Particle agglomeration in flows: fast data-driven spatial decomposition algorithm for CFD simulations

Kerlyns Martínez Rodríguez, Mireille Bossy, Christophe Henry

Particle agglomeration in flows: fast D2SD algorithm for CFD simulations

Kerlyns Martínez Rodríguez[a,b,*], Mireille Bossy[a], Christophe Henry[a]

[a] *Université Côte d'Azur, Inria, CaliSto laboratory, Sophia-Antipolis, France*
[b] *Center for Mathematical Modeling, University of Chile.*

Abstract

Computational fluid dynamics simulations in practical industrial/environmental cases often involve non-homogeneous concentrations of particles. In turns, this induces the propagation of numerical error when using population balance equation (PBE)-like algorithms to compute agglomeration inside each cell of an Eulerian mesh. In this article, we apply the data-driven spatial decomposition (D2SD) algorithm to control such error in simulations of particle agglomeration. Significant improvements are made to design a fast D2SD version, minimizing the additional computational cost by developing re-meshing criteria. Through the application to some practical simulation cases, we show the importance of splitting the domain when calculating agglomerations, so that within each elementary cell there is a spatially uniform distribution of particles.

Keywords: Agglomeration, Particle-laden flows, Population Balance Equation (PBE), Mesh-independent, Particle-based mesh, Computational Fluid Dynamic (CFD)

1. Introduction

1.1. General context: particle agglomeration

Particle agglomeration is the process whereby solid particles dispersed in a flow adhere together to form larger structures (referred to as aggregates [1, 2]). The agglomeration process is at play in a range of fields belonging to both natural and applied sciences. For instance, in natural sciences, agglomeration occurs in geo-morphology [3] (e.g. delta river formation), meteorology [4, 5, 6] (e.g. droplet growth in clouds) or astrophysics [7] (e.g. planetoids growth). In applied sciences, agglomeration plays a major role in a number of industrial applications including in waste-water treatment facilities [8, 9], in combustion systems [10, 11], in oil/sludge refining [12] and food industry [13].

From this brief overview of application, it appears that the process of agglomeration involves a variety of objects (molecules, polymers, solids, droplets, bubbles, etc.) and covers a wide range of temporal and spatial scales (from molecular scales with proteins up to astrophysical scales with planetoids). Depending on the field, this process is also referred to as aggregation (as for solid materials), flocculation (e.g. polymers), coalescence (such as droplets/bubbles) or coagulation (e.g. non-Newtonian fluids).

In this paper, we focus on the prediction of agglomeration of solid particles in the colloidal range (i.e. with a size ranging from a few nanometers up to a few micrometers) that are suspended in a flow.

*Corresponding author
Email addresses: kerlyns.martinez-rodriguez@inria.fr (Kerlyns Martínez Rodríguez), mireille.bossy@inria.fr (Mireille Bossy), christophe.henry@inria.fr (Christophe Henry)

1.2. Modelling of particle agglomeration: existing approaches and limitations

Given the importance of the agglomeration process in many natural and industrial applications, accurate predictions are paramount. In particular, different models dedicated to particle-laden flows have been proposed in the literature (see, e.g. [14] for more details). The main approaches, from microscopic to macroscopic, are summarised below (see also [15]):

a. *N*-particle tracking (or individual particle tracking) with collision detection. These approaches are based on a microscopic level of description. They usually consist in a coupling of three approaches: (i) a fine-scale simulation of the fluid phase (e.g. DNS); (ii) an explicit Lagrangian tracking of the motion of a large number of particles (e.g. explicit solver for translational and rotational equations of motion) and (iii) a direct collision-detecting algorithm (e.g. with an overlap criterion [16] or based on a geometric criterion [17]). Such approaches allow to obtain directly the number of agglomeration events as a result of the simulation, providing detailed information on aggregate properties (e.g. shape of agglomerates, spatial/temporal correlations between collisions) [18]. However, they suffer from their high computational costs and are thus used only in idealised or specific small-scale cases.
b. One-particle (Lagrangian) tracking with given collision frequencies. These methods have a "coarser" level of description. They usually consist in a coupling of three approaches: (i) an Eulerian simulation of a flow (e.g. using RANS models when dealing with turbulent flows); (ii) a one-particle pdf Lagrangian model for the motion of particles [19, 20, 21, 22] (based on stochastic equations) and (iii) a detection model for particle collisions (e.g. a proba-

bilistic collision algorithms [23, 24] or a mesh-based algorithm [25, 26, 27]). These approaches are thus based on a kind of Bird-like algorithm [28, 29, 30], which rely on a separate treatment of the transport step (where the particles do not interact with each other but with a mean field obtained from the whole set of particles) and of the collision step (interaction without transport).
It should be noted here that, in order to avoid the high computational costs associated with the tracking of all real particles, the notion of parcel is often used: one "numerical" parcel is a statistical representation of a large number of real particles. In that case, the collision-detection algorithm is adapted for parcels. Similar formulations have been used recently replacing the probability of collisions between pairs of particles/parcels with the discretised Population Balance Equations [31].

c. Population Balance Equations (PBE). PBE are macroscopic approaches that allow to perform fast evaluations of the kinetics of particle agglomeration at large-scales. They describe the evolution in time of mean-field quantities. More precisely, these equations describe the rate of variation of the number density of a population of particles due to agglomeration within a control volume $\mathbb{V}_C$. As particles agglomerate, the distribution of sizes changes according to the following integro-partial-differential equation [32, 33]:

$$\frac{\partial n(\mathbb{v},t)}{\partial t} = \frac{1}{2}\int_0^{\mathbb{v}} \alpha_{\mathbb{v}-\mathbb{w},\mathbb{w}}\,\beta_{\mathbb{v}-\mathbb{w},\mathbb{w}}\,n(\mathbb{v}-\mathbb{w},t)n(\mathbb{w},t)d\mathbb{w} - \int_0^{\infty} \alpha_{\mathbb{v},\mathbb{w}}\,\beta_{\mathbb{v},\mathbb{w}}\,n(\mathbb{v})n(\mathbb{w})d\mathbb{w}, \tag{1}$$

where $n(\mathbb{v},t)$ is the number density of particles of volume $\mathbb{v}$ at time t within the control volume $\mathbb{V}_C$, α is the collision efficiency and $\beta_{\mathbb{v},\mathbb{w}}$ the collision frequency kernel between particles of volume $\mathbb{v}$ and $\mathbb{w}$. The first term on the r.h.s. of Eq. (1) corresponds to the formation of new aggregates of volume $\mathbb{v}$ produced by the collision between pairs of smaller aggregates (such that the sum of their volumes equals $\mathbb{v}$) while the second term on the r.h.s. accounts for the loss of particles of volume $\mathbb{v}$ due to their agglomeration with other particles.
PBE approaches are compatible with the level of description used in standard Computational Fluid Dynamics (CFD) simulations [34, 33] (e.g. Euler-Euler simulations with turbulence models). However, they suffer from several limitations that have been well identified in the literature [33, 35].

In the present paper, we focus on hybrid approaches which combine Eulerian simulations of the fluid phase, Lagrangian tracking of particles and PBE-inspired treatment of agglomeration (see for instance [31]). The use of a mesh-based PBE-inspired algorithm for the detection of particle collisions induces limitations that are related to well-known PBE drawbacks, including:

a. Only binary collisions are considered.
b. There is no correlation between individual particles.
c. The velocity field acting on particles is constant in each cell of the mesh.
d. Information on the collision efficiency α and on the collision frequency kernel β are needed as an input.
e. The population of particles is assumed uniformly distributed in each volume control $\mathbb{V}_C$ considered (e.g. cells in a mesh).

In this paper, we focus on the uniform distribution assumption, referred later as the spatially-uniform condition. It means that no information on the particle coordinates is required in the collision step [33]. Yet, in practice, this spatially-uniform condition must be fulfilled locally in each volume element (i.e. computational cell in CFD simulations). However, this assumption does not necessarily hold within complex flows, where particles can be injected locally in the system or accumulate in specific regions, thus leading to significant gradients of particle concentration at the system scale [26]. As an immediate consequence, numerical errors are quickly propagated in the simulation [36, 15], which result in a need for prior mesh-dependency analysis dedicated to the collision step. In this context, spatial-decomposition algorithms can prove useful in any Euler-Lagrange approach that may rely on two independent meshes (one for the fluid phase and another one for the particle phase).

A variety of attempts have been made to reduce the mesh-dependency of the agglomeration results [37, 38, 39, 40, 41]. Initially, the proposed methods focused on computing the agglomeration over a volume comprising the cell (within the mesh used for the CFD simulations) and its surrounding neighbours. Other proposed methods generate a specific mesh for the computation of the particle agglomeration (e.g. the NTC algorithm) by randomly choosing the orientation, every time step, before splitting the domain, or by introducing an adaptive collision mesh method. The main drawback of these approaches lies in the fact that they do not rely on any information coming from the spatial distribution of the particles/parcels. More recently, a new data-driven spatial decomposition (D2SD) algorithm has been proposed to detect non-homogeneous concentrations in a set of particles (point data) within a regular volume [15]. This D2SD algorithm also provides a spatial splitting according to the spatial repartition of particles. One of the many advantages of the D2SD algorithm is that the parameters are automatically tuned through the statistical information coming from the data (position of particles). Thus, there is no bias coming from arbitrary parameters. The algorithm is coupled with a battery of uniformity tests to check whether the input and output data satisfy the spatially-uniform condition. Then, the output data, along with the domain, are classified in regions according to their density of presence.

1.3. Objectives of the paper

The present article is a follow-up of the previous paper that described the D2SD algorithm [15]. The previous paper was focussed on assessing the robustness and accuracy of the algorithm to detect regions of non-homogeneous concentration. Furthermore, the importance of applying the method when computing particle agglomeration in complex 3D industrial/environmental cases has been illustrated in a simple case, and the application of the D2SD algorithm to CFD simulations was left out.

The objective of this paper is thus to adapt the D2SD algorithm respecting requirements for standard CFD codes and to use the algorithm in a practical 3D case. More precisely, this article proposes numerical schemes for an efficient use of D2SD, decreasing the computational cost of D2SD or/and the number of times the algorithm is called during the simulation. Several options are assessed, introducing a criterion to avoid applying the full version of the D2SD algorithm every time step, or simplifying uniformity tests. The main difficulty is to ensure that the adapted algorithm has an appropriate balance between its accuracy and its computational costs.

1.4. Layout of the paper

To address the above objectives, the paper is organised as follows. First, the full D2SD algorithm is briefly recalled in Section 2.1. Faster versions of the D2SD algorithm to fit the computational efficiency and costs of CFD simulations are detailed in Sections 2.2. This fast D2SD algorithm is analysed in terms of accuracy and computational efficiency on a series of numerical experiments that are presented in Section 3. Finally, numerical simulations are performed in a practical 3D case. The results are analysed in Section 4, with a specific emphasis on the accuracy of the results obtained.

