcp.2005.04.027.

[25] R. I. Patterson and W. Wagner. A stochastic weighted particle method for coagulation–advection problems. *SIAM Journal on Scientific Computing*, 34(3): B290–B311, 2012. doi:10.1137/110843319.

[26] R. I. Patterson, W. Wagner, and M. Kraft. Stochastic weighted particle methods for population balance equations. *Journal of Computational Physics*, 230(19):7456 – 7472, 2011. doi:10.1016/j.jcp.2011.06.011.

[27] S. Rjasanow and W. Wagner. A stochastic weighted particle method for the boltzmann equation. *Journal of Computational Physics*, 124(2):243–253, 1996. doi:10.1006/jcph.1996.0057.

[28] M. Sander, R. I. Patterson, A. Braumann, A. Raj, and M. Kraft. Developing the PAH-PP soot particle model using process informatics and uncertainty propagation. *Proceedings of the Combustion Institute*, 33(1):675 – 683, 2011. doi:10.1016/j.proci.2010.06.156.

[29] Y. Sentoku and A. Kemp. Numerical methods for particle simulations at extreme densities and temperatures: Weighted particles, relativistic collisions and reduced currents. *Journal of Computational Physics*, 227(14):6846 – 6861, 2008. doi:10.1016/j.jcp.2008.03.043.

[30] S. Shima, K. Kusano, A. Kawano, T. Sugiyama, and S. Kawahara. The super-droplet method for the numerical simulation of clouds and precipitation: a particle-based and probabilistic microphysics model coupled with a non-hydrostatic model. *Quarterly Journal of the Royal Meteorological Society*, 135(642):1307–1320, 2009. doi:10.1002/qj.441.

[31] W. Wagner. Stochastic, analytic and numerical aspects of coagulation processes. *Mathematics and Computers in Simulation*, 62(3-6):265 – 275, 2003. doi:10.1016/S0378-4754(02)00236-7. 3rd IMACS Seminar on Monte Carlo Methods.

[32] H. Zhao and C. Zheng. Correcting the multi-Monte Carlo method for particle coagulation. *Powder Technology*, 193(1):120 – 123, 2009. doi:10.1016/j.powtec.2009.01.019.

[33] H. Zhao, C. Zheng, and M. Xu. Multi-Monte Carlo approach for general dynamic equation considering simultaneous particle coagulation and breakage. *Powder Technology*, 154(2-3):164 – 178, 2005. doi:10.1016/j.powtec.2005.04.042.

[34] H. Zhao, F. E. Kruis, and C. Zheng. A differentially weighted Monte Carlo method for two-component coagulation. *Journal of Computational Physics*, 229(19):6931 – 6945, 2010. doi:10.1016/j.jcp.2010.05.031.