A high-dimensional, stochastic model for twin-screw granulation
Part 2: Numerical methodology

Andrew D. McGuire\textsuperscript{1}, Sebastian Mosbach\textsuperscript{1}, Kok Foong Lee\textsuperscript{1},
Gavin Reynolds\textsuperscript{2}, Markus Kraft\textsuperscript{1,3}

released: 17 October 2017

\textsuperscript{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

\textsuperscript{2} Pharmaceutical Technology & Development
AstraZeneca
Charter Way
Macclesfield, SK10 2NA
United Kingdom
E-mail: gavin.reynolds@astrazeneca.com

\textsuperscript{3} School of Chemical
and Biomedical Engineering
Nanyang Technological University
62 Nanyang Drive
Singapore 637459
E-mail: mk306@cam.ac.uk

Preprint No. 192

Keywords: granulation, twin-screw, stochastic particle model, population balance
Abstract

In the second part of this study, we present the stochastic weighted particle population balance framework used to solve the twin-screw granulation model detailed in the first part of this study. Each stochastic jump process is presented in detail, including a new nucleation jump event capable of capturing the immersion nucleation processes in twin-screw granulation. A variable weighted inception algorithm is presented and demonstrated to reduce the computational cost of simulations by up to two orders of magnitude over traditional approaches. The relationship between the performance of the simulation algorithm and key numerical parameters within the nucleation jump process are explored and optimum operating conditions identified. Finally, convergence studies on the complete simulation algorithm demonstrate that the algorithm is very robust against changes in the number of stochastic particles used, provided that the number of particles exceeds a minimum required for numerical stability.

Highlights

- Four-dimensional, stochastic population balance model for twin-screw granulation.
- Stochastic implementation of an immersion nucleation mechanism
- Variable weighting algorithm for particle inception.
- Significant computational speed-up over traditional inception methods.
## Contents

1 Introduction 4

2 Stochastic particle methods for granulation 7
  2.1 Particle type-space 7
  2.2 Stochastic weighted algorithm 8
    2.2.1 Inception 8
    2.2.2 Nucleation 9
    2.2.3 Coagulation/compaction 11
    2.2.4 Breakage 12
    2.2.5 Transport 12
    2.2.6 Overall jump rate 13
    2.2.7 Continuous processes 13

3 Twin-screw granulation model 13
  3.1 Inception/initialisation 14
  3.2 Nucleation 15
  3.3 Coagulation 16
  3.4 Compaction 16
  3.5 Breakage 17
  3.6 Compartmentalisation/transport 17
  3.7 Penetration 17

4 Numerical studies 17
  4.1 Simulation details 18
    4.1.1 Simulation stop time and repetitions 18
    4.1.2 Quantification of errors 18
    4.1.3 Binary tree 19
    4.1.4 Hardware set-up 19
  4.2 Inception sampling methods 19
  4.3 Immersion nucleation parameters 24
    4.3.1 Selection of the nucleation weight 24
    4.3.2 Simplification of the jump 25
    4.3.3 Selection of ensemble particles 25
1 Introduction

In the first part of this study we introduced, optimised and investigated the qualitative behaviour of a four dimensional model for twin-screw granulation under various operating conditions. In this part of the study we are primarily concerned with presenting and investigating the properties of the numerical methods employed/developed to solve this high dimensional model and overcome the numerical challenges inherent to the stochastic modelling of twin-screw systems.

Granulation is the transformation of solid primary particles to agglomerate form. These agglomerates may have a size, porosity and liquid distribution that allow them to be utilised within the pharmaceutical and food industries, among others [26]. The granulation process is typically modelled mechanistically using population balance models (PBM) [39] (though non-mechanistic pure neural network approaches have also been investigated [9, 42]). Using PBMs, the particle ensemble is transformed through processes such as nucleation, coagulation, breakage, consolidation, layering and wetting.

In general, granulation PBMs are numerically solved using variations of the sectional method [5, 8, 28, 29]. Through the sectional method, each particle dimension (e.g. solid volume, liquid volume etc.) is mapped onto a discrete grid. This allows the model to be represented as a system of ordinary differential equations, which can be solved numerically. The dimensionality of these sectional PBMs generally ranges from one [29] to three [7, 8] and a lumped parameter technique is often employed to estimate additional particle properties (such as internal gas volume [5]). To capture spatial inhomegenieties in processing conditions within the equipment, granulation PBMs often represent the system as a network of well-mixed compartments [10, 29]. In recent years, PBMs have been coupled (both unidirectionally and bidirectionally) with other simulation frameworks, such as the Discrete Element Method (DEM) [19]. DEM is typically used to capture collision data and compartment residence times for use in the PBM(s) [4, 7, 32]. However, the high computational cost of DEM simulations has, in cases, incentivised their replacement with artificial neural networks (ANN). Barrasso and co-workers [6, 24] demonstrated the use of an ANN (trained on DEM data) in PBM-ANN couplings. The resulting couplings have been shown to reduce the computational cost of solution by several orders of magnitude (over PBM-DEM couplings), whilst maintaining the ability to predict key DEM data.

Though sectional models for granulation have been successfully applied to numerous systems in the past [8, 29], the method itself has a number of key limitations. The first of these limitations is the upper bound placed on the dimensionality of the particle model. In sectional solvers, this limit arises as the number of equations to be solved rapidly increases with the addition of each additional dimension. It is generally agreed that at most three dimensions can be specified before the solution process becomes computationally unfeasible [5]. Though, four dimensional sectional models have been tested, they can take several days or even weeks to solve [3]. This bound on dimensionality ultimately limits the number of ways that the modelled particles can be characterised and therefore limits the complexity of the particle mechanisms that can be employed. The second limitation of the sectional approach is that the resolution of the model along any particular dimension is controlled by the grid employed. All the particle mass within the same grid section is treated identically, meaning that there is a finite number of particles types than can be
An alternative method of solving population balance models is the Monte Carlo particle method (sometimes referred to as MC-PBM or stochastic PBM). MC-PBMs have been successfully applied to the simulation of high-shear batch granulation [11–16, 23, 30, 31]; silica nano-particle synthesis [34]; combustion engine modelling [17, 35]; soot formation [2, 18]; aerosol wet scavenging [44] and more general coagulation processes [43]. Under this framework, a representative sample volume of the system is modelled using an ensemble of computational particles. One such implementation of the stochastic particle method is Direct Simulation Monte Carlo, also known as the Direct Simulation Algorithm (DSA). In DSA, each computational particle represents a single physical particle in the ‘real’ system. Each computational particle has an associated particle vector $x$ that is used to characterise the physical particle it represents. In the case of granulation, this particle description may include the solid volume, various states of liquid volume and internal gas volume.

MC-PBM ensembles are evolved in time through Markov jump events that may act on/transform one or more particles at a time. These jump events represent the various particle mechanisms such as coagulation, breakage etc. These processes can be numerically challenging under DSA since highly unbalanced coagulation-breakage rates can lead to the undesired depletion or accumulation of computational particles [30]. Another disadvantage of the DSA method is that a large number of computational particles and repetitions of the simulation are often required to achieve an accurate sampling of the solution, both of which come at the cost of increased CPU time and memory [48]. The stochastic weighted particle method (also called the Stochastic Weighted Algorithm (SWA)), originally proposed by Rjasanow and Wagner [40], has been developed to overcome a number of the limitations of the DSA method. In SWA each computational particle has an associated statistical weight $w$. Using a number based weighting scheme under SWA, the number of ‘real’ particles represented by each computational particle $x$ is proportional to $w$, and the ensemble is represented as a collection of coordinates $(x, w)$. SWA analogues of various particle processes such as coagulation [22, 38, 49] and breakage [31, 47] have been developed. Using SWA these processes can be formulated such that they conserve the number of computational particles (through the transformations of weights). Examples of constant number SWA processes are the coagulation jump processes presented by Patterson et al. [37] and the breakage processes of Lee et al. [31]. Zhao and co-workers [45, 46, 49] have also presented a SWA scheme to dynamically control the number of computational particles that fall within predefined volume intervals. This is done through the periodic modification of the number of computational particles and statistical weights within each volume interval. This “shift” action acts to maintain statistical precision across the full range of the distribution.

In this paper, we are primarily concerned with the solution of the twin-screw granulation model presented in the first part of this study. Twin-screw granulation is a continuous method of wet granulation. The system consists of two co-rotating screws which convey solid material along the screw barrel (Figure 1). Liquid binder is injected at some point along the barrel length, which combines with the solid material and, under the action of the screw induced shear forces, leads to the formation of granules. For a comprehensive overview of twin-screw granulation as a process, the reader is referred to the first part of
In this paper, we present the numerical framework used to solve the four-dimensional MC-PBM for twin-screw granulation (presented in the first part of this study) using the stochastic weighted particle method. The stochastic algorithm presented in this paper includes nucleation, inception, coagulation and fragmentation jump processes. A novel nucleation jump process is constructed, which is capable of representing the multi-particle immersion nucleation processes in TSG systems. Further to this, we propose a new implementation of the SWA for the efficient initialisation/inception of particles from wide primary particle size distributions. A number of numerical studies are carried out to assess:

(i) the relative performance of the newly proposed inception scheme;

(ii) the optimal choice of selected numerical parameters within the new nucleation jump process;

(iii) the general convergence properties of the algorithm in the context of the twin-screw granulation test case.

To the best of the authors’ knowledge, this is the first time that a stochastic PBM has been applied to twin-screw granulation.
The remainder of the paper is structured as follows: in Section 2 we describe the particle type-space and the formulation of each stochastic jump process. The twin-screw granulation model is then briefly detailed in Section 3. In Section 4 we investigate potential algorithms for particle inception, the optimisation of key numerical parameters of the nucleation jump process and, finally, the convergence properties of the complete algorithm. The paper finishes with the presentation of the main findings of the work in Section 5.