2. Adaptation of the D2SD algorithm to CFD simulations

2.1. Summary of the D2SD algorithm

For the sake of clarity and self-sustainability, the D2SD algorithm [15] is briefly recalled in the following. It basically generates a dynamic mesh accounting for the spatial repartition of particles while respecting the spatially-uniform condition (required by PBE-like models). Thus, without the need of user-parameters, the algorithm detects deviations from uniformity in the spatial distribution of the particles within the normalised domain $[0,1]^d$, d being the dimension, and generates a domain partition considering this information.

More precisely, the D2SD algorithm only requires information on the positions of a set of point particles $\mathbf{X} = \{X^1, X^2, \dots, X^n\}$ in a space domain $\mathcal{D}$ generically reduced to $[0,1]^d$ in this paper. Then, the method comprises the following three steps (all of them depicted in Figure 1):

`step 1.` Statistical uniformity test.

The first step consists of checking whether the input sample $\mathbf{X}$ corresponds to the case of uniformly-distributed particles. For that purpose, several statistical uniformity tests (considered from the classical literature of goodness of fit) are applied. More precisely, we apply three different discrepancy tests (symmetric, centred and star discrepancy tests) together with a Henze-Zirkler normality test applied to the sample $\left\{\Phi^{-1}(X^1), \Phi^{-1}(X^2), \dots, \Phi^{-1}(X^n)\right\}$, with Φ the cumulative distribution function of a standard normal random variable. Additionally, a Pearson test is applied in order to check the independence of the sample (more details can be found in [15]). Then, considering the results of the different tests, a majority voting criterion is implemented, accepting the spatially-uniform condition if -at least- three tests accept the uniformity.

`step 2.` Spatial decomposition. The 2nd step splits iteratively to search for an optimal spatial decomposition that respects the spatial-uniform condition. It is applied only if the spatially-uniform condition is rejected in `step 1`. In general terms, the idea is to approximate the probability density function (pdf) associated with the set of particle positions and to find a suitable pdf-threshold (through an optimisation step) that separates the clusters from the ambient uniform particles. This optimal decomposition is obtained with the following sub-steps:

`step 2.a` Compute the bin size for the empirical pdf.

The size of the bin is computed with the Freedman and Diaconis rule [42] using information from the data. This allows to accurately approximate the pdf of particle concentration. Precisely, the bin size for the jth-coordinate is given by:

$$h_j(\mathbf{X}) = 2\frac{\texttt{iqr}(\mathbf{X}_j)}{n^{3/2}}, \tag{2}$$

with $\texttt{iqr}(\cdot)$ denoting the interquartile range of a random sample.

`step 2.b` Compute the empirical pdf.

The approximate density (i.e. concentration) is computed for each bin defined in `step 2.a`, with:

$$\texttt{pdf}(i) = \frac{\sum_{j=1}^{n} \mathbf{1}_{X^j \in \texttt{Bin}(i)}}{n \prod_{j=1}^{d} h_j(\mathbf{X})},$$

for all $i \in \{1, 2, \dots, \#\text{bins}\}$, and each $\texttt{Bin}(i) \subset [0,1]^d$ having dimensions $h_1(\mathbf{X}) \times h_2(\mathbf{X}) \times \dots \times h_d(\mathbf{X})$.

`step 2.c` Compute the ideal score.

An ideal score is computed quantifying how far the data is from a uniform sample. For that purpose, we resort to the score function defined by the distance of $\mathbf{X}$ to the boundary of the domain $\mathcal{D} = [0,1]^d$ [43]:

$$d_b(\mathbf{X}, \mathcal{U}) := \int_0^1 |G_{n,\mathbf{Z}}(z) - H_{\partial\mathcal{D},\mathcal{U}}(z)|\, dz, \tag{3}$$

where $G_{n,\mathbf{Z}}$ stands for the empirical cumulative distribution function of the relative depth sample $\mathbf{Z} = \{Z^1, Z^2, \dots, Z^n\} \in \mathbb{R}^n$, with $Z^i = 2d(X^i, \partial\mathcal{D})$, and $H_{\partial\mathcal{D},\mathcal{U}}(z) = 1-(1-z)^d$, for $0 \le z \le 1$. Here, $d(X^i, \partial\mathcal{D})$ denotes the Eulerian distance between the point X^i in $\mathcal{D}$ and the boundary of the domain $\partial\mathcal{D}$; the integral in Equation (3) can be approximated with a trapezoid method.

The ideal score $S_{\texttt{ideal}}$ is then obtained by averaging the score associated with the targeted uniformly distributed sample $\mathbf{U}$ of size n i.e.

$$S_{\texttt{ideal}} = \langle d_b(\mathbf{U}, \mathcal{U}) \rangle,$$

where $\langle\cdot\rangle$ stands for the expectation under the law of the sample (computed with a Monte Carlo approximation).

`step 2.d` Detect concentration anomalies w.r.t. uniformity and merge bins according to concentration level.

For that purpose, we introduce a threshold λ, which allows to separate the pdf (and therefore the domain) into high-concentration values and uniform ambient pdf values. To avoid the use of arbitrary parameters, percentile indicators are used as a measure of dispersion. In this context, to find an optimal value of this threshold λ, the 1d-distribution of the pdf-values (called here reduced-pdf) together with its percentiles are computed. Then, an optimal threshold $\lambda^* \in [Q_{60}, Q_{80}]$ will be chosen from an iterative procedure starting from $\lambda_0 = Q_{60}$.

The high-concentration regions (corresponding to the interval $[\lambda^*, \max_i \texttt{pdf}(i)]$) are further divided into five different sub-levels of particle concentration using the re-normalised percentiles (i.e. related to the reduced-pdf values): $\widetilde{Q}_{20}$, $\widetilde{Q}_{40}$, $\widetilde{Q}_{60}$, $\widetilde{Q}_{80}$. Similarly, the void bins and bins with low concentration (corresponding to those bins having a pdf-value below Q_{20}) are identified. The set of bins that have not been classified at this point will be considered as ambient uniform regions.

As a result, a set of eight binary d-dimensional arrays of size $\otimes_{1\leq i\leq d}$`Nb_bins(i)` is obtained, where `Nb_bins(i)` denotes the number of bins in the ith-dimension. In order to reduce the statistical noise in the computation of agglomeration events within cells having a small amount of particles, the final domain decomposition is constructed by merging bin-cells identified with the same level of particle.

`step 2.e` Create a synthetic sample.

A synthetic sample is introduced to score the decomposition obtained at a given value of the threshold. For that purpose, particles within the bin-cells identified as low- or high-particle concentration regions are replaced by synthetic uniformly distributed particles, forming (together with the particles within the ambient regions) the synthetic sample $\mathbf{X}^\lambda$. The number of synthetic particles inside a cell is determined using the cell volume matching the ambient concentration.

`step 2.f` Compute the score of the synthetic sample and compare.

Considering the decomposition obtained from `step 2.d` and the synthetic sample $\mathbf{X}^\lambda$ constructed in `step 2.e`, we compute the associated error (with respect to the ideal score computed in `step 2.c`):

$$\mathbf{E}(\mathbf{X}^\lambda) = \frac{|S_{\texttt{ideal}} - \langle S(\lambda)\rangle_{\text{synthetic}}|}{S_{\texttt{ideal}}},$$

with $S(\lambda) = d_b(\mathbf{X}^\lambda, \mathcal{U})$ defined in (3), and the bracket $\langle S(\lambda)\rangle_{\text{synthetic}}$ denotes the average of the score over several samples of the synthetic observations.

`step 2.d`, `2.e`, `2.f` are applied iteratively, starting from $\lambda = Q_{60}$ till $\lambda = Q_{80}$ to find the optimal threshold λ^* (i.e. with a score closest to the ideal score):

$$\lambda^* = \texttt{argmin}_\lambda \mathbf{E}(\mathbf{X}^\lambda). \tag{4}$$

`step 3.` Posteriori uniformity check.

The final step of the algorithm consists of checking whether the synthetic sample obtained for the optimal threshold $\mathbf{X}^{\lambda^*}$ corresponds to a uniform sample. To that extent, uniformity tests (as in `step 1`) are applied. If the uniformity is rejected (i.e. majority of rejection), we apply iteratively `step 2` and `step 3` until the a-posteriori uniformity test is accepted. This means that the D2SD algorithm is applied once more on the synthetic sample until we get a uniform sample $\mathbf{X}^{\lambda^*}$.

Once the required steps of the D2SD algorithm have been completed, the output is a spatial decomposition distinguishing between areas of zero concentration, low concentration, medium concentration and high concentration. As required by PBE-based approaches, each bin-cell defined with the D2SD algorithm respects the local spatially-uniform condition.

2.2. Fast version of the D2SD for CFD purposes

As mentioned in the introduction, some existing CFD software relies on Lagrangian methods for particle tracking coupled to PBE-inspired algorithms for particle agglomeration. Yet, the use of a PBE-like model requires a uniform particle distribution in each volume considered (i.e. the control volume in PBE corresponds to a single cell in the mesh and not to the whole domain $\mathcal{D}$). For that reason, the D2SD algorithm appears as a suitable candidate to create a new mesh that meets this condition. This particle mesh is thus only related to the position of particles in the whole domain $\mathcal{D}$ and is independent from the mesh used for the fluid simulation. It is only used during the agglomeration step and has no influence on particle transport (which, if necessary, can be still carried out using the mesh used for the fluid simulation). The D2SD algorithm has been shown to properly and accurately capture the presence of low, mild or high non-uniformities in the particle concentration (either void regions or over-concentration [15]). The accuracy of the algorithm has been characterised in a range of 2D and 3D numerical experiments that are representative of realistic situations. When using the D2SD algorithm with PBE approaches for particle agglomeration, it has also been shown that better results are obtained compared to an arbitrary choice of mesh.

However, these tests have shown that the numerical costs associated with the use of the D2SD algorithm can dramatically increase when a large number of particles are involved. This is mostly related to the numerical costs associated with the uniformity verification steps (`step 1` and `step 3`) and the iteration of `steps 2.d`, `2.e`, `2.f`. To take all the benefice of using the D2SD algorithm in CFD software, it is necessary to reduce the numerical cost associated with this spatial decomposition since it is expected to be applied at every time step (i.e. using the current information on particle positions). The solely but key

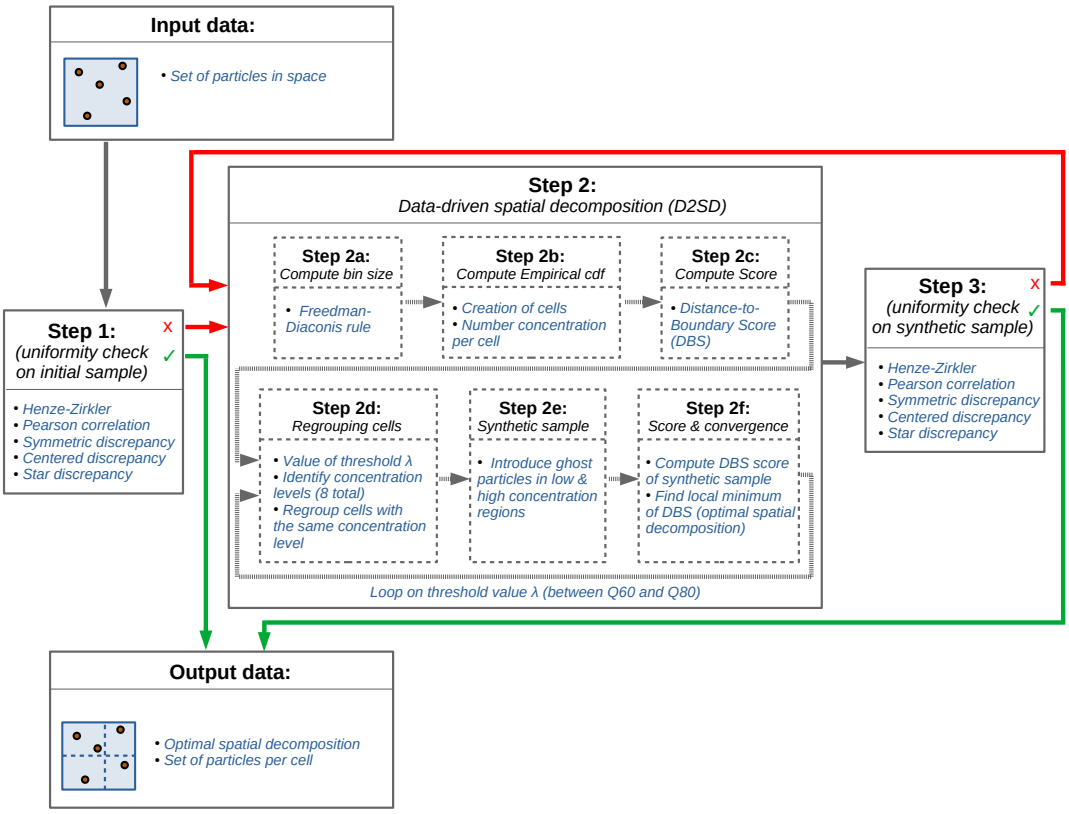


Figure 1: Sketch of the D2SD algorithm. Figure taken from [15].

issue when designing a fast D2SD algorithm is to avoid losing too much accuracy. For that reason, in the rest of the paper, we will often refer to efficiency measured as the ratio between the computational costs and the accuracy of the algorithm proposed.

Drawing on these computational issues, the aim of this section is to present some decision/selection criteria that offer a reduced version of the D2SD algorithm. In particular, we assess here three levels of simplification that are summarised in the following:

`Algo 1` Full D2SD with re-meshing criterion.

The idea would be to update the spatial decomposition only when deemed necessary, using the full version of the D2SD algorithm (i.e. `step 2` in Section 2.1) together with uniformity tests. This implies first to develop a criterion to decide when D2SD needs to be applied again, i.e. when the spatial decomposition with particle positions $\mathbf{X}_t$ at time t becomes too different from the spatial decomposition applied to the sample $\mathbf{X}_{t-s}$ at the time distance s. The criterion is composed of two tests (see Section 2.2.2 below), one based on the fluctuation of the displacement of the particles within the domain $\mathcal{D}$ and the other based on the variation of the particle concentration.

`Algo 2` Quick D2SD, no re-meshing criterion.

This option consists of applying only the steps corresponding to the spatial decomposition algorithm using only a priori information, speeding up the process. This means that the uniformity tests are limited to sample size smaller than n_{crit} and that the iterative search of an optimal threshold (`steps 2.d`, `2.e`, `2.f`) is simplified. More precisely, the idea is to choose a-priori a value $\hat{\lambda}$ in place of the optimal value λ^*. With this, the D2SD algorithm reduces to implementing the `steps 2.a`, `2.b`, `2.d`, together with `step 1` when $n < n_{crit}$. The choice for $\hat{\lambda}$ is discussed in Section 2.2.1.

`Algo 3` Quick D2SD with re-meshing criterion.

This version combines the two previous options, i.e. a re-meshing criteria is used to avoid applying the spatial decomposition algorithm every time step while using a simplified decomposition with an a-priori evaluation of the pdf-threshold $\hat{\lambda}$.

These three levels of simplification to speed-up the computation of the spatial decomposition rely on two main notions: the quick D2SD algorithm and the re-meshing criteria that we detail in the following.

2.2.1. *Quick D2SD: a faster computation of spatial decomposition*

As described before, the quick D2SD algorithm requires an "as good as possible" guess value $\hat{\lambda}$ instead of the optimal one λ^* defined in (4). Notice that, within the procedures considered up

to `step 2.c`, the reduced-pdf (required to classify the regions by levels of concentration) is already computed. Thus, the idea is to use the statistical information provided by the pdf in order to propose an a priori threshold. In this context, since we know that the optimal threshold is searched within the set $[Q_{60}, Q_{80}] \cap \cup_i \texttt{pdf}(i)$, starting from $\lambda_0 = Q_{60}$, we choose here

$$\hat{\lambda} := Q_{60}, \tag{5}$$

Indeed, the algorithm will continue to classify 5 different concentration levels in $[\hat{\lambda}, Q_{80}]$. By underestimating λ^* with $\hat{\lambda}$, the decomposition will induce a higher numerical error, but at least possibly distributed on the 5 levels instead of just one. This choice will be assessed later in Section 3.

2.2.2. Re-meshing criterion

In computations of particle agglomeration using Euler-Lagrange approaches, variations of the particle concentration can be linked to three possible sources: (i) the motion of particles can lead to their departure from the current cell, (ii) the agglomeration/fragmentation of particles can induce concentration variations and (iii) the presence of particle sources/sinks can directly change the number of particles in a cell. Since the last two possibilities can be related to each other, we propose here a re-meshing criterion based on two separate criteria: a first one on particle dispersion within $\mathcal{D}$ and a second one on the variation in the particle concentration due to sources/sinks/agglomeration events.

In both cases, we consider the input pair $(\mathbf{X}_k, \mathtt{M}_k)$, with $\mathbf{X}_k$ the set of particles position and $\mathtt{M}_k$ its corresponding spatial decomposition at a given iteration k of the simulation. Before formalising the re-meshing criterion, we describe below the principle of the parts that compose it.

- Criterion on particle displacement fluctuation

  A usual criterion related to displacement in CFD codes is the CFL condition (to ensure the numerical convergence of the solver). However, a CFL condition ensuring that, on average, the particles are not moving by more than one bin-cell during a time step is not relevant here. The key notion is the change in the particle concentration due to the fluctuation in particle displacement. More precisely, the idea is to analyse the variance of particle scattering between time steps. We resort here to the bin size computed in `step 2.a` of the D2SD algorithm since it is directly related to the information about how the particles are dispersed in space (here the interquartile range, see Equation (2)). The fluctuations in the particle scattering between time steps are evaluated by comparing directly the respective bin sizes.

  Denote $\texttt{N\_bins}(\mathbf{X}_t)$ as the number of bins estimated for the set of positions $\mathbf{X}_t$. The idea then is to quantify the percentage of lost/win bins between time steps and to use this value as an indicator of the fluctuations in particle displacement. More precisely, we compute

$$\Delta_t \texttt{N\_bins} = \frac{(\texttt{N\_bins}(\mathbf{X}_{t+\Delta t}) - \texttt{N\_bins}(\mathbf{X}_t))}{\texttt{N\_bins}(\mathbf{X}_t)}\%. \tag{6}$$

  Positive values of $\Delta_t \texttt{N\_bins}$ will represent the percentage of bins won between the mesh at time $t + \Delta t$ and the mesh at time t. Similarly, negative values of $\Delta_t \texttt{N\_bins}$ will represent the percentage of lost bins. Clearly, if $\Delta_t \texttt{N\_bins} = 0$, there is no change in the bin size.

  Hence, the criterion is to apply the D2SD algorithm at $t+\Delta t$ only if $\|\Delta_t \texttt{N\_bins}\| > \epsilon_1\%$, where ϵ_1 is a given tolerance threshold. Otherwise, if $\|\Delta_t \texttt{N\_bins}\| \leq \epsilon_1\%$, the spatial decomposition is not changed (i.e. the mesh is the same).

- Criterion on concentration variation

  The second indicator measures how much particle concentration changes between two time steps due to sources/sinks/agglomeration events.

  Denote $\mathcal{L}$ the set of 8 different levels of concentration and $\mathcal{D}_\ell$ the region associated with the level ℓ of concentration. We define the probability measure $\mathbb{P}$ on $\mathcal{L}$ as follows:

$$\forall\, h \in \mathcal{L},\ \mathbb{P}(h) = \sum_{\ell \in \mathcal{L}} \frac{\texttt{Vol}(\mathcal{D}_\ell)}{\texttt{Vol}(\mathcal{D})} \mathbf{1}_{\{h=\ell\}}.$$

  Next, for a given mesh $\mathtt{M}$, we define the density at the level $\ell \in \mathcal{L}$ and time t as:

$$\rho_t^{\mathtt{M}}(\ell) = \frac{N_t^{\mathtt{M}}(\ell)}{N_t^{\mathtt{M}}(\mathcal{D})\texttt{Vol}(\mathcal{D}_\ell)}, \tag{7}$$

  where $N_t^{\mathtt{M}}(\ell)$ is the number of particles from the sample $\mathbf{X}_t$ that are within the level ℓ of the mesh $\mathtt{M}$, and $N_t^{\mathtt{M}}(\mathcal{D})$ the size of $\mathbf{X}_t$. As a measure of variation of the concentration, we define the coefficient of variation:

$$\rho_{t,t+\Delta t}^{\mathtt{M}} := \mathbb{E}\left[\frac{|\rho_{t+\Delta t}^{\mathtt{M}} - \rho_t^{\mathtt{M}}|}{|\rho_{t+\Delta t}^{\mathtt{M}} - \rho_t^{\mathtt{M}}| + 1}\right], \tag{8}$$

  for a given spatial decomposition $\mathtt{M}$, and $\mathbb{E}$ the expectation under the measure $\mathbb{P}$ defined above.

  By construction, the coefficient of variation in Formula (8) will be greater as the difference of concentration between the samples $\mathbf{X}_t$ and $\mathbf{X}_{t+\Delta t}$ (with respect to the mesh $\mathtt{M}$) is greater. In other words, if $\rho_{t,t+\Delta t}^{\mathtt{M}}$ is close to zero, then no significant variation of concentrations between time steps is detected and we can keep the decomposition $\mathtt{M}$ to compute the agglomeration at time $t + \Delta t$. More precisely, if $\rho_{t,t+\Delta t}^{\mathtt{M}} < \epsilon_2$ where ϵ_2 is another tolerance threshold, we keep the mesh $\mathtt{M}$. Otherwise, we apply D2SD/quick D2SD to obtain a new spatial decomposition.