2 Stochastic particle methods for granulation

2.1 Particle type-space

In this work, particles are elements of the general type-space $X$, which may be multi-dimensional and either discrete or continuous. Particles may take position in a bounded sequence of compartments $L$ which forms a linear chain. Particles may flow between compartments and inter-particle processes are only permitted between particles in the same compartment $z \in L$. In each compartment $z$, a set of stochastic particles is simulated in order to describe the population balance problem. The stochastic particles take the form

$$ (z_i, x_i, w_i), \quad i = 1, \ldots, N_T(t), $$

where $x \in X$, $w_i \in (0, w_{\text{max}}]$ is the statistical weight of the particle with index $i$ and $N_T(t)$ is the total number of particles across all $z$ at time $t$.

The statistical weights offer an additional level of freedom in the construction of the simulation algorithm, whilst maintaining the property

$$ \frac{1}{V_{\text{samp}} \sum_{i=1}^{N(z,t)} w_i \phi(z, x_j) \xrightarrow{V_{\text{samp}} \to \infty} \int_{X} \phi(z, x_j) P(t, z, dx)}. $$

Here, $V_{\text{samp}}$ is the normalisation parameter or sample volume associated with compartment $z$; $N(z, t)$ is the number of stochastic particles in compartment $z$; $\phi(z, x_j)$ is some suitable test function which is continuous and with compact support [37] and finally $P(t, z, dx)$ is a concentration measure that corresponds to the solution of the population balance problem to be solved. As mentioned in Section 1, (2) may be interpreted as saying that the physical concentration of particles represented by stochastic particle $x_i(t)$ is approximately $w_i/V_{\text{samp}}$. In the same way, the total particle concentration in compartment $z$ is $\sum w_i/V_{\text{samp}}$. As in Lee et al. [31], the sample volume is initialised such that

$$ \frac{1}{V_{\text{samp}} \sum_{i=1}^{N(z,0)} w_i} \approx \int_{X} P(0, z, dx). $$

The particle doubling/reduction method employed by Lee et al. [31] is used to control the number of computational particles in the system. Each compartment is initialised with $N(0) = 0.75N_{\text{max}}$ computational particles, where $N_{\text{max}}$ is the maximum number of particles permitted. For the purposes of error reduction, the minimum number of particles permitted is $(3/8)N_{\text{max}}$. 
The Markov jump process that evolves the system in time is defined by the possible jumps and their associated rates. Each available jump and the associated rate is a function of the state of the system. At each $t$ there exists a list of possible jumps which have independent, exponentially distributed waiting times. In this way, the earliest jump is selected and carried out, generating a modified system of particles with a new state and new rates. The solution then steps forward in time and the process is repeated until some stopping condition is met. The time step is described by the distribution:

$$P(t, t_{\text{wait}} \geq \theta) = \exp(-R_{\text{total}}^{\text{SWA}}(t)\theta), \quad \theta \geq 0,$$

where $R_{\text{total}}^{\text{SWA}}$ is the total jump rate.

### 2.2 Stochastic weighted algorithm

The possible SWA jump processes and associated rates used in this work are:

#### 2.2.1 Inception

The inception process is used to introduce purely solid particles into the system, representing the continuous addition of feed powder to the first simulated compartment of the twinscrew system. In this jump process, a new computational particle of the form

$$(z, x, w_{\text{incept}})$$

is added to compartment $z$ at rate

$$R_{\text{incept}}^{\text{SWA}}(z, t)f_{\text{incept}}(x)dx,$$

where $w_{\text{incept}}$ is the inception weight function (which may have a dependence on $x$), $f_{\text{incept}}(x)$ is the probability distribution of computational particles $(x, w_{\text{incept}})$ to be sampled by the inception jump and $R_{\text{incept}}^{\text{SWA}}(z)$ ([# events/unit time]) is the inception jump rate in compartment $z$. This is computed as

$$R_{\text{incept}}^{\text{SWA}}(z, t) = \frac{R_{\text{incept}}^{\text{SWA}}(z, t)V_{\text{samp}}(z, t)}{\bar{w}_{\text{incept}}}.$$  

Here, $R_{\text{incept}}(z, t)$ is the model inception rate ([# events/unit time/unit volume]) in compartment $z$ and $\bar{w}_{\text{incept}}$ is the number average statistical weight of an incepted particle, given as

$$\bar{w}_{\text{incept}} = \int_{X} w_{\text{incept}}f_{\text{incept}}(x)dx = \int_{X_{\text{incept}}} w_{\text{incept}}f_{\text{incept}}(x)dx.$$  

where $X_{\text{incept}} \subset X$ is the set of inception particles.

It is often the case that we wish to sample particles of the form (5) in such a way that it reproduces some physical particle distribution $q_{0,\text{incept}}(y)$, where $y$ is some scalar property of $x$. An simple example is that presented in the first part of this study, where
$y$ represents the diameter of solid, dry, non-porous primary particles. In such a case, $q_{0,\text{incept}}(y)$ is the number based diameter distribution for primary particles and the associated scalar function would simply map the solid volume of particles to the particle diameter as $y(x) = (s_\alpha(x)/k_v)^{1/3}$, where $k_v$ is some shape factor. In the general case, given some physical $q_{0,\text{incept}}(y)$, we would like to investigate how this distribution can be effectively represented in the computational particle domain with associated distribution, $q_{c,\text{incept}}(y)$ (by leveraging $w_{\text{incept}}(y)$).

To do so, we need to derive a relationship between the inception weight $w_{\text{incept}}(y)$, the physical distribution $q_{0,\text{incept}}(y)$ and the computational particle distribution $q_{c,\text{incept}}(y)$. Firstly, in order to move from $y \in \mathbb{R}^+$ to the space of inception particles $X_{\text{incept}}$, it is convenient to define some vector valued function $\Theta_{\text{incept}} : \mathbb{R}^+ \mapsto X_{\text{incept}}$. In the simple example described in the previous paragraph this would be characterised as $\Theta_{\text{incept}}(y) = (k_v y^3, 0, 0, 0)$.

In order to reproduce some physical distribution $q_{0,\text{incept}}(y)$ with a set of measures on $X_{\text{incept}}$, we must then satisfy the equality

$$\int_{X_{\text{incept}}} w_{\text{incept}}(y(x)) f_{\text{incept}}(x) dx = q_{0,\text{incept}}(y)$$

and from (8) we observe

$$\frac{w_{\text{incept}}(y) f_{\text{incept}}(\Theta_{\text{incept}}(y))}{\bar{w}_{\text{incept}}} = q_{0,\text{incept}}(y).$$

Let us define

$$q_{0,\text{incept}}(y) := y f_{\text{incept}}(\Theta_{\text{incept}}(y)),$$

then it follows from (10) that the required inception weighting function is

$$w_{\text{incept}}(y) = \frac{q_{0,\text{incept}}(y)}{q_{c,\text{incept}}(y)}.$$  

Since $\bar{w}_{\text{incept}}$ defines $w_{\text{incept}}(y)$, $\bar{w}_{\text{incept}}$ simply acts as a scaling constant on the inception weights and can therefore be freely selected to control the magnitude of $K_{\text{incept}}(z,t)$ by way of (7).

In this paper, we investigate the performance of the complete stochastic weighted algorithm for various forms of $q_{c,\text{incept}}(y)$ (and hence $w_{\text{incept}}(y)$) within the context of a twin-screw granulation model.

### 2.2.2 Nucleation

The nucleation process involves the addition of a droplet particle and the subsequent rapid addition of particles already present in the droplet zone to this droplet, forming a *nucleus*. Since the addition of particles to the droplet/partially formed nucleus is assumed to rapid relative to all other particle processes, the formation of the nucleus is carried out as an iterative process. The jump begins with the selection of a droplet particle that is to be added...
to compartment \( z \). This is the initial state of the partially formed nucleus \((x_{\text{nuc}}, w_{\text{nuc}})\). Particles within the ensemble are then selected for addition to this partially formed nucleus, analogous to a series of coagulation events between the partially formed nucleus and particles which are already located in compartment \( z \). The resulting nucleus particle is then added to compartment \( z \).

Since the addition of particles to the partially formed nucleus is assumed to be rapid, the jump rate for complete nucleation process (addition of multiple particles to the droplet and inception of the complete nucleus) is equal to the rate of droplet addition given as

\[
R_{\text{nuc}}^{\text{SWA}}(z, t) = \frac{R_{\text{drop}}(z, t) V_{\text{samp}}(z, t)}{w_{\text{nuc}}},
\]

(13)

where \( R_{\text{drop}}(z, t) \) is the model droplet addition rate. An important point to be taken from (13) is that the rate of nucleation jumps can be controlled through the selection of \( w_{\text{nuc}} \). This is important when sampling rare model events which have a high impact on the ensemble properties of interest. In this paper we will investigate how the choice of \( w_{\text{nuc}} \) affects the performance of the completed simulation algorithm and identify optimal operating regions for this parameter.

Following the approach taken by Kotalczyk and Kruis [27] to carry out coagulation processes, the addition of individual particles of the form \((x_i, w_i)\) to the partially formed nucleus \((x_{\text{nuc}}, w_{\text{nuc}})\) is carried out as

\[
(z, x_{\text{nuc}}, w_{\text{nuc}}), (z, x_i, w_i) \mapsto (z, T_{\text{nuc}}(x_{\text{nuc}}, x_i), \gamma_{\text{nuc}}(w_{\text{nuc}}, w_i)), (z, x_{\text{nuc}}, w_{\text{nuc}} - \gamma_{\text{nuc}}(w_{\text{nuc}}, w_i)), (z, x_i, w_i - \gamma_{\text{nuc}}(w_{\text{nuc}}, w_i)).
\]

(14)

In (14), \( T_{\text{nuc}} \) is the nucleation particle transform which represent the state of the partially formed nucleus after it has coalesced with particle \( x_i \). \( \gamma_{\text{nuc}} \) is the nucleation weight transfer function which, again, follows the approach of Kotalczyk and Kruis [27] such that

\[
\gamma_{\text{nuc}}(w_{\text{nuc}}, w_i) = \min(w_{\text{nuc}}, w_i).
\]

(15)

In order to achieve convergence, the jump process in (14) occurs with rate [27]

\[
K_{\text{nuc}}(x_{\text{nuc}}, x_i) \max(w_{\text{nuc}}, w_i).
\]

(16)

where \( K_{\text{nuc}} \) is the nucleation kernel. Since it is computationally expensive to compute the expression in (16) for all possible ensemble particles \((x_i, w_i)\), a majorant form of this kernel is used. This takes the form:

\[
\tilde{K}_{\text{nuc}}(x_{\text{nuc}}, w_{\text{nuc}}, x_i, w_i) = K_{\text{nuc}}(x_{\text{nuc}}, x_i)(w_{\text{nuc}} + w_i).
\]

(17)

The procedure used to pick particle \((x_i, w_i)\) to take part in the jump process (14) is discussed later in this paper, since a number of simplifications may be made after the introduction of the model nucleation kernel \( K_{\text{nuc}}(x, y) \). In order to ensure that the jump process (14) is performed at the correct rate, upon selection of the ensemble particle \((x_i, w_i)\) to be added to the partially formed nucleus, the jump is performed with probability

\[
\frac{\max(w_j, w_{\text{nuc}})}{w_i + w_{\text{nuc}}},
\]

(18)
otherwise the jump is fictitious and we move on to the selection of the next ensemble particle. For a more detailed account of the majorant kernel technique the reader is referred to Lee et al. [32].