The criterion based on particle displacement fluctuation is less computationally expensive but also less accurate in some cases where the variation in the number of bins is balanced. Meanwhile, the criterion based on the difference of the particle concentration is more accurate but at higher computational cost (as it requires the computation of densities). Therefore, the criterion proposed in this article is based on a two-step verification, starting with the verification of the particle displacement fluctuation, and when necessary, checking the particle concentration difference.

Formally, considering the input $(\mathbf{X}_t, \mathbf{X}_{t+\Delta t}, \mathtt{M}_t, \epsilon_1, \epsilon_2)$, the re-meshing criterion consists of the three following steps (sketched in Fig. 2) coming before `step 1` in Section 2.1:

`step 0.a` Verify the percentage of the won/lost bins. Compute Δ_t`N_bins` in Equation (6). If $|\Delta_t$`N_bins`$| < \epsilon_1\%$ continue to `step 0.b`, otherwise continue to `step c.1`.

`step 0.b` Verify the difference of the concentration of particles. Compute $\rho^{\mathtt{M}}_{t,t+\Delta t}$ in Equation (8). If $\rho^{\mathtt{M}}_{t,t+\Delta t} < \epsilon_2$, then continue to `step 0.c2`, otherwise continue to `step 0.c1`

`step 0.c1` Update the spatial decomposition. Apply quick/full D2SD to the set of particles position $\mathbf{X}_{t+\Delta t}$ and deduce the decomposition $\mathtt{M}_{t+\Delta t}$.

`step 0.c2` Keep the previous mesh. Set $\mathtt{M}_{t+\Delta t} = \mathtt{M}_t$.

Note that `step 0.b` does not add computational cost when the decomposition `M` is good enough for the snapshot $\mathbf{X}_{t+\Delta t}$ since in this case the densities must be computed for the agglomeration step.

We now can summarise the steps of fast-D2SD, for each time-iteration:

`steps 0 + modified step1 + step 2 (fully/partly)`, where in the `modified step 1` we add a criterion to apply the uniformity test (i.e. the original `step 1`) only when $n_t := \sum_{\ell\in\mathcal{L}} N_t(\ell) < n_{crit}$, with n_{crit} a critical sample size that can be set according to a desired error level ϵ_3. The aim of this modification is to lower the computational cost due to uniformity checks, and the tolerance level can be chosen knowing that the larger the number of particles, the lower the numerical error within the computation of agglomeration events. More precisely, in [15, Fig. 11] it has been numerically shown that -in the case of a uniformly distributed population of particles- a relative error of $\epsilon_3 = 10^{-1}$ is obtained by applying only `step 2` of the D2SD algorithm with a family of $n_{crit} = 100$ particles, against a relative error of 10^{-2} by applying the full D2SD algorithm.

3. Testing the efficiency of the adapted D2SD algorithm

In this section, we evaluate the accuracy and efficiency of each of the simplification options proposed for the D2SD algorithm. This is done through a series of two-dimensional numerical experiments that consider both common situations and more specific scenarios where one of the indicators may fail.

To quantify the error resulting from the use of an adapted D2SD algorithm instead of the original D2SD algorithm, we introduce the measure $\mathcal{E}(k, \mathtt{M}_0)$. It is defined as the relative error between the agglomerations events computed using an initial mesh $\mathtt{M}_0$ and the agglomeration events computed using the full D2SD algorithm $\mathtt{M}^*_k$, at the time iteration k. Precisely, we define:

$$\mathcal{E}(k, \mathtt{M}_0) = \frac{|\mathcal{A}(\mathtt{M}^*_k) - \mathcal{A}(\mathtt{M}_0)|}{\mathcal{A}(\mathtt{M}^*_k)}, \tag{9}$$

where, for any mesh M,

$$\mathcal{A}(M) = \sum_{\ell\in\mathcal{L}} \beta \frac{N_t^M(\ell)^2}{\mathtt{Vol}(\mathcal{D}_\ell)} \Delta t, \tag{10}$$

is the number of agglomeration events computed using the PBE formulation on the spatial decomposition obtained (and $\alpha = 1$). Δt is the time step, and $\beta = \beta_{v_1,v_1}$ the primary particles collision kernel.

3.1. Testing the quick D2SD algorithm without re-meshing criterion

We start by assessing the accuracy and efficiency of the quick D2SD algorithm, which relies on the 60th percentile as an a-priori value for the level threshold (instead of the iterative search of the optimal one). For that purpose, we perform 2D simulation in the same five cases investigated in the previous paper, which were selected to represent a range of situations that can happen in CFD simulations [15]:

i. homogeneous distributed population of particles;

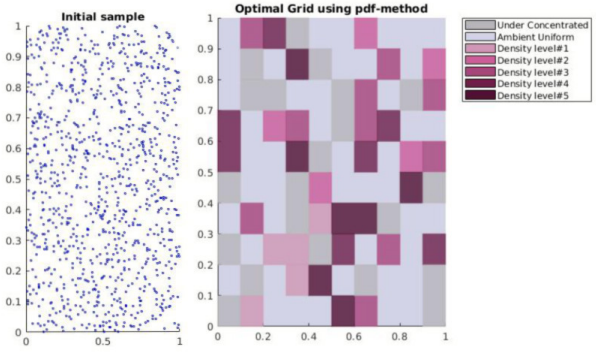


Figure 3: Testing D2SD with $\hat{\lambda} = Q_{60}$ for a population of particles uniformly distributed. In this case $\lambda^* = Q_{60}$.

ii. slightly non-homogeneous distribution;

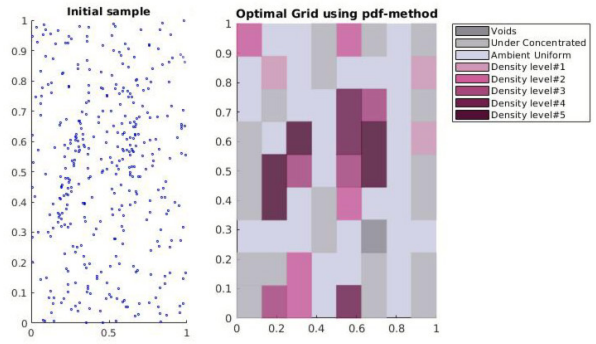


Figure 4: Testing D2SD with $\hat{\lambda} = Q_{60}$ for a population of particles containing clusters of particles with a mild concentration. In this case $\lambda^* = Q_{60}$.

iii. medium non-homogeneous distribution;

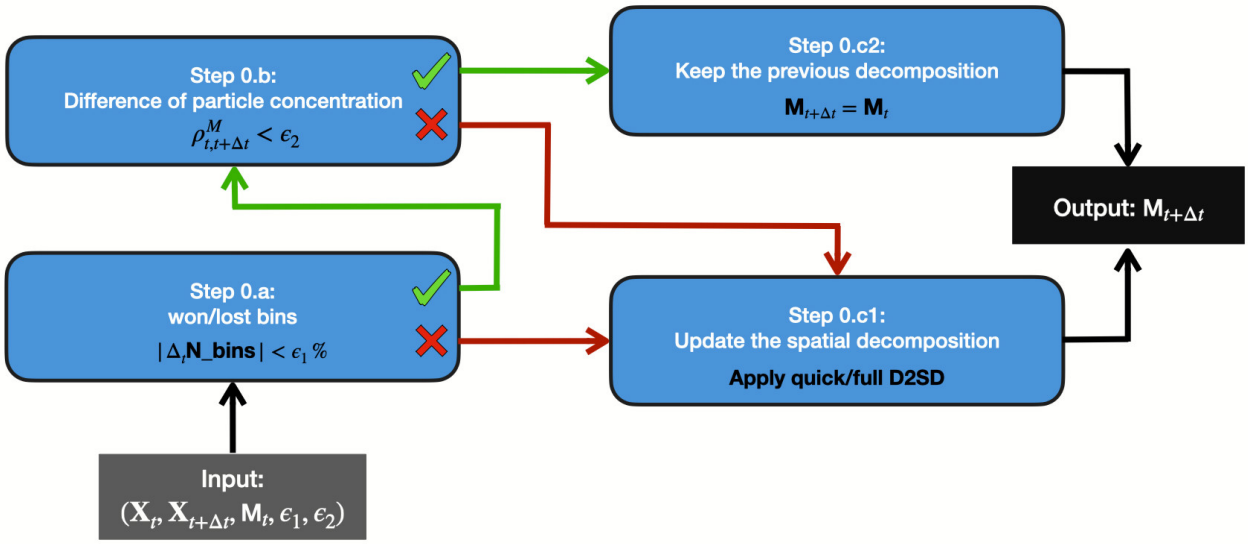


Figure 2: Sketch of the re-meshing criterion.

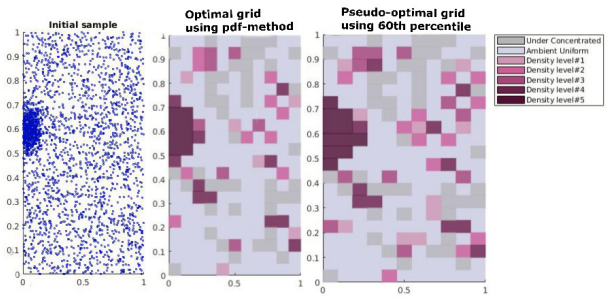


Figure 5: Testing D2SD with $\hat{\lambda} = Q_{60}$ for a population of particles containing clusters of particles with a medium concentration. Plot of the mesh with $\lambda = \hat{\lambda}$ is included since $\lambda^* \neq \hat{\lambda}$.

CASE	Thresholds evaluated	$\frac{\lambda^*}{\hat{\lambda}}$	$\frac{\mathcal{A}^*}{\mathcal{A}}$	$\frac{\text{CPU}^*}{\text{CPU}}$
Homogeneous	3	1	1	1
Slightly non-homogeneous	2	1	1	1
Medium non-homogeneous	3	1.08	1.15	1.51
Highly non-homogeneous	9	1.25	1.01	2.21
Symmetric	4	1	1.019	3.19

Table 1: Testing D2SD with $\hat{\lambda} = Q_{60}$ for a population of particles: uniform distributed, containing clusters of particles with a mild-medium or high concentration or with a symmetric distribution.

iv. highly non-homogeneous distribution;

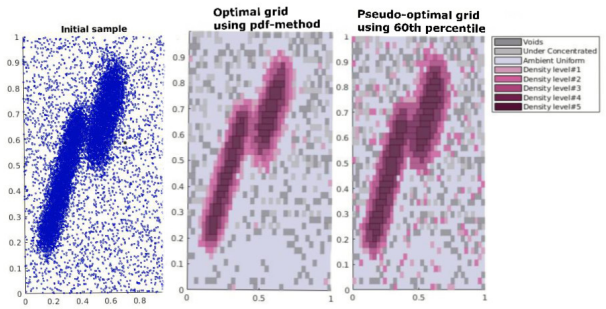


Figure 6: Testing D2SD with $\hat{\lambda} = Q_{60}$ for a population of particles: containing clusters of particles with a high concentration. Plot of the mesh with $\lambda = \hat{\lambda}$ since $\lambda^* \neq \hat{\lambda}$.

v. symmetric case (with an empty region).

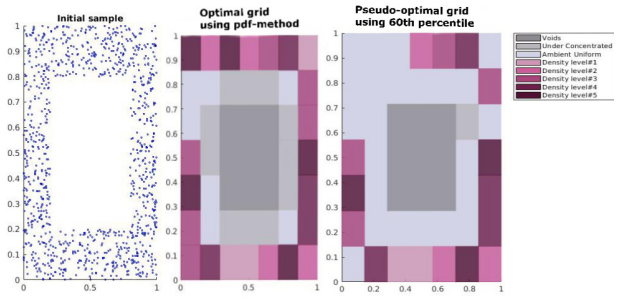


Figure 7: Testing D2SD with $\hat{\lambda} = Q_{60}$ for a population of particles with a symmetric distribution. In this case $\lambda^* = \hat{\lambda}$ for the first iteration. Plot of the mesh with $\lambda = \hat{\lambda}$ and only one iteration is included.

Figures 3 - 7 display the five cases considered, showing the spatial distribution (left) together with the optimal mesh decomposition and the mesh decomposition obtained with $\hat{\lambda} = Q_{60}$ (when it differs from the optimal mesh). It is complemented by the results in Table 1, which contains information about: the

number of thresholds tested in the `step 2.f` of the original D2SD algorithm, the ratio between the optimal threshold and the 60th percentile, the ratio between the number of agglomerations for the optimal mesh $\mathcal{A}^*$ or for the mesh $\mathcal{A}$ obtained with $\hat{\lambda} = Q_{60}$, and the ratio between the computational times of the original D2SD algorithm and the quick D2SD method.

The first key result is that the quick D2SD mesh coincides with D2SD (optimal mesh) in the cases of a uniform spatial distribution or a slightly non-homogeneous distribution. This means that the optimal threshold corresponds to the 60th percentile, i.e. $\lambda^* = Q_{60}$. In these cases, the quick D2SD algorithm provides the exact same results (without error). In the other cases, the optimal threshold does not coincide with the 60th percentile, the largest difference being reached for the highly non-homogeneous case. Yet, despite this difference in the threshold, the error made in the number of agglomeration events does not exceed 15% (with the biggest difference occurring for the medium non-uniform distribution). In terms of computational costs, the use of the quick D2SD algorithm can speed up the results up to 200% or 300% in some cases compared to the original D2SD algorithm.

To further analyse these results, we evaluate the ratio of the number of agglomeration events $\frac{\mathcal{A}^*}{\mathcal{A}}$ in each of the five cases as a function of the number of initial particles. This allows to assess the balance between the accuracy and efficiency of the quick D2SD algorithm (i.e. numerical error to computational cost). The results are presented in Figure 8, where we can verify how the accuracy of the reduced D2SD algorithm stabilises as the number of particles n increases, and the difference in cost increases with n. In general, we can say that $\lambda = Q_{60}$ is a suitable choice for the threshold in order to decrease the computational cost. The only difficulty arises when the number of particles is small, where it may be preferable to apply the original D2SD algorithm.

These tests confirm that the quick D2SD algorithm can be safely used to efficiently compute particle agglomeration in most of the cases of interest, as long as the expected results are within a 15% error range.

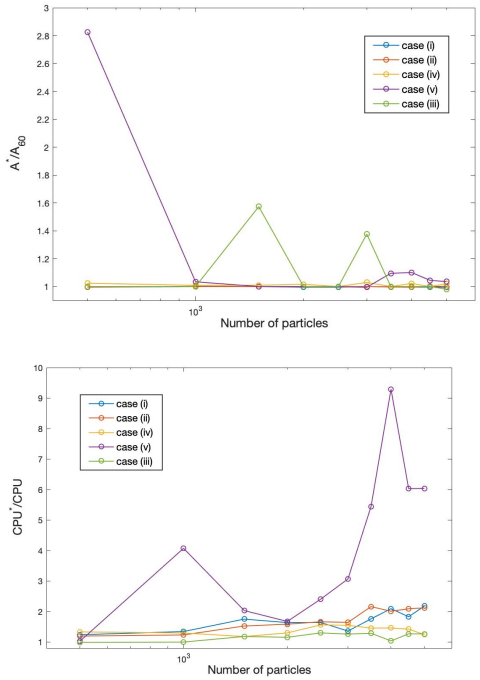


Figure 8: Testing D2SD with $\hat{\lambda} = Q_{60}$ for a population of particles of size $n = \{500l : l \in \mathbb{N}, 1 \leq l \leq 10\}$ in the case: homogeneous, slightly, medium and highly non-homogeneous, and symmetric.

3.2. Testing the re-meshing criterion

We assess here the criterion that determines when the D2SD algorithm should be applied again, assuming that it has already been applied on a previous time step. For that purpose, we design test cases that account for both inhomogeneities in the spatial distribution of particles and the motion of particles with time. In Section 3.2.1, we start by describing and testing the D2SD algorithms on each case to set up the value of the thresholds ϵ_1 and ϵ_2. Then, the fast D2SD algorithms are tested on selected cases and their efficiency is assessed in Section 3.2.2.

3.2.1. Setting-up thresholds ϵ_1 and ϵ_2

We have selected six different cases that are presented in the following. For each case, we apply the following methodology: (1) we start with a set of initial particle positions $\mathbf{X}_1$; (2) we apply the D2SD algorithm to obtain the initial optimal decomposition $\mathtt{M}_1^*$; (3) particles are displaced during a time step Δt and (4) we apply again the D2SD algorithm (reference case) and we compare the results to those obtained with the re-meshing criteria. This allows to determine if a new spatial decomposition needs to be applied and then compute the number of agglomeration events. This procedure is repeated for ten iterations intending to assess the evolution of the error and indicators as the simulation progresses. For each iteration k, we compare the two meshes obtained, i.e. $\mathtt{M}_k$ versus $\mathtt{M}_k^*$. To that extent, we compute the evolution of the indicators $\Delta_{m,k}\mathtt{N\_bins}$ (i.e. percentage of won/lost bins) and $\rho_{m,k}^{\mathtt{M}_m}$ (difference of particle concentration with respect to the previous and initial mesh) for various time steps $k = 2, \ldots, 10$ and initial time $m = 1, k-1$.

Each case and the corresponding results are described in the following:

a. Steady-state homogeneous particle distribution:
This corresponds to a case where the spatial distribution of particles is homogeneous at the beginning of the simulation and remains so with time, as displayed in Figure 9 (top).
Due to the conservation of homogeneity with time, it can be seen that the optimum splitting obtained by applying the D2SD algorithm every time step does not change significantly (bottom figures). This is further confirmed in the graphs on the right in Fig. 9: it shows that the number of bins won/lost (obtained when comparing the results with either the previous iteration or with the initial iteration) remains below 10%. In addition, the concentration difference with respect to the previous/initial iteration is almost always below 0.15 (except for the seventh iteration, which remains around 0.18).
Drawing on the fact that we expect re-meshing to be unnecessary in this case, these results illustrate that the threshold for the won/lost bins can be realistically taken close to the value of 15-20%.

b. A moving cluster in a homogeneous distribution:
This illustrates the cases where a cluster within a homogeneous distribution simply changes location with time, as shown in Figure 10.
The results obtained for the number of won/lost bins show that the number of bins changes significantly both w.r.t. the previous iteration or w.r.t. the initial iteration. These trends hold especially as the cluster approaches the periodic boundaries (see the 2-3 last iterations). Meanwhile, a high difference in the concentration is observed all the time (roughly around 0.4) regardless if we compare the current iteration to the previous or initial one. This is due to the constant change of the cluster location with time, which is never at the same location between successive iterations (see the plots on the right in Fig. 10).
This case illustrates the interest of relying on the concentration differences and not only on the number of won/lost bins. In fact, if a threshold of $15-20\%$ for the won/lost bins is used, the results would suggest to re-mesh all the time except in the third and fourth iterations. Yet, we expect here to re-mesh between all iterations due to the cluster different location (no overlap occurs between two iterations). Applying a second criterion on the concentration difference is thus mandatory: drawing on the concentration difference measured here and to the one for case (a.), it appears that a threshold of $0.15-0.2$ for the concentration difference is a suitable candidate.

c. A cluster that disperses uniformly in space:
This illustrates cases where a particle cluster disperses in the domain to reach a homogeneous distribution at longer times, as shown in Figure 11.
The dispersion of the initial cluster in space translates into a strong evolution of the optimum splitting obtained by applying the D2SD algorithm every time step. This is especially true when comparing results with the one got at the 1^{st} iteration, where the distribution is still far from a homogeneous one (bottom figures). As a result, the number of won/lost bins compared to the initial iteration is always very high. Yet, the number of won/lost bins w.r.t. the previous iteration quickly reaches values within the $15-20\%$ threshold as a stationary case is reached (see the right plots of Fig. 11). The same trend is observed for the concentration difference, which is always around 0.5 when comparing it to the initial iteration but quickly decreases when comparing to the previous iteration.
Using the threshold of $15-20\%$ for the won/lost bins and $0.15-0.2$ for the concentration difference, the present results would suggest to re-mesh the first two iterations. After the third iteration, a steady-state is reached and re-meshing is not likely anymore.

d. A cluster that disperses non-uniformly in space:
The case of an initial cluster with a radial displacement shows how the algorithm reacts to a time-evolving non-homogeneous distribution. As seen in Figure 12 (top), the initial cluster is located in the centre of the square and, due to the radial displacement, the distribution evolves towards a high concentration at the border.
This peculiar dispersion is well tracked by applying the D2SD algorithm every time step (bottom figures). Looking at the evolution of the number of won/lost bins (see right plots in Figure 12), it can be seen that it quickly decreases to values below -20% when comparing to the initial iteration while small fluctuations are reached when comparing consecutive iterations. Meanwhile, the concentration difference w.r.t. the initial value grows steadily from 0.2 to a plateau value of 0.4 in 5-6 iterations. The concentration difference w.r.t. consecutive iterations stays around 0.2 and even drops to lower values after 6 iterations.
Resorting to the thresholds mentioned previously, this case suggests that a first re-meshing will be performed at iteration 2. Consecutive re-meshing are to be expected between iterations 2 and roughly 8. For the two last iterations, no re-meshing would be applied. This is because a steady-state is close to being reached (where particles are all on the border), leading to lower values of won/lost bins and concentration difference between consecutive iterations.