As mentioned previously, the particle addition jump in (14) is carried out in an iterative manner on all partially formed nucleus i.e particle \((z, T_{\text{nuc}}(x_{\text{nuc}}, x_i), \gamma_{\text{nuc}}(w_{\text{nuc}}, w_i))\) is passed back through the jump process (14) with a new ensemble particle \(x_i\). This process is repeated until

\[ T(x_{\text{nuc}}, x_i) \in E, \quad (19) \]

where \(E \subset X\) is the set of particle vectors corresponding to completely formed nuclei. The definition of \(E\) in the context of the twin-screw model is introduced in Section 3. When (19) is satisfied, \(T(x_{\text{nuc}}, x_i)\) may no longer be passed through jump (14) and a new computational particle is incepted into compartment \(z\) with the form

\[ (z, T(x_{\text{nuc}}, x_i), w_{\text{nuc}}). \quad (20) \]

In the event that the jump (14) creates two partially formed nuclei\(^1\), then jump (14) must also be separately carried out on particle \((z, x_{\text{nuc}}, w_{\text{nuc}} - \min(w_{\text{nuc}}, w_i))\) until the stopping condition (19) is satisfied for this particle.

### 2.2.3 Coagulation/compaction

SWA coagulation jumps take different forms depending on whether or not the particles can be successfully coalesced. Additionally, this jump process includes a potential compaction transformation on the particles involved. For successful coalescence, the jump takes the form

\[ (z, x_i, w_i), (z, x_j, w_j) \mapsto (z, T_{\text{comp}}(x_i + x_j), \gamma_{\text{coag}}(x_i, w_i, x_j, w_j)), (z, x_j, w_j), \quad (21) \]

and for unsuccessful coalescence (i.e. a compaction/rebound):

\[ (z, x_i, w_i), (z, x_j, w_j) \mapsto (z, T_{\text{comp}}(x_i), w_i), (z, x_j, w_j). \quad (22) \]

In the above, \(\gamma_{\text{coag}}\) is the coagulation weight transfer function and \(T_{\text{comp}}\) is the compaction transformation. As in Lee et al. [31] we impose that

\[ \gamma_{\text{coag}}(x_i, w_i, x_j, w_j) = w_i \frac{m(x_i)}{m(x_i + x_j)}. \quad (23) \]

The total collision jump rate in compartment \(z\) is

\[ R_{\text{col}}^{\text{SWA}}(z, t) = \frac{1}{V_{\text{samp}}(z, t)} \sum_{i \neq j}^{N_{\text{col}}(z, t)} K_{\text{col}}(z, x_i, x_j)w_j. \quad (24) \]

\(^1\)Note that additional, partially formed nuclei particles can be formed by (14) if \(w_{\text{nuc}} > w_j\).
2.2.4 Breakage

In this paper, the breakage jumps take the form

\[(z, x_i, w_i) \mapsto (z, y, \gamma_{\text{frag}}(x_i, w_i, y)), \quad (25)\]

which occurs at rate

\[g_{\text{break}}(z, x). \quad (26)\]

Here, \(\gamma_{\text{frag}}\) is the breakage weight transfer function which takes the form [31]

\[\gamma_{\text{frag}}(x_i, w_i, y) = \frac{m(x_i)}{m(y)}. \quad (27)\]

As in [31, 32], \(y\) is selected as:

\[y = x_j, \quad \text{with probability } = \frac{m(x_j)}{m(x_i)}, \quad (28)\]

\[y = x_i - x_j, \quad \text{with probability } = 1 - \frac{m(x_j)}{m(x_i)}, \quad (29)\]

and \(x_j\) is selected according to the probability measure \(B_{\text{break}}(x_i)\).

The total breakage jump rate in compartment \(z\) is

\[R^{SWA}_{\text{break}}(z, t) = \sum_{i=1}^{N(z,t)} g_{\text{break}}(z, x_i). \quad (30)\]

2.2.5 Transport

Particles are permitted to move between compartments according to the jump

\[(z, x_i, w_i) \mapsto (z + 1, x_i, F_c(z)w_i), \quad (31)\]

at rate

\[\frac{1}{\tau(z)}, \quad (32)\]

Here, \(\tau(z)\) is the characteristic residence time of compartment \(z\) and \(F_c\) is the transport weight scaling factor, which takes the form [34]

\[F_c(z) = \frac{V_{\text{samp}}(z + 1)}{V_{\text{samp}}(z)} \quad (33)\]

for a series of compartments with equal real volume.

The total transport jump rate in compartment \(z\) is

\[R^{SWA}_{\text{outflow}}(z, t) = \frac{N(z,t)}{\tau(z)}. \quad (34)\]
2.2.6 Overall jump rate

The total jump rate of the stochastic weight algorithm $R_{SWA}^{total}$ is given by

$$R_{SWA}^{total}(t) = \sum_{z \in L} \left[ R_{SWA}^{col}(z,t) + R_{SWA}^{break}(z,t) + R_{SWA}^{nuc}(z,t) + R_{SWA}^{incept}(z,t) + R_{SWA}^{outflow}(z,t) \right].$$

(35)

Particle process $p \in \{\text{col, break, nuc, incept, outflow}\}$ in compartment $z$ is carried out at time $t$ with probability

$$\frac{R_{SWA}^p(z,t)}{R_{SWA}^{total}(t)}.$$  

(36)

2.2.7 Continuous processes

Intra-particle processes (such as liquid penetration) may be modelled as continuous processes. To minimise the computational cost of modelling such mechanisms, these are carried out using the linear process deferment algorithm (LPDA) [36]. LPDA has been successfully employed in the solution of stochastic population balance models for granulation [31], silica nanoparticle synthesis [34] and soot particle formation [36, 37]. In the LPDA, each stochastic particle is time stamped and the application of all linear processes is deferred until:

1. The next occurrence of a non-linear event (e.g. coagulation). At which point linear processes are applied to the particles involved in the jump.

2. The compartmental system has been simulated to/beyond the next deferment check point time (based on predefined list of deferment check points). At which point linear processes are applied to all stochastic particles, thereby updating each particle to the current simulation time. This occurs between stochastic jumps events.

3 Twin-screw granulation model

In this section we give a brief account of the twin-screw granulation model used in the numerical investigation. Only essential features are introduced for the propose of aiding readability. For an in-depth discussion of the model formulation the reader is directed to the first part of this study.

In the twin-screw model, particles $x \in X$ are described using a four-dimensional particle vector $x = (s_o, l_e, l_i, p)$ where: $s_o$ is the volume of original solid, $l_e$ is the volume of external liquid, $l_i$ is the volume of internal liquid and $p$ is the pore volume. The key derived particle properties are summarised in Table 1. The possible particle processes (see Figure 2) are: nucleation, particle collision/compaction (which may or may not lead
Table 1: Summary of derived particle properties. $\rho_s$ and $\rho_l$ are the solid and liquid densities, respectively.

<table>
<thead>
<tr>
<th>Property</th>
<th>Nomenclature</th>
<th>Expression</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume</td>
<td>$v(x)$</td>
<td>$s_o(x) + l_e(x) + p(x)$</td>
<td>m$^3$</td>
</tr>
<tr>
<td>Diameter</td>
<td>$d(x)$</td>
<td>$(6v(x)/\pi)^{1/3}$</td>
<td>m</td>
</tr>
<tr>
<td>Mass</td>
<td>$m(x)$</td>
<td>$\rho_s s_o(x) + \rho_l (l_e(x) + l_l(x))$</td>
<td>kg</td>
</tr>
<tr>
<td>Porosity</td>
<td>$\varepsilon(x)$</td>
<td>$p(x)/v(x)$</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 2: Particle processes in the TSG model presented in the first part of this study.

to coalescence), particle breakage, particle transport (between compartments) and liquid penetration (where external liquid becomes internal liquid). The individual models for each of these processes is presented in the next section.