e. Local injection in a homogeneous distribution:
This case is expected to be representative of typical CFD simulations with an injection of particles. As seen in Figure 13 (top), the initial distribution is homogeneous and becomes non-homogeneous with time due to the local injection of particles (this is not compensated by the uniform displacement since the injection is the dominant phenomenon).
This complex evolution of the distribution is well captured by applying the D2SD algorithm every time step (bottom figures). However, as displayed in Figure 13 (right), the won/lost bins with respect to the initial iteration increases due to the development of a strongly non-homogeneous distribution. This is especially true in the first few iterations, where even the won/lost bins between consecutive iterations peaks up to 200%. However, at longer times, both the won/lost bins and concentration difference between consecutive iterations reach values similar to the homogeneous distribution in time. Again, this is due to the establishment of a steady-state situa-

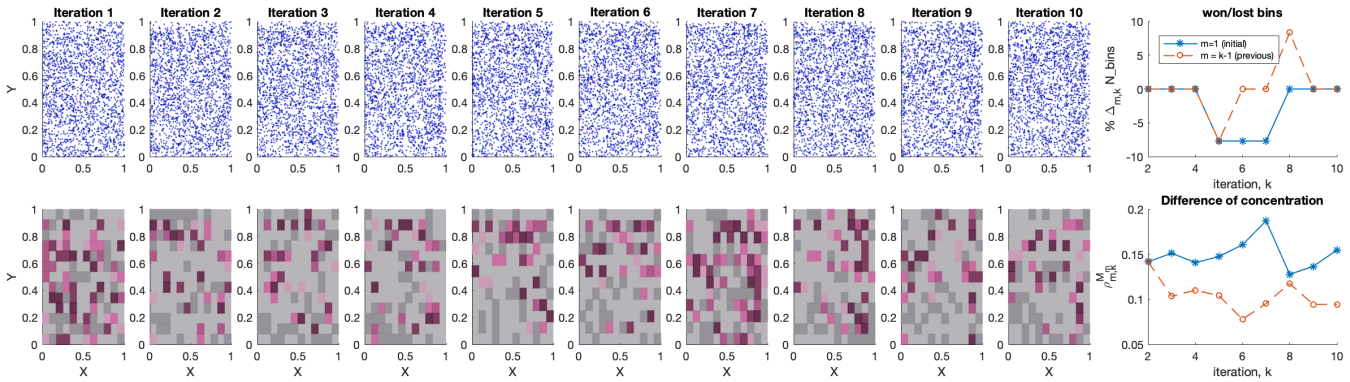


Figure 9: Steady-state homogeneous particle distribution, with parameter $\Delta t = 0.1$. Evolution of the won/lost bins $\Delta_{m,k}$`N_bins` (right-top figure) and difference of particle concentration $\rho_{m,k}^{M_m}$ (right-bottom figure) is shown with respect to: initial iteration ($m = 1$, in blue and $*$ symbol) and previous iteration ($m = k - 1$, in orange and $\circ$ symbol).

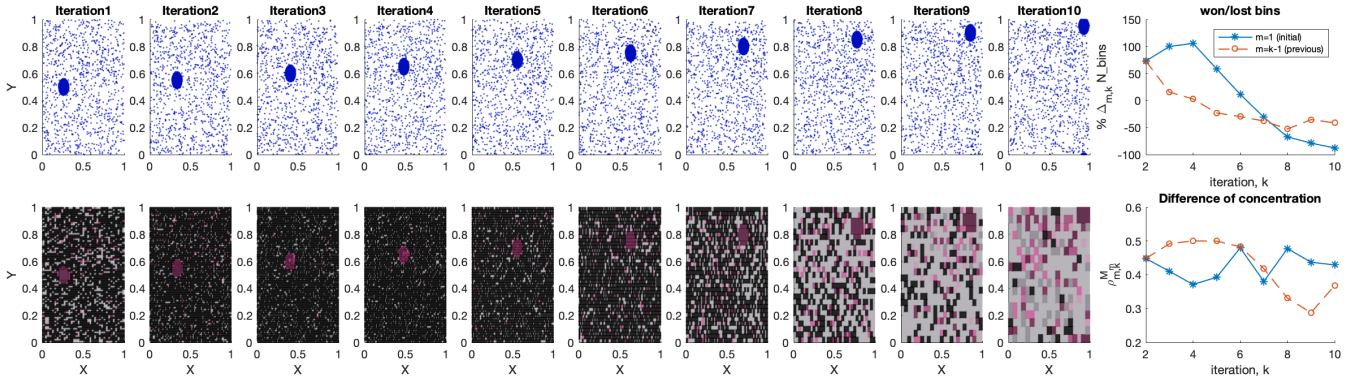


Figure 10: Moving cluster in a homogeneous distribution, with parameter $\Delta t = 0.1$. Evolution of the won/lost bins $\Delta_{m,k}$`N_bins` (right-top figure) and difference of particle concentration $\rho_{m,k}^{M_m}$ (right-bottom figure) is shown with respect to: initial iteration ($m = 1$, in blue and $*$ symbol) and previous iteration ($m = k - 1$, in orange and $\circ$ symbol).

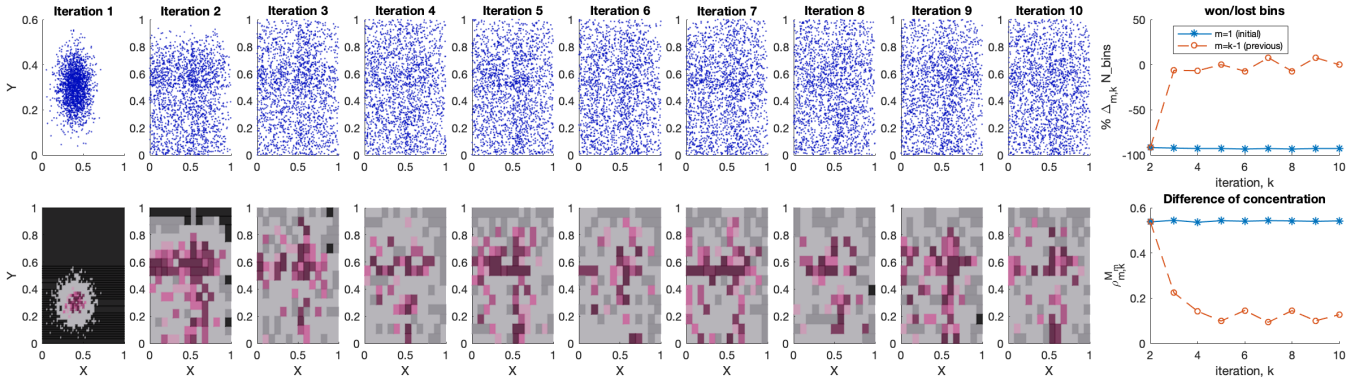


Figure 11: Cluster that disperses uniformly in space, with parameter $\Delta t = 0.5$. Evolution of the won/lost bins $\Delta_{m,k}$`N_bins` (right-top figure) and difference of particle concentration $\rho_{m,k}^{M_m}$ (right-bottom figure) is shown with respect to: initial iteration ($m = 1$, in blue and $*$ symbol) and previous iteration ($m = k - 1$, in orange and $\circ$ symbol).

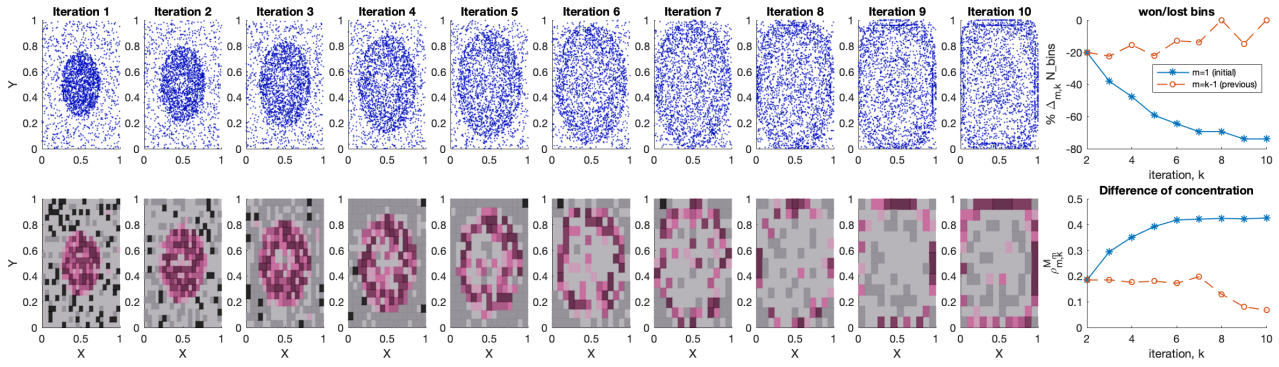


Figure 12: Cluster that disperses non-uniformly in space, with parameter $\Delta t = 0.1$. Evolution of the won/lost bins $\Delta_{m,k}$N_bins (right-top figure) and difference of particle concentration $\rho_{m,k}^{M_m}$ (right-bottom figure) is shown with respect to: initial iteration ($m = 1$, in blue and $*$ symbol) and previous iteration ($m = k-1$, in orange and $\circ$ symbol).

tion, where the number of particles remains constant in the domain (due to the balance between newly injected particles and particles going out of the square of observation).
As in case (d.), this case illustrates the interest of relying on a two-step process: the won/lost bins with a threshold of $15-20\%$ provide a quick first acceptance/rebuttal which is completed by the 2nd criterion on the concentration difference with a threshold around $0.15-0.2$ (when needed). Here, re-meshing is expected to occur during the second and third iterations and again around iteration 9.

f. Two clusters (expanding/shrinking in space):
This case is representative of a situation where two initial clusters evolve in time (see also Figure 14): one is expanding in space while the other one is shrinking. The rates of expansion and shrinking are designed with opposite values, such that the number of bins remains constant.
As seen in the right plots of Figure 14, the number of won/lost bins always remains below 10% regardless if it is compared to the initial or previous iteration. The information on the change in particle positions comes from the concentration difference, which is seen to steadily increases from 0.1 to 0.5 when compared to the initial iteration. The concentration difference w.r.t. the previous iteration remains relatively low (below 0.16).
This case thus highlights the interest of the two-step process. Here, we expect the 1^{st} step (won/lost bin) to be accepted every time, but the concentration difference suggests to re-mesh every few iterations because of the change in particle locations. Thus, resorting to a single-step process (i.e. using information on won/lost bins only) would lead to incorrect results while the two-step process does capture this change. The concentration difference will provide more information on the actual changes in concentrations.

All these cases support the use of a re-meshing criteria based on two indicators, namely the number of won/lost bins and the concentration difference. The next step consists of applying dynamically the re-meshing criteria described by `steps 0.1-0.4` in Section 2.2.2 and evaluating the error $\mathcal{E}(k, \mathtt{M}_k)$ made on the computation of particle agglomeration (see Equation (9)), with $\mathtt{M}_k$ the output of the criteria at iteration k. This means that the criteria are applied at every iteration: if rejected, the current iteration becomes the new reference mesh but, if accepted, the reference mesh is left untouched (see Figure 2).

3.2.2. *Efficiency of re-meshing*

Here, we analyse the results obtained with the re-meshing criteria applied dynamically. For that purpose, we focus on the following cases (which were seen to be key in the previous figures): the homogeneous distribution in time (case a), the uniformly dispersion of clusters (case c), the local injection (case e) and the two clusters (case f). Based on the results of case (a), we set the tolerance thresholds as $\epsilon_1 = 20(\%), \epsilon_2 = 1/6$. The value $\epsilon_2 = 1/6$ can also be justified on a bound of 20% error for the difference $|\rho_{k-1}^{\mathtt{M}_{k-1}} - \rho_k^{\mathtt{M}_{k-1}}|$, during iteration k.

As displayed in Figure 15, for each of these cases and each (time) iteration k, we illustrate the evolution of the indicators $\Delta_{k-1,k}$N_bins, $\rho_{k-1,k}^{\mathtt{M}_{k-1}}$ and the error $\mathcal{E}(k, \mathtt{M}_k)$ ($\mathtt{M}_k$ being the output of the criteria at iteration k). For the sake of clarity, we have also added markers around the iteration identifier (see x-axis on plots 15(a) and 15(b)) to highlight every time a re-meshing is suggested by each criterion. Several conclusions can be drawn from this set of figures. First, we observe that the error made in evaluating the number of agglomeration events always remains below 0.4, even in complex cases such as case (f). Second, in the cases where the population of particles reaches a state of uniform distribution (cases (a) and (c)), the criteria suggest keeping the spatial decomposition unchanged (i.e. no re-meshing is applied), in accordance with the small error made on the calculation of agglomerations.

To further analyse how the setup of the thresholds ϵ_1 and ϵ_2 influences the error made on the prediction of agglomeration, we focus on the case of clusters expanding/shrinking (case f). For that purpose, two different pairs of thresholds are used for (ϵ_1, ϵ_2): $\{(10\%, \frac{1}{6}), (20\%, \frac{1}{11})\}$. The results obtained are shown in Figure 16: it can be seen that, by tuning the values of ϵ_i, a lower error on the agglomeration prediction is obtained. This is related to the fact that, with lower tolerance thresholds, a higher number of re-meshing events are triggered. Indeed, compared to the error associated with a threshold of 20%, the error associated

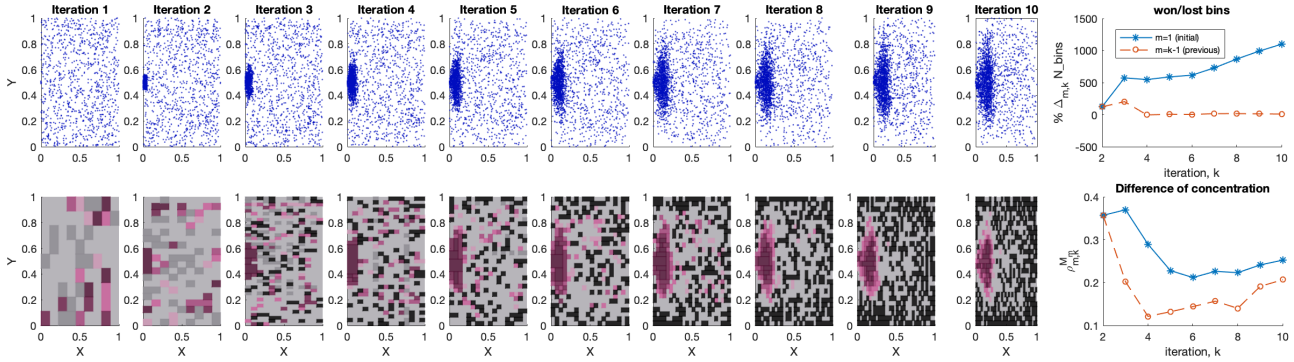


Figure 13: Local injection in a homogeneous distribution, with parameter $\Delta t = 0.1$. Evolution of the won/lost bins $\Delta_{m,k}$`N_bins` (right-top figure) and difference of particle concentration $\rho_{m,k}^{M_m}$ (right-bottom figure) is shown with respect to: initial iteration ($m = 1$, in blue and $*$ symbol) and previous iteration ($m = k - 1$, in orange and $\circ$ symbol).

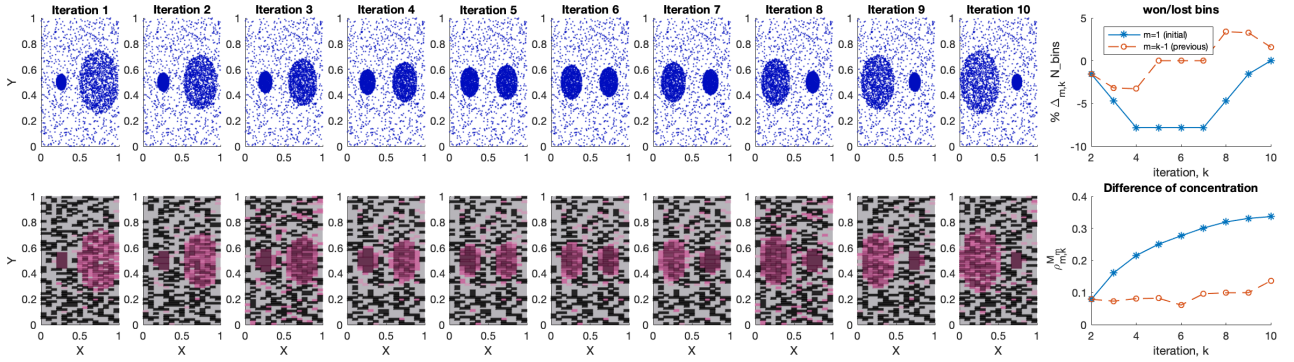
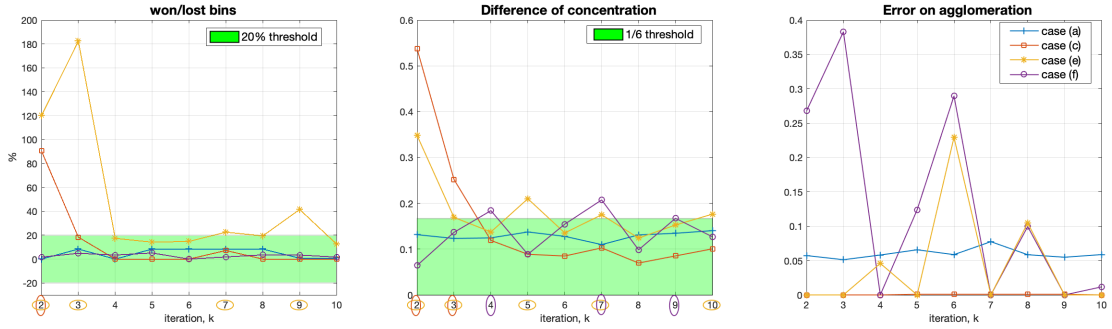


Figure 14: Two clusters (expanding/shrinking in space), with parameter $\Delta t = 0.1$. Evolution of the won/lost bins $\Delta_{m,k}$`N_bins` (right-top figure) and difference of particle concentration $\rho_{m,k}^{M_m}$ (right-bottom figure) is shown with respect to: initial iteration ($m = 1$, in blue and $*$ symbol) and previous iteration ($m = k - 1$, in orange and $\circ$ symbol).



(a) Evolution of $\Delta_{k-1,k}$`N_bins`

(b) Evolution of $\rho_{k-1,k}^{\mathtt{M}_{k-1}}$

(c) Evolution of $\mathcal{E}(k, \mathtt{M}_k)$

Figure 15: Illustration of the studied cases: uniform case (case a), dispersing cluster (case c), injection of particles (case e) and extraction/expansion of clusters (case f). Plots for the evolution of the won/lost bins $\Delta_{k-1,k}$`N_bins` and difference of concentration $\rho_{k-1,k}^{\mathtt{M}_{k-1}}$ are shown using thresholds $\epsilon_1 = 20\%$ and $\epsilon_2 = 1/6$. Additionally, the corresponding error on agglomeration events is plotted in Fig. 15(c)

with a threshold of 10% decreases by almost half (an error of about 0.4 against one of about 0.25). However, a higher accuracy in the results obtained also comes with higher computational costs since more re-meshing occurs. These experiments confirm that the proposed re-meshing criteria is a suitable candidate to increase the computational efficiency of the D2SD algorithm while respecting a user-defined criterion on the error made in the evaluation of particle agglomeration events.

3.3. *Testing quick D2SD not every time step*

In this section, we assess the accuracy and efficiency of the algorithm combining both the quick D2SD algorithm and the re-meshing criteria. For that purpose, we consider case (f) with clusters expanding/shrinking and we compare the error made between the computation of agglomeration events implementing the quick D2SD versus full D2SD algorithms, both using the re-meshing criterion, i.e. `Algo 3` versus `Algo 1`. The results for $\epsilon_1 = 20(\%)$ and $\epsilon_2 = 1/6$ are displayed in Figure 17: we observe that the error propagates as the iterations progress but remains within acceptable levels. When applying `Algo 1` (full D2SD with re-meshing criterion), we can observe a decrease of the error made between the computation of agglomeration events. Whereas by applying `Algo 3` (quick D2SD with re-meshing criterion), the error propagates and increases as iterations progress. Regarding the computational time, denoting by $\texttt{CPU}^*$ the cumulative computational time (over all iterations) associated with the implementation of the D2SD algorithm: `Algo 1` was executed in approximately $0.40[\texttt{CPU}^*]$, while `Algo 3` took $0.29[\texttt{CPU}^*]$. These results underline the great advantage, as far as computational cost is concerned, of the fast versions of the D2SD algorithm proposed in this paper. In particular, it highlights the improvement regarding the computational cost when the `Algo 3` is implemented. However, this gain comes with a loss of accuracy, which can be clearly seen in the three last iterations in Figure 17. Therefore, when implementing the quick version of the D2SD algorithm, it is suggested to refresh from time to time the spatial decomposition using the full version in order to stop the propagation of numerical errors.