3.1 Inception/initialisation

In this model, particles with diameter $d$ are incepted in the form of primary solid particles with the form

$$x_{\text{incept}}(d) = (s_{o,\text{incept}}(d),0,0,0),$$  \hfill (37)

which defines $X_{\text{nuc}}$. The volume of solid is then simply given by

$$s_{o,\text{incept}}(d) = \pi d^3/6.$$  \hfill (38)

These particles are incepted into compartment $z$ according to the inception model

$$R_{\text{incept}}(z,t) = \begin{cases} \frac{M_{\text{feed}}}{(\rho_s \bar{v}_{\text{incept}} V_{\text{real}}(z))} & \text{if } z = 1, \\ 0 & \text{otherwise}, \end{cases}$$  \hfill (39)
where $\dot{M}_{\text{feed}}$ is the mass feed rate to the real TSG system. $V_{\text{real}}(z)$ is the physical volume of compartment $z$ and $\bar{v}_{\text{incept}}$ is the arithmetic mean feed particle volume. This is given by

$$\bar{v}_{\text{incept}} = \frac{\int_{d_{\text{min}}}^{d_{\text{max}}} v(d) q_{0,\text{incept}}(d) dd}{\int_{d_{\text{min}}}^{d_{\text{max}}} q_{3}(d) d^{-3} dd}. \quad (40)$$

In this paper, we assume that all real compartments have equal volume. Thus, if we have $n_{\text{comp}}$ compartments and a total real system volume of $V_{\text{real},T}$ then

$$V_{\text{real}}(z) = \frac{V_{\text{real},T}}{n_{\text{comp}}}. \quad (41)$$

The feed distribution $q_{0,\text{incept}}$ used for inception and particle initialisation is taken from the volume fraction distribution presented in [25] for the Lactose Impalpable excipient grade. The diameter range is then split into 30 sections (spaced logarithmically). The discretised data is then converted into number distribution form $q_{0,\text{incept}}(d)$ using the relation [1]

$$q_{0,\text{incept}}(d) = \frac{q_{3}(d)d^{-3}}{\int_{0}^{\infty} q_{3}(d)d^{-3} dd}. \quad (42)$$

### 3.2 Nucleation

The twin-screw immersion nucleation model can be defined by the rate of droplet addition

$$R_{\text{drop}}(z,t) = \begin{cases} (\text{LSR}) \dot{M}_{\text{feed}} / v_{\text{drop}} \rho_{l} V_{\text{real}}(z), & \text{if } z = 1, \\ 0, & \text{otherwise}, \end{cases} \quad (43)$$

where LSR and $\rho_{l}$ are the liquid-solid mass feed ratio and liquid density, respectively.

The initial state of the partially formed nucleus particle $x_{\text{nuc}}$ is characterised by the droplet volume $v_{\text{drop}}$ as

$$x_{\text{nuc}}^{\text{start}} = (0, v_{\text{drop}}, 0, 0). \quad (44)$$

The nucleation kernel takes the form

$$K_{\text{nuc}}(x_{\text{nuc}}, x_{i}, z, t) = \begin{cases} R_{\text{nuc}}(z) \min(v(x_{\text{nuc}}), v(x_{i}))/\sum_{j=1}^{N(z,t)} v(x_{j}) & \text{if } \min(v(x_{\text{nuc}}), v(x_{i})) < v_{\text{drop}}, \\ 0 & \text{otherwise}, \end{cases} \quad (45)$$

and since, for the conditions we wish to model $v(x_{\text{nuc}}) > v(x_{i}) \forall i \in \{1, \ldots, N(z,t)\}$, this can be simplified to

$$K_{\text{nuc}}(x_{\text{nuc}}, x_{i}, z, t) = \begin{cases} R_{\text{nuc}}(z)v(x_{i})/\sum_{j=1}^{N(z,t)} v(x_{j}) & \text{if } v(x_{i}) < v_{\text{drop}}, \\ 0 & \text{otherwise}, \end{cases} \quad (46)$$
The set of valid, completely formed nuclei is defined as \( E = \{ x \mid l_e(x) = 0 \} \).

The nucleation type-space transformation \( T_{\text{nuc}} \) is characterised by the individual property transformations:

\[
\begin{align*}
    s_0(x_{\text{nuc}}) &\leftarrow s_0(x_{\text{nuc}}) + s_0(x_i) \\
    l_e(x_{\text{nuc}}) &\leftarrow l_e(x_{\text{nuc}}) \\
    l_i(x_{\text{nuc}}) &\leftarrow l_i(x_{\text{nuc}}) \\
    - \min \left( \left[ \phi_{\text{max}} - \phi(x_i) \right] 1_{\{x_i|\phi(x_i) \leq \phi_{\text{max}}\}}(x_i)s_0(x_i), l_e(x_{\text{nuc}}) \right), \\
    l_i(x_{\text{nuc}}) &\leftarrow l_i(x_{\text{nuc}}) + l_i(x_i) \\
    + \min \left( \left[ \phi_{\text{max}} - \phi(x_i) \right] 1_{\{x_i|\phi(x_i) \leq \phi_{\text{max}}\}}(x_i)s_0(x_i), l_e(x_{\text{nuc}}) \right), \\
    p(x_{\text{nuc}}) &\leftarrow l_i(x_i)/s^*,
\end{align*}
\]

where \( \phi(x) = (l_i + l_e)/(s_0) \) is the particle liquid saturation, \( \phi_{\text{max}} \) is the permitted particle liquid saturation level permitted for the material of interest and \( 1_{\{A\}}(x) \) is to be understood as the indicator function of set \( A \) where \( A \subset X, x \in X \).

### 3.3 Coagulation

Particle collisions are modelled using a size-independent collision kernel given as

\[
K_{\text{col}}(z, x_i, x_j) = n_{\text{screw}}k_{\text{col}}(z),
\]

where \( n_{\text{screw}} \) is the operating screw speed and \( k_{\text{col}} \) is the collision rate constant in compartment \( z \).

The Stokes criterion, as detailed in Braumann et al. [14], is used to determine whether or not a particular collision results in successful coalescence of the collision partners. Under this criterion, the probability of successful collision is dependent on the height of particle asperities \( H_a \) and particle coefficient of restitution \( e_{\text{coag}} \).

### 3.4 Compaction

Each collision event leads to the compaction of the particles involved which reduces their porosity and squeezes liquid to particle surface. The compaction transform \( T_{\text{comp}}(z, x) \) is characterised by the individual property transforms:

\[
\begin{align*}
    s_0 &\leftarrow s_0 \\
    l_e &\leftarrow l_e + (T_{\text{comp}}^p(z, x) - p)l_i/p, \\
    l_i &\leftarrow l_i - (T_{\text{comp}}^p(z, x) - p)l_i/p, \\
    p &\leftarrow T_{\text{comp}}^p(z, x),
\end{align*}
\]

where

\[
T_{\text{comp}}^p(z, x) = \frac{\varepsilon_1(z, x)(s_0 + l_e + l_i)}{\varepsilon_1(z, x) \left( 1 - \frac{l_i}{p} \right) - 1}
\]

16
and
\[
\varepsilon_i(z, x) = \begin{cases} 
k_{\text{comp}}(z)[\varepsilon(x) - \varepsilon_{\text{min}}] + \varepsilon(x), & \text{if } \varepsilon(x) \geq \varepsilon_{\text{min}}, \\ 
\varepsilon(x), & \text{otherwise.}
\end{cases}
\] (57)

Here \( k_{\text{comp}} \) is the compaction rate constant, \( \varepsilon(x) \) is the particle porosity (as defined in Table 1) and \( \varepsilon_{\text{min}} \) is the minimum porosity permitted.

### 3.5 Breakage

In the TSG model, individual particles undergo binary breakage at rate
\[
\gamma_{\text{break}}(z, x) = \begin{cases} 
k_{\text{att}}(z)n_{\text{screw}}^2v(x), & \text{if } v(x) \geq r_{\text{min}}^{\text{parent}} \text{ and } l_e(x) + l_i(x) + p(x) \neq 0, \\ 0, & \text{otherwise},
\end{cases}
\] (58)

where \( k_{\text{att}} \) is the attrition rate constant and \( r_{\text{min}}^{\text{parent}} \) is the minimum agglomerate size that can undergo breakage. As in Braumann et al. [14], the daughter distribution of each breakage event \( B_{\text{frag}}(x) \) is described by a beta function with skewness parameters \( \alpha_{\text{daughter}} \) and \( \beta_{\text{daughter}} \).

### 3.6 Compartmentalisation/transport

To reflect the variable processing environments along the length of the twin-screw, in both parts of this study the screw is modelled as three perfectly mixed compartments arranged in series (see Figure 1).

### 3.7 Penetration

The transfer of external liquid to internal liquid is modelled as a continuous process. The rate of liquid penetration for each computational particle is controlled by a penetration rate constant \( k_{\text{pen}} \) and liquid binder viscosity \( \mu_{\text{binder}} \) as
\[
r_{\text{pen}} = k_{\text{pen}}\mu_{\text{binder}}^{-1/2}l_e(p - l_i),
\] (59)

such that
\[
\frac{dx_o}{dt} = 0, \quad \frac{dl_e}{dt} = -r_{\text{pen}}, \quad \frac{dl_i}{dt} = r_{\text{pen}}, \quad \frac{dp}{dt} = -r_{\text{pen}}.
\] (60)

### 4 Numerical studies

In this section we investigate the influence of key numerical factors such as the choice of inception weighting scheme, the statistical weights/weight transfer function associated
with the nucleation process and the general convergence properties of complete stochastic weighting algorithm in the context of the twin-screw model. To ensure that the results of this analysis are relevant to typical model operating conditions (equipment operating conditions and rate constant values), all numerical tests are carried out using the equipment operating conditions and optimised model rate constants used/optimised in the first part of this study. These parameters are repeated in Table 2 and Table 3, respectively.

4.1 Simulation details

In this section we describe the simulation conditions used throughout the numerical studies. These parameters are to be assumed in all cases unless explicitly stated otherwise.

4.1.1 Simulation stop time and repetitions

Since the twin-screw granulation system is a continuous system, we are primarily interested in the steady state solution of the twin-screw PBM. As such, dynamic results are not assessed in this study. Visual analysis of the responses showed that the time at which simulation responses ceased to drift with \( t \) was controlled by the total system residence time. The final simulation time \( t_{\text{stop}} = 5 \sum_{z \in Z} \tau(z) \approx 40 \)s proved to be an acceptable choice and is used across all numerical studies in this paper.

Each simulation is carried out until \( t = t_{\text{stop}} \) and this process is repeated \( n_{\text{runs}} \) times, each time with a different seed to the pseudo-random number generator. In this study, each repetition will be referred to as a run.

4.1.2 Quantification of errors

For the stochastic simulations in this work, temporal functionals \( M(t) \) are reported as averages taken over all runs as

\[
\eta(t) = \frac{1}{n_{\text{runs}}} \sum_{i=1}^{n_{\text{runs}}} M_i(t),
\]

where the functional may be a particle ensemble property such as the mass fraction of particles in a particular sieve class.

The half-width of the confidence internals are

\[
c(t) = 1.64 \sqrt{\frac{\sum_{i=1}^{n_{\text{runs}}} (M_i(t) - \eta(t))^2}{n_{\text{runs}}^2}},
\]

which corresponds to a confidence interval \( P = 0.9 \) [31]. In cases where the statistical error is to be measured, the mean confidence interval \( \bar{c} \) of a measured distribution is used. This is computed as

\[
\bar{c} = \frac{1}{n_{\text{points}}} \sum_{j=1}^{n_{\text{points}}} c_j,
\]
where $n_{\text{points}}$ is the number of discrete points in the responses distribution of interest.

When a high precision solution (HPS) is used as a reference point, the sum of squared errors of prediction (SSE) is used to represent the systematic error and is given by:

$$\text{SSE} = \sum_{j=1}^{n_{\text{runs}}} (\eta_{\text{HPS},j} - \eta_j)^2.$$  \hspace{1cm} (64)

All HPS’s were run with $N_{\text{max}} = 65536$, $n_{\text{runs}} = 10$ and $w_{\text{nuc}} = 0.001$.

### 4.1.3 Binary tree

To ensure that the computational cost of solving the model is minimised, binary tree caches are employed to store key particle properties, thus enabling fast computation of ensemble wide properties (e.g. overall breakage rate) \cite{33, 36, 37, 41}.

### 4.1.4 Hardware set-up

All test simulations were are carried out using a single core of an Intel® Sandy Bridge® E5-2670 3.30GHz Processor with 4GB of RAM per core.

### 4.2 Inception sampling methods

In this section, we investigate various forms of the inception function $q_{0,\text{incept}}^{[c]}(d)$ and the resulting weight transfer functions $w_{\text{incept}}(d)$ (see (11) and (12)) in the context of the twin-screw granulation model presented in Section 3. We are reminded that $q_{0,\text{incept}}^{[c]}(d)$ represents the frequency at which stochastic particles of the form $(z, x_{\text{incept}}(d), w_{\text{incept}}(d))$ with diameter $d$ are selected within an inception jump, such that some physical inception particle size distribution $q_{0,\text{incept}}(d)$ is recovered.

In previous SW A studies \cite{31, 32}, the authors have employed

$$q_{0,\text{incept}}^{[c]}(d) = q_{0,\text{incept}}(d), \quad w_{\text{incept}} = 1,$$

$$\rightarrow w_{\text{incept}}(d) = 1,$$  \hspace{1cm} (65)

which we shall refer to as the equi-weighted inception scheme for SWA (EWI-SWA).

EWI-SWA is attractive due to its ease of implementation, however, in some cases, it has been observed that significant reduction in computational cost can be attained using more complex forms of $w_{\text{incept}}(d)$, depending upon the ensemble property of interest.

As an example, let us consider the case where we are interested in measuring the mass distribution (as a function of $d$) within the steady state ensemble of the twin-screw granulation simulation. It is often the case that the real density distribution of particles $q_{0,\text{incept}}(d)$ covers a $d$ range which is several orders of magnitude in size (such is the case of that presented for Lactose Impalpable in Hagrasy et al. \cite{25}). Given such a situation, we see from

\footnote{Note that (64) does not have any weighting factors since all $\eta$ of interest are of the same physical dimension and order of magnitude.}
Table 2: Summary of unoptimised model parameters used the first part of this study, used in numerical tests.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{\text{drop}}$</td>
<td>Operating parameter</td>
<td>$2 \times 10^{-3}$</td>
<td>m</td>
</tr>
<tr>
<td>$M_{\text{feed}}$</td>
<td>Operating parameter</td>
<td>4.0</td>
<td>kg hr$^{-1}$</td>
</tr>
<tr>
<td>$n_{\text{screw}}$</td>
<td>Operating parameter</td>
<td>6.67</td>
<td>rev s$^{-1}$</td>
</tr>
<tr>
<td>LSR</td>
<td>Operating parameter</td>
<td>0.35</td>
<td>-</td>
</tr>
<tr>
<td>$V_{\text{real,T}}$</td>
<td>Equipment geometry</td>
<td>$4.05 \times 10^{-5}$</td>
<td>m$^3$</td>
</tr>
<tr>
<td>$\epsilon_{\text{cogr}}$</td>
<td>Material property</td>
<td>0.2</td>
<td>-</td>
</tr>
<tr>
<td>$\rho_l$</td>
<td>Material property</td>
<td>998</td>
<td>kg m$^{-3}$</td>
</tr>
<tr>
<td>$\rho_s$</td>
<td>Material property</td>
<td>1545</td>
<td>kg m$^{-3}$</td>
</tr>
<tr>
<td>$\mu_{\text{binder}}$</td>
<td>Material property</td>
<td>$10^{-3}$</td>
<td>Pa s</td>
</tr>
<tr>
<td>$d_{\text{max}}$</td>
<td>Model parameter</td>
<td>$3.31 \times 10^{-6}$</td>
<td>m</td>
</tr>
<tr>
<td>$d_{\text{min}}$</td>
<td>Model parameter</td>
<td>$8.26 \times 10^{-4}$</td>
<td>m</td>
</tr>
<tr>
<td>$H_a$</td>
<td>Model parameter</td>
<td>$5 \times 10^{-6}$</td>
<td>m</td>
</tr>
<tr>
<td>$k_{\text{reac}}$</td>
<td>Model parameter</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>$v_{\text{min, parent}}$</td>
<td>Model parameter</td>
<td>$2.5 \times 10^{-5}$</td>
<td>m</td>
</tr>
<tr>
<td>$v_{\text{max, max}}$</td>
<td>Model parameter</td>
<td>$v_{\text{drop}}$</td>
<td>m$^3$</td>
</tr>
<tr>
<td>$V_{\text{compart}}$</td>
<td>Model parameter</td>
<td>$1.35 \times 10^{-5}$</td>
<td>m$^3$</td>
</tr>
<tr>
<td>$\alpha_{\text{daughter}}$</td>
<td>Model parameter</td>
<td>5.0</td>
<td>-</td>
</tr>
<tr>
<td>$\beta_{\text{daughter}}$</td>
<td>Model parameter</td>
<td>2.0</td>
<td>-</td>
</tr>
<tr>
<td>$\epsilon_{\text{min}}$</td>
<td>Model parameter</td>
<td>0.5</td>
<td>-</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Model parameter</td>
<td>2.76</td>
<td>s</td>
</tr>
<tr>
<td>$\phi_{\text{max}}$</td>
<td>Model parameter</td>
<td>1.08</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 3: Optimised model rate constants generated in first part of this study, used in numerical tests.

<table>
<thead>
<tr>
<th>Compartment index z</th>
<th>Parameter k_{col}</th>
<th>k_{att}</th>
<th>k_{comp}</th>
<th>Parameter k_{col}</th>
<th>k_{att}</th>
<th>k_{comp}</th>
<th>k_{pen}</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,3</td>
<td>m^3</td>
<td>m^{-1}s</td>
<td>-</td>
<td>1-3</td>
<td>m^3</td>
<td>m^{-1}s</td>
<td>kg^{-1}m^{-2}s^{-3}</td>
</tr>
<tr>
<td>Value</td>
<td>1.21×10^{-10}</td>
<td>9.42×10^6</td>
<td>0.395</td>
<td>9.99×10^{-13}</td>
<td>1.09×10^9</td>
<td>0.954</td>
<td>10.0</td>
</tr>
</tbody>
</table>

Figure 3: An illustrative example of the various sampling methods studied. In VWI-SWA, many small particles are grouped in a few computational particles, reducing the total number required to accurately sample ‘mass-rich’ regions of the distribution closer to d_{max}.

Figure 3(a) (blue trace) that the particles which occur with the greatest frequency are those with d much smaller than d_{max}. Consequently, most ‘real’ particles that are incepted into the physical system will have a volume (and thus mass) which is several orders of magnitude less than that held by particles with d close to the upper limit of the inception range.

Now, if one were to employ the EWI-SWA scheme (i.e. q_{0,\text{incept}}(d) = q_{0,\text{incept}}(d)) in such a situation, then it follows that most of the stochastic particles that are incepted will also carry very little volume (and hence very low mass), and thus have a very limited effect on the compartment mass distribution. However, occasionally stochastic particles with large d will be incepted into compartment 1, causing a temporary but significant change in the system mass distribution. This results in a high degree of stochastic noise within the measured product mass distribution. In order to mitigate this noise, it is required that the maximum number of computational particles N_{\text{max}} and/or n_{\text{run}} be increased, both of which increase the computational cost of simulation.

From the thought experiment above, we see that it would be advantageous to spread the
incepted mass over the incepted stochastic particles in a more uniform manner, while still recovering \( q_{0,\text{incept}} \). This is not only true of the inception process, but also the initialisation of the ensemble at \( t = 0 \). We further note that the transport processes also have the ability to induce a high degree of noise in the mass distribution when the compartment particle mass is poorly distributed across the ensemble of stochastic particles.

Methods to control the distribution of selected quantities, analogous to that described above, have been employed by Zhao and co-workers [45, 46, 49] and DeVille et al. [20] in the context of alternative particle models. In this paper we adapt the size dependent particle weighting approach of Zhao and Zheng [45] to the inception jump process (5) of the twin-screw model and also, to the initialisation of the ensemble (outlined in Section 2). Specifically, in this work, we investigate the case where \( q_{0,\text{incept}}^{[c]} \) is a uniform distribution over \( d \) with normalised form

\[
q_{0,\text{incept}}^{[c]}(d) = (d_{\text{max}} - d_{\text{min}})^{-1}. \tag{66}
\]

Here, \( d_{\text{max}} \) and \( d_{\text{min}} \) are the maximum and minimum particle diameters of the distribution \( q_{0,\text{incept}} \), respectively.

It follows from (12) that

\[
w_{\text{incept}}(d) = \bar{w}_{\text{incept}}(d_{\text{max}} - d_{\text{min}})q_{0,\text{incept}}(d). \tag{67}
\]

The use of (66) and (67) for inception will be referred to as the variable weighted inception scheme for SW A (VWI-SWA). The use of VWI-SWA results in a stochastic particle distribution analogous to that illustrated in Figure 3(b). Numerical tests carried out to assess the performance of the EWI-SWA and VWI-SWA inception algorithms described above. To ensure that the rates of each process were matched between compared sets of simulations at \( t = 0 \), the values of \( w_{\text{nuc}} \) was modified in the EWI-SWA case. For VWI-SWA \( \bar{w}_{\text{nuc}} \)

![Figure 4: A comparison of the product mass fraction distributions (z=3) and CPU times using the VWI-SWA and EWI-SWA algorithms with \( N_{\text{max}}=8192 \). A high precision solution using VWI-SWA is included but sits directly behind the blue trace.](image)

22
was set to 16.38 in all simulations.

The product ensembles of each simulation are sieved using a sieve set starting from 32\(\mu\)m to 8064\(\mu\)m with a \(\sqrt{2}\) geometric progression. Sieve mass fractions are plotted against the mid-point of the corresponding sieve intervals.

The resulting sieved exit mass fraction distributions using both EWI-SWA and VWI-SWA with \(N_{\text{max}} = 8192\) are presented in Figure 4. It is noted that, for the EWI-SWA scheme, most of the computational feed particles have been utilised to form the lower end of the distribution between 10-100\(\mu\)m. As a result, sampling of the primary particle distribution is much poorer for larger particle diameters (where most of the ensemble mass resides). The error in these larger sieve classes for the EWI-SWA scheme has carried over into the steady state solution. By contrast, the VWI-SWA scheme has sampled the initial distribution with much higher precision in the larger sieve classes, resulting in a steady state distribution with relatively small confidence intervals across the complete diameter range. Since both simulations operated with the same bounds on the number computational particles permitted, it is expected that both EWI and VWI-SWA schemes should result in CPU times which are of the same order of magnitude. This is confirmed by the inset CPU plot in Figure 4, where the weighted inception algorithm has showed a similar but slightly reduced CPU time over the EWI-SWA scheme. The variations in CPU time likely reflect the interaction between the particle selection procedure of the nucleation algorithm and the ensemble weight distribution. This interaction may change the rate at which particles are fully depleted (deleted from the ensemble) by the mechanism, altering the equilibrium computational particle count and, hence, the CPU time.

![Figure 5](image-url)

**Figure 5:** A comparison of the total CPU time vs. associated inverse average confidence interval half-width in the sieved exit mass fraction distribution for 40s of simulation with \(n_{\text{runs}} = 10\). The number of computational particles (attached green blocks) is varied to yield different CPU times and 1/\(\bar{c}\) values.

The scaling performance of each inception/initialisation algorithm was assessed by run-
ning a series of simulations with varying $N_{\text{max}}$ (and therefore the minimum number of computational particles, as $N_{\text{min}} = 3N_{\text{max}}/8$). Simulations were run with $N_{\text{max}}=1024$ and increased by a factor of two to a final $N_{\text{max}}=65536$. Simulations using the EWI-SWA scheme proved to be numerically unstable for $N_{\text{max}}=[1024, 2048]$, and thus results for these operating points are not reported here. The resulting set of mean confidence interval half-widths $\bar{c}$ across the resulting sieve exit mass fraction distribution and their associated CPU times are presented in Figure 5. Here, $1/\bar{c}$ is used to quantify the precision of each response.

It is clear from Figure 5 that the VWI-SWA scheme offers a much higher precision solution than the equivalent EWI-SWA (same CPU time). Specifically, the VWI-SWA scheme can be seen to yield solutions in CPU times almost two orders of magnitude lower than the EWI-SWA for the same level of precision.

### 4.3 Immersion nucleation parameters

In this section, we investigate the selection of the key parameter $w_{\text{nuc}}$, the selection of ensemble particles $x_i$ and simplification of the iteration process within the nucleation jump process (14) for the twin-screw granulation test case.

#### 4.3.1 Selection of the nucleation weight

We are reminded that the statistical weight of incepted nuclei $w_{\text{nuc}}$ is a free parameter within the context of the nucleation jump process. Furthermore, we observe from (13) and (43) that the rate of nucleation jumps in compartment $z = 1$ has the characteristic

$$R^{\text{SWA}}_{\text{nuc}}(1, t) \propto (\bar{v}_{\text{drop}}w_{\text{nuc}})^{-1}. \quad (68)$$

In previous SWA efforts in the modelling of high shear batch granulation [30, 32], alternative droplet particle inception jump processes with jump rates given by equations of the form (68) have been implemented with the equivalent of $w_{\text{nuc}} = 1$. In these examples the value of $\bar{v}_{\text{drop}}$ was very small (on the order of $10^{-13}$ m$^3$) to reflect the operation of the equipment being modelled. As a result, the rate of this process was significant (relative to the total jump rate) allowing it to be sampled with a high degree of accuracy.

However, in the case of twin-screw granulation, operating values of $\bar{v}_{\text{drop}}$ can be much larger (on the order of $10^{-8}$ m$^3$). Hence, if $w_{\text{nuc}} = 1$, then the nucleation jump rate can be exceptionally low relative to the total jump rate, resulting in poor sampling of this jump process. In addition to this, it is noted that a single nucleation jump can have a significant impact on the particle mass distribution within the ensemble (particularly when $w_{\text{nuc}}$ is on the order of $\bar{v}_{\text{incept}}$), and hence poor nucleation sampling results in a high degree of stochastic noise in the product mass distribution. This ultimately places a limit on the size of $\bar{v}_{\text{drop}}$ that can be modelled with an acceptable degree of error (with $w_{\text{nuc}}=1$). Again, like the case of poor solid particle inception sampling, this noise can be mitigated by increasing $N_{\text{max}}$ and $n_{\text{runs}}$ at an additional computational cost. In this study, we avoid the need to increase $N_{\text{max}}$ and/or $n_{\text{runs}}$ by dynamically selecting $w_{\text{nuc}}$ according to the operating conditions to be modelled. It will be demonstrated that, in doing so, we can ensure the
effective sampling of this important jump process within the context of the twin-screw granulation model.

### 4.3.2 Simplification of the jump

As mentioned in Section 2.2.2, the nucleation particle addition jump (14) has the drawback of potentially creating more than a single partially formed nucleus in the jump products (when \( w_{\text{nuc}} < w_i \)). This additional, partially formed nuclei must also then be separately and repeatedly passed through jump (14) until the condition (19) is satisfied. This is undesirable as these additional nuclei have the potential to create yet more nuclei, so on and so forth, before (19) is satisfied. Hence it is possible that this process may become computationally taxing and flood the ensemble with particles which are physically very similar, and possesses very low statistical weights. To avoid these issues we note that, in the context of the twin-screw model, ensemble particles \( x_i \) that are able to attached to the partially formed (i.e. those for which \( K_{\text{nuc}}(x_{\text{nuc}}, x_i) \neq 0 \) are generally much smaller in volume than the corresponding nucleus \( x_{\text{nuc}} \). Hence, the addition of a single \( x_i \) to \( x_{\text{nuc}} \) has relatively little effect on the state of \( x_{\text{nuc}} \) i.e.

\[
T_{\text{nuc}}(x_{\text{nuc}}, x_i) \approx x_{\text{nuc}}. \tag{69}
\]

Furthermore, we note that the statistical weight of the first partially formed nucleus (in the case where two nuclei are formed) is \( \min(w_{\text{nuc}}, w_i) = w_i \), which is generally much greater than the weight of the secondary nucleus \( (w_{\text{nuc}} - w_i) \). As such, the product particles may be combined into a single representative particle using a weighted average of the particles of \( (T_{\text{nuc}}(x_{\text{nuc}}, x_i), w_i), (x_{\text{nuc}}, \min(w_{\text{nuc}} - w_i)) \) with the form:

\[
(w_{\text{nuc}}^{-1} w_i T_{\text{nuc}}(x_{\text{nuc}}, x_i) + (w_{\text{nuc}} - w_i)x_{\text{nuc}}), w_{\text{nuc}}. \tag{70}
\]

It was observed that, for the systems modelled in this paper, this simplification had no discernible effect on the model solution but offered modest reduction in simulation times (5-10%) and code complexity, therefore this simplification is employed in all simulations within this study. An illustrative example of the immersion nucleation jump process if presented in Figure 6.

### 4.3.3 Selection of ensemble particles

As alluded to in previous sections, the selection of computational particles \((x_i, w_i)\) for addition to partially formed nuclei by way of jump (14) can be simplified based on the model nucleation kernel \( K_{\text{nuc}} \) employed. Combining (17) and (46) we see that the majorant form of the nucleation kernel for the twin-screw model is

\[
\tilde{K}_{\text{nuc}}(x_{\text{nuc}}, w_{\text{nuc}}, x_i, w_i, z, t) = R_{\text{drop}}(z, t) \frac{v(x_i)(w_{\text{nuc}} + w_i)1_{\{v(x_i) < v_{\text{drop}}\}}(x_i)}{\sum_{j=1}^{N(z,t)} v(x_j)} \frac{1_{\{v(x_i) < v_{\text{drop}}\}}(x_i)}{\sum_{j=1}^{N(z,t)} v(x_j)}. \tag{71}
\]
Since only a single nuclei particle is considered at a time, the total rate of this process is

\[
\sum_{i=1}^{N(z,t)} \tilde{K}_{nuc}(x_{nuc}, w_{nuc}, x_i, w_i, z, t) = \sum_{i=1}^{N(z,t)} R_{\text{drop}}(z, t) \frac{v(x_i)(w_{\text{nuc}} + w_i) \mathbb{I}_{\{v(x_i) < v_{\text{drop}}\}}(x_i)}{\sum_{j=1}^{N(z,t)} v(x_j)} 
\]

\[
= \frac{R_{\text{drop}}(z, t) w_{\text{nuc}}}{\sum_{j=1}^{N(z,t)} v(x_j)} \sum_{i=1}^{N(z,t)} v(x_i) \mathbb{I}_{\{v(x_i) < v_{\text{drop}}\}}(x_i) + R_{\text{drop}}(z, t) \sum_{i=1}^{N(z,t)} v(x_i) w_i \mathbb{I}_{\{v(x_i) < v_{\text{drop}}\}}(x_i). 
\]

Thus, particle \((x_i, w_i)\) is selected to take part in the jump (14) based on the selection property \(v(x_i)\) with probability

\[
P_{v(x_i)} = \frac{w_{\text{nuc}} \sum_{i=1}^{N(z,t)} v(x_i) \mathbb{I}_{\{v(x_i) < v_{\text{drop}}\}}(x_i)}{w_{\text{nuc}} \sum_{i=1}^{N(z,t)} v(x_i) \mathbb{I}_{\{v(x_i) < v_{\text{drop}}\}}(x_i) + \sum_{i=1}^{N(z,t)} v(x_i) w_i \mathbb{I}_{\{v(x_i) < v_{\text{drop}}\}}(x_i)}. 
\]

It follows that particle \((x_i, w_i)\) is selected to take part in the jump (14) based on the selection property \(v(x_i)w_i\) with probability

\[
P_{v(x_i)w_i} = 1 - P_{v(x_i)}. 
\]

Using a binary tree to store properties \(v(x_i) \mathbb{I}_{\{v(x_i) < v_{\text{drop}}\}}(x_i)\) and \(v(x_i)w_i \mathbb{I}_{\{v(x_i) < v_{\text{drop}}\}}(x_i)\) for each computational particle allows for rapid evaluation of the the summations in (75) and hence the evaluation of \(P_{v(x_i)}\) and the selection of particle \(x_i\) with the desired probability measure.

The complete nucleation algorithm (including the particle selection procedure described above) is provided in Appendix A.

4.3.4 Nucleation performance

To assess the effect of \(w_{\text{nuc}}\) on the performance of twin-screw simulations, several simulations were carried out using the VWI-SWA algorithm and \(N_{\text{max}} = 1024\) for various values of \(w_{\text{nuc}}\) in the range \(5 \times 10^{-5} - 5 \times 10^{-2}\). All errors were constructed from the sieved mass fraction distribution in the final reactor \((z = 3)\).

From Figure 7(a) we note that the mean confidence interval half-width \(\bar{c}\) varies relatively smoothly with \(w_{\text{nuc}}\) and \(\bar{c}\) exhibits an increase towards the limits of the \(w_{\text{nuc}}\) range investigated. The picture painted by the SSE in the same figure is less clear towards low \(w_{\text{nuc}}\), however, from PSD’s in Figure 7(b) we see that all simulation solutions lie in close proximity to the HPS, regardless of \(w_{\text{nuc}}\) and thus the SSE values are subject to a high degree of noise in this operating range. The increase in \(\bar{c}\) and SSE towards the upper ranges of the tested \(w_{\text{nuc}}\) is indicative of the poor sampling of the nucleation jump, as discussed in Section 4.3.1. Towards the lower range of \(w_{\text{nuc}}\), the nucleation jump rate is very high and the absolute liquid mass carried by the stochastic nuclei particles at the start of the jump is relatively low. Since the amount of solid particles that interact with the nuclei is roughly proportional to the absolute liquid mass of the stochastic nuclei particle, the
Figure 6: An illustration of the SWA immersion nucleation scheme. Each particle represents a single computational particle. The angle of the unshaded region of each particle is constructed such that it is proportional to the associated statistical weight of the particle. Each pathway corresponds, respectively, to the first, intermediate (with particle removal) and final iteration of the particle selection loop of Algorithm 1 (steps 3-12).

number of solid particles acted on by each nuclei jump becomes much lower. Hence we expect greater fluctuations in the primary particle mode at low $w_{\text{nuc}}$, as observed in the left-most peak in the PSD of Figure 7(b).

As expected, in Figure 7(c) we see that, in general, the CPU time is reduced with increasing $w_{\text{nuc}}$ as the total jump rate in reduced. This trend diminishes towards large $w_{\text{nuc}}$ as nucleation ceases to be the dominant jump process. For $w_{\text{nuc}} > 0.007$, coagulation becomes the dominant jump process (in terms of CPU-time) and so the computation time becomes independent of $w_{\text{nuc}}$. In Figure 7(d), the combination of error and CPU time data for various values of $w_{\text{nuc}}$ shows that there exists a favourable range of $w_{\text{nuc}}$ approximately between $10^{-2} - 5 \times 10^{-4}$ where both the error and computation time are minimised.
Figure 7: The effect of varying $w_{\text{nuc}}$ on simulation performance. All results correspond to the steady state sieved mass fraction distribution in $z = 3$. In (d) the value of $w_{\text{nuc}}$ corresponding to each point is indicated within the attached blocks.
4.4 Convergence properties

It is useful to consider the convergence properties of the complete stochastic algorithm for the twin-screw test case. This is done by assessing how the systematic and statistical error in the final mass distribution depends on $N_{\text{max}}$ and $n_{\text{runs}}$. All convergence tests are carried out with the VWI-SWA algorithm and $w_{\text{nuc}} = 0.001$.

![Graphs showing convergence properties](image)

**Figure 8:** Convergence properties of the twin-screw SWA framework using VWI-SWA.

In Figure 8(a) and 8(b) we show the dependence of the final mass distribution and its associated error on $N_{\text{max}}$ for an approximately fixed computational budget by holding $N_{\text{max}} \times n_{\text{runs}}$ constant at 32768. In this region of operating space, we see that there is no correlation between $N_{\text{max}}$ and the statistical or systematic error measures and that all
measured mass distributions lie on top of the HPS. Both of these observations indicate that the simulation algorithm is extremely robust against changes in $N_{\text{max}}$, provided that the $n_{\text{runs}}$ is scaled appropriately. Lower values of $N_{\text{max}}$ could not be explored as the algorithm became numerically unstable for $N_{\text{max}} < 128$. These instabilities are induced by the fact that the incepted nuclei particles can no longer gather sufficient solid mass to complete the nucleation jump process.

Thus, in order to access regions of operating space where the systematic and statistical error is more significant, it is required that lower values of $N_{\text{max}} \times n_{\text{runs}}$ are used. In Figure 8(c) we show the variation in the final mass distribution and the associated confidence intervals with fixed $N_{\text{max}}=128$ for varying values of $n_{\text{runs}}$. This figure shows that $N_{\text{max}} \times n_{\text{runs}}$ can be lowered to 8192 before the mass distribution begins to significantly deviate from the HPS. It is also clear that at even with $n_{\text{runs}}=4$ the solution still lies in close proximity to the HPS, though the confidence intervals have dramatically increased in size, as expected.

This region of operating space is further explored in Figure 8(d), which shows how the statistical and systematic error varies with $n_{\text{runs}}$ for low values of $N_{\text{max}}$. Here, if we compare simulations of equal $N_{\text{max}} \times n_{\text{runs}}$ (i.e. comparable CPU times) then we observe that, in general, there is no clear dependence between $N_{\text{max}}$ and statistical or systematic error for low values of $N_{\text{max}}$. From this analysis we may conclude that, for a twin-screw test case investigated, if a desired level of error in the final mass distribution is desired (either systematic or statistical), then there is no significant benefit of increasing $N_{\text{max}}$ provided that $N_{\text{max}}$ exceeds the value required for computational stability (128 in this case).
5 Conclusions

In this paper, we have presented a stochastic weighted particle framework that can be used to solve the four-dimensional twin-screw granulation PBM introduced and optimised in the first part of this study. A new nucleation jump process is outlined, which is compatible with an immersion nucleation model in the context of wet granulation. A variable weighting particle inception/initialisation algorithm was constructed to aid the efficient sampling of wide primary particle size distributions. This inception algorithm was shown to deliver performance increases of between one and two orders of magnitude over the traditional equi-weighted inception scheme. The numerical properties of the new nucleation jump process were explored and it was observed that there exists a specific range of nuclei statistical weight that minimises both the error in the final mass fraction distribution and the computational cost of simulation. The convergence properties of the complete algorithm were assessed using the twin-screw test case and it was shown that the algorithm is extremely robust against changes in the number of computational particles used.

Acknowledgements

The authors would like to thank AstraZeneca for funding this work. This project was partly funded by the National Research Foundation (NRF), Prime Minister’s Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.
Abbreviations

ANN Artificial neural network
DSA Direct simulation algorithm
DEM Discrete element method
EWI-SWA Equally weighted inception for the stochastic weighted algorithm
HPS High precision solution
LPDA Linear process deferment algorithm
PBM Population balance model
MC-PBM Monte Carlo population balance model
SWA Stochastic weighted algorithm
TSG Twin-screw granulation
VWI-SWA Variable weighting inception for the stochastic weighted algorithm

Nomenclature

Roman symbols

$B_{\text{break}}$ breakage daughter distribution probability measure
$c$ confidence interval half-width
$\bar{c}$ average confidence interval half-width (of a distribution)
$d$ particle diameter
$d_{\text{drop}}$ droplet diameter
$d_{\text{max}}$ maximum primary particle diameter
$d_{\text{min}}$ minimum primary particle diameter
$E$ set of valid nuclei particles
$f_{\text{incept}}$ inception stochastic particle probability distribution
$F_c$ transport weight scaling factor
$g_{\text{break}}$ particle breakage rate
$H_a$ height of surface asperities
$k_{\text{att}}$ breakage rate constant
$k_{\text{col}}$ collision rate constant
$k_{\text{comp}}$ compaction rate constant
$k_{\text{pen}}$ penetration rate constant
$k_{\text{reac}}$ number of compartments
$K_{\text{col}}$ size independent collision kernel
$k_v$ shape factor
$K_{\text{nuc}}$ nucleation kernel
$\bar{K}_{\text{nuc}}$ majorant nucleation kernel
$l_e$ external liquid volume
$l_e,\text{nuc}$ nuclei external liquid volume
$l_i$ internal liquid volume
L \hspace{10pt} \text{the set of compartment indices} \\
LSR \hspace{10pt} \text{operating liquid solid mass flowrate ratio} \\
m \hspace{10pt} \text{particle mass} \\
M \hspace{10pt} \text{model functional} \\
M_{\text{feed}} \hspace{10pt} \text{solid mass flowrate} \\
n_{\text{runs}} \hspace{10pt} \text{number of simulation repetitions} \\
n_{\text{screw}} \hspace{10pt} \text{screw speed} \\
N \hspace{10pt} \text{number of stochastic particles} \\
n_{\text{comp}} \hspace{10pt} \text{number of compartments} \\
N_{\text{max}} \hspace{10pt} \text{maximum number of stochastic particles/compartment} \\
N_{\text{min}} \hspace{10pt} \text{maximum number of stochastic particles/compartment} \\
N_{T} \hspace{10pt} \text{total number of stochastic particles in all compartments} \\
p \hspace{10pt} \text{pore volume} \\
P \hspace{10pt} \text{density of the population distribution} \\
q_{0,\text{incept}} \hspace{10pt} \text{number based physical particle size distribution for inception} \\
q_{0,\text{incept}}^{\text{c}} \hspace{10pt} \text{number based computational particle size distribution for inception} \\
q_{3} \hspace{10pt} \text{number based particle size distribution} \\
r_{\text{pen}} \hspace{10pt} \text{particle penetration rate} \\
R_{\text{incept}} \hspace{10pt} \text{model particle inception rate} \\
R_{\text{SWA break}} \hspace{10pt} \text{SWA breakage jump rate} \\
R_{\text{SWA col}} \hspace{10pt} \text{SWA collision jump rate} \\
R_{\text{SWA incept}} \hspace{10pt} \text{SWA particle inception jump rate} \\
R_{\text{nuc}} \hspace{10pt} \text{model nucleation rate} \\
R_{\text{SWA nuc}} \hspace{10pt} \text{SWA nucleation jump rate} \\
R_{\text{outflow}} \hspace{10pt} \text{model particle outflow rate} \\
R_{\text{SWA outflow}} \hspace{10pt} \text{SWA particle outflow jump rate} \\
R_{\text{SWA total}} \hspace{10pt} \text{SWA total jump rate} \\
S_{0} \hspace{10pt} \text{original solid volume} \\
S_{0,\text{incept}} \hspace{10pt} \text{inception original solid volume function} \\
SSE \hspace{10pt} \text{sum of squared errors of prediction} \\
t \hspace{10pt} \text{time} \\
t_{\text{wait}} \hspace{10pt} \text{jump waiting time} \\
T_{\text{comp}} \hspace{10pt} \text{compaction type transformation} \\
T_{\text{comp ext}} \hspace{10pt} \text{compaction external liquid transformation} \\
T_{\text{comp int}} \hspace{10pt} \text{compaction internal liquid transformation} \\
T_{\text{comp pv}} \hspace{10pt} \text{compaction pore volume transformation} \\
T_{\text{nuc}} \hspace{10pt} \text{nucleation type transformation} \\
v \hspace{10pt} \text{particle volume} \\
v_{\text{feed}} \hspace{10pt} \text{number average feed particle volume} \\
v_{\text{min}} \hspace{10pt} \text{minimum volume for breakage} \\
v_{\text{drop}} \hspace{10pt} \text{droplet volume} \\
\nu_{\text{max}} \hspace{10pt} \text{maximum particle volume involved in nucleation} \\
m \hspace{10pt} \text{kg} \\
m s^{-1} \\
\text{rev s}^{-1} \\
m^{3} \\
m^{-3} \\
m^{-1} \\
m^{-1} \\
m^{3} \\
m^{3} \\
m^{3} \\
m^{-3} \\
m^{-1} \\
m^{-1} \\
m^{-1} \\
m^{-1} \\
m^{-1} \\
m^{3} \\
m^{3} \\
s \\
s \\
s \\
m^{3} \\
m^{3} \\
m^{3} \\
m^{3} \\
m^{3} \\
m^{3} \\
m^{3} \\
m^{3} \\
m^{3} \\
m^{3}
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\bar{V}_{\text{drop}})</td>
<td>number averaged droplet volume</td>
<td>(\text{m}^3)</td>
</tr>
<tr>
<td>(V_{\text{real}})</td>
<td>physical compartment volume</td>
<td>(\text{m}^3)</td>
</tr>
<tr>
<td>(V_{\text{real,T}})</td>
<td>total physical volume of all compartments combined</td>
<td>(\text{m}^3)</td>
</tr>
<tr>
<td>(V_{\text{samp}})</td>
<td>compartment sample volume</td>
<td>(\text{m}^3)</td>
</tr>
<tr>
<td>(w)</td>
<td>particle statistical weight</td>
<td>-</td>
</tr>
<tr>
<td>(w_{\text{incept}})</td>
<td>mean inception particle statistical weight</td>
<td>-</td>
</tr>
<tr>
<td>(w_{\text{feed}})</td>
<td>feed particle statistical weight</td>
<td>-</td>
</tr>
<tr>
<td>(w_{\text{nuc}})</td>
<td>nuclei statistical weight</td>
<td>-</td>
</tr>
<tr>
<td>(x)</td>
<td>particle vector</td>
<td>-</td>
</tr>
<tr>
<td>(x_{\text{nuc}})</td>
<td>nuclei particle vector</td>
<td>-</td>
</tr>
<tr>
<td>(x_{\text{nuc}}^{\text{start}})</td>
<td>initial nuclei particle vector</td>
<td>-</td>
</tr>
<tr>
<td>(X)</td>
<td>particle type-space</td>
<td>-</td>
</tr>
<tr>
<td>(X_{\text{nuc}})</td>
<td>nuclei particle type-space</td>
<td>-</td>
</tr>
<tr>
<td>(z)</td>
<td>compartment index</td>
<td>-</td>
</tr>
</tbody>
</table>

**Greek symbols**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha_{\text{daughter}})</td>
<td>breakage distribution parameter</td>
<td>-</td>
</tr>
<tr>
<td>(\beta_{\text{daughter}})</td>
<td>breakage distribution parameter</td>
<td>-</td>
</tr>
<tr>
<td>(\gamma_{\text{coag}})</td>
<td>coagulation weight transfer function</td>
<td>-</td>
</tr>
<tr>
<td>(\gamma_{\text{frag}})</td>
<td>breakage weight transfer function</td>
<td>-</td>
</tr>
<tr>
<td>(\gamma_{\text{nuc}})</td>
<td>nucleation weight transfer function</td>
<td>-</td>
</tr>
<tr>
<td>(\varepsilon)</td>
<td>particle porosity</td>
<td>-</td>
</tr>
<tr>
<td>(\varepsilon_1)</td>
<td>post-compaction particle porosity</td>
<td>-</td>
</tr>
<tr>
<td>(\varepsilon_{\text{min}})</td>
<td>minimum particle porosity</td>
<td>-</td>
</tr>
<tr>
<td>(\eta)</td>
<td>run averaged model measure</td>
<td>-</td>
</tr>
<tr>
<td>(\Theta_{\text{incept}})</td>
<td>inception particle function</td>
<td>-</td>
</tr>
<tr>
<td>(\mu_{\text{binder}})</td>
<td>binder viscosity</td>
<td>(\text{Pa s})</td>
</tr>
<tr>
<td>(\rho_{\text{b}})</td>
<td>binder density</td>
<td>(\text{kg m}^{-3})</td>
</tr>
<tr>
<td>(\rho_{\text{s}})</td>
<td>solid density</td>
<td>(\text{kg m}^{-3})</td>
</tr>
<tr>
<td>(\phi)</td>
<td>liquid saturation</td>
<td>-</td>
</tr>
<tr>
<td>(\phi_{\text{max}})</td>
<td>maximum liquid saturation</td>
<td>-</td>
</tr>
<tr>
<td>(\tau)</td>
<td>compartment residence time</td>
<td>(\text{s})</td>
</tr>
</tbody>
</table>
A Appendix: SWA immersion nucleation algorithm

Algorithm 1: The SWA immersion nucleation algorithm.

1 Set the droplet volume $v_{\text{drop}}$ and nucleus particle vector $x_{\text{nuc}}$

$$v_{\text{drop}} \leftarrow \pi d_{\text{drop}}^3 / 6, \quad x_{\text{nuc}} \leftarrow (0, v_{\text{drop}}, 0, 0)$$

2 while $I_e(x_{\text{nuc}}) > 0$

3 Choose particle selection procedure according to (75) and (76).
4 Choose particle $x_i$ from compartment $z$ according to selection procedure.
5 Generate a uniform random number $U(0, 1)$.

6 if $U < \max(w_i, w_{\text{nuc}}) / (w_i + w_{\text{nuc}})$ then

7 Set $\gamma_{\text{nuc}} \leftarrow \min(w_i, w_{\text{nuc}})$.
8 Create particle $(z, T_{\text{nuc}}(x_{\text{nuc}}, x_i), \gamma_{\text{nuc}})$.
9 Set $w_i \leftarrow w_i - \gamma_{\text{nuc}}$.
10 Set $w_{\text{nuc}} \leftarrow w_{\text{nuc}} - \gamma_{\text{nuc}}$.
11 if $w_i = 0$ then

12 Remove particle $x_i$ from the ensemble

13 if $w_{\text{nuc}} > 0$ then

14 Product nuclei are to be combined

$$x_{\text{nuc}} \leftarrow (w_{\text{nuc}}^{-1}(w_i T_{\text{nuc}}(x_{\text{nuc}}, x_i) + (w_{\text{nuc}} - w_i) x_{\text{nuc}}, w_{\text{nuc}})$$

15 $w_{\text{nuc}} \leftarrow w_{\text{nuc}} + \gamma_{\text{nuc}}$

else

Jump is fictitious. Go to Step 3

16 Add newly formed nucleus $(x_{\text{nuc}}, w_{\text{nuc}})$ into compartment $z$.
17 Move forward in time and select next jump event.
References