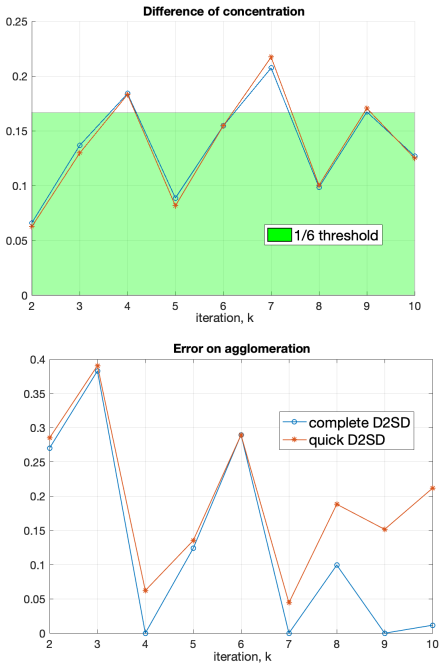


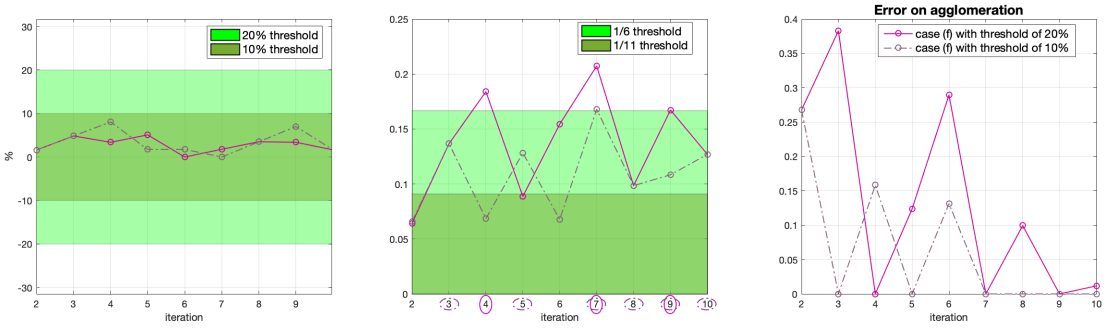
Figure 17: Evolution of the difference of particle concentration and error on agglomeration events in the case (f.): expansion/shrinking of clusters. Results for `Algo 3` (quick D2SD plus re-meshing criterion) are plotted in orange while the results for `Algo 1` (full D2SD plus re-meshing criterion) are plotted in blue.

4. Application to a practical case and validation

In this section, the quick D2SD algorithm is coupled to a 3D simulation of particle agglomeration to assess the accuracy and efficiency of the method. More precisely, the quick D2SD is applied and compared to numerical results obtained with micro-scale simulations for the agglomeration of particles undergoing purely diffusive motion. Since the aim is to validate the D2SD algorithm, we focus our attention on a case involving non-homogeneous repartition of particles, here with a local injection. The case studied together with the micro-scale model and theoretical expectations on the rate of agglomeration are presented in Section 4.1 while the use of the quick D2SD algorithm and the comparison to these results are discussed in Section 4.2.

4.1. *Micro-scale simulation of particle agglomeration with local injection*

Case studied: local particle injection. A case of non-homogeneous particle repartition has been selected in order to highlight the interest of using the D2SD algorithm in such complex cases, as well as to assess the accuracy of the quick D2SD algorithm. We have opted here to simulate the case of particles undergoing purely Brownian motion since a theoretical estimation of the agglomeration rate can be computed in that case. More precisely, we focus on the non-homogeneous case of particles within a spherical domain of radius R_s, but with a higher concentration around the sphere centre. Since the aim of these micro-scale simulations is to extract from the results statistical information on the agglomeration rate, the case studied has to be a steady-state. For that reason, we did not



(a) Evolution of $\Delta_{k-1,k}\mathtt{N\_bins}$ (b) Evolution of $\rho_{k-1,k}^{\mathtt{M}_{k-1}}$ (c) Evolution of $\mathcal{E}(k,\mathtt{M}_k)$

Figure 16: Illustration of the extraction/expansion of clusters (case f). Plots for the evolution of the won/lost bins $\Delta_{k-1,k}\mathtt{N\_bins}$ and difference of concentration $\rho_{k-1,k}^{\mathtt{M}_{k-1}}$ are shown using thresholds $(\epsilon_1,\epsilon_2) = (20\%, \frac{1}{6})$ (light green) and $(\epsilon_1,\epsilon_2) = (10\%, \frac{1}{11})$ (dark green). Additionally, the corresponding error on agglomeration events is plotted in Fig. 15(c)

set-up a simulation within a periodic sphere in which particles are initially non-homogeneously distributed and then move according to purely diffusive motion since the expected steady state corresponds to homogeneously distributed particles within the sphere (see case c in Section 3.2.1). We rather opted for a continuous injection of particles near the sphere centre, which results in a non-homogeneous steady-state where particles are more concentrated near the sphere centre. Particles reaching the domain boundary are removed from the simulation. When a collision between two particles is detected, one of the colliding partners is removed from the simulation (this corresponds to the sticky case where particles adhere to each other upon colliding).

Theoretical collision rate. As mentioned previously, we have focused on this case since the collision rate can be estimated theoretically (details about the derivation of the collision rate are provided in Appendix A). Here, the average number of particles evolves as

$$\frac{\mathrm{d}}{\mathrm{d}t}\langle N(t)\rangle = -\frac{\beta}{\mathtt{Vol}(\mathcal{D})}\langle N(t)\,(N(t)-1)\rangle \tag{11}$$

where the collision rate β is given by

$$\beta = \frac{8\pi R_p^2\,\sigma}{\sqrt{2\pi\gamma}}. \tag{12}$$

It should be noted here that this formula for the collision rate was obtained, assuming that particles are homogeneously distributed within the domain. Actually, in this case, the same formula holds, although particles are not distributed homogeneously: this is because in each circular region around the centre of the domain, the concentration can be considered as locally homogeneous. This means that the same collision rate holds and that a higher number of collisions occurs in regions with higher concentrations (since the collision frequency is multiplied by the local particle concentration to compute the number of collision events).

Microscopic Langevin simulations. We perform here numerical simulations of particles undergoing purely Brownian motion using the following Langevin equations:

$$\begin{aligned} d\mathbf{X}_i &= \mathbf{V}_i dt, \\ d\mathbf{V}_i &= -\gamma\mathbf{V}_i dt + \sigma dB_i(t), \end{aligned} \tag{13}$$

with the particle position $\mathbf{X}_i$, the fluid velocity $\mathbf{V}_i$, the friction γ, the noise amplitude σ and a family of independent Brownian motion $(B_i(t); t \geq 0)_i$. All the particles are considered to have the same radius R_p. N_{inject} particles are injected in the domain at the particle centre every N_{iter} iterations. Particles reaching the domain boundary (here at $R_s = 0.5$) are eliminated from the simulation. This allows to obtain a steady-state once the number of particles entering the domain is compensated by the number of particles exiting the domain. The motion of particles is computed with a time step small enough that the average particle displacement within one time step is close to the particle radius, i.e. $|\mathbf{X}_i(t+\Delta t) - \mathbf{X}_i| \simeq R_p$. This allows to detect the collision between two particles using geometric arguments assuming a ballistic regime within the time step (i.e. the motion of a particle within one time step follows a straight line). When such a collision is detected, one of the partner is removed from the simulation. This choice corresponds to the case of sticky collisions (i.e. a collision does lead to agglomeration) The collision rate is extracted from these simulations by recording the number of collisions N_{coll} that occurred during a certain amount of time t_{simu}. Since the particles are not homogeneously distributed in the domain, the collision rate is analysed as a function of the distance from the sphere centre (where particles are injected at a given frequency). For that purpose, we define a number of spherical rings according to their distance from the centre, here comprised between $r\,R_s/50$ and $(r+1)\,R_s/50$, with $r = 0,\ldots,49$. Then, drawing on Eq. (11) which gives the evolution of average number of particles in the domain, the collision rate is estimated within each ring:

$$\beta_{approx.} = \frac{N_{coll}}{t_{simu}N_{p,n}(N_{p,n}-1)} \tag{14}$$

where $N_{p,n}$ is the average number of particles in the ring labelled r.

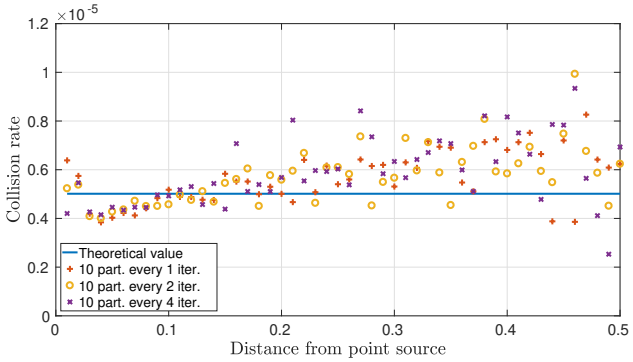


Figure 18: Comparison between the theoretical value of the collision rate and numerical estimations obtained with fine Langevin simulations using various particle concentrations.

Results obtained. Figure 18 displays the comparison between the theoretical value of the collision rate and the results obtained with the numerical simulations as a function of the distance from the point source (where particles are injected locally). As expected from standard PBE formulations, the collision rate appears to be independent of the distance from the point source despite the variation in the particle concentration. The results are less accurate as the distance from the point source increases since the number of particles present in these regions is lower (leading to higher statistical noise).

4.2. Simulations with fast D2SD

The next step consists in coupling the quick D2SD algorithm to standard CFD simulations. Here, the particle positions generated by the fine Langevin simulations are used as an input data for the quick D2SD algorithm every time step. This allows to test the accuracy and efficiency of the quick D2SD algorithm.

The quick D2SD algorithm has been applied to 1000 consecutive iterations sampled from the simulation with 10 particles injected every 4 time step. We have used a spatial transformation proposed in [15], that allows to analyse the homogeneity in the particle concentration using spherical coordinates instead of Cartesian ones (which better fit the present case). The results obtained for the identification of high concentration regions from the quick D2SD are illustrated in Fig. 19: it can be seen that clusters of particles are clearly identified near the sphere centre while other regions further from the centre can also be identified as having a concentration slightly higher than the average one.

Using Eq. 10, we compare the number of agglomeration events computed using the D2SD algorithm to the one directly extracted from fine Langevin simulations and the one provided by applying the PBE formulation on the whole domain. The results are shown in Table 2, which shows that the evaluation of the number of agglomeration events is greatly improved using the D2SD algorithm compared to the results obtained using the PBE formula over the whole domain. In fact, the PBE formulation applied over the whole domain leads to a significant underestimation of the number of agglomeration events (around 75%). Meanwhile, the error made using the D2SD algorithm is much smaller: it increases slightly using the quick D2SD algorithm with re-meshing criterion instead of the full D2SD algorithm (from 18.5 % to 20.2 %) but the computational costs are reduced by a factor 3 with the fast D2SD algorithm.

CASE	**Langevin simulation**	**Full D2SD**	**Quick D2SD**	**PBE one cell**
Nb. agglo. events	523	426	417	130

Table 2: Values of the number of agglomeration events over 1000 iterations computed using the fine Langevin simulation, the full or quick D2SD algorithm with re-meshing and the PBE formula over the whole domain (one cell only).

5. Conclusion

In this paper, a fast version of the data-driven spatial decomposition (D2SD) algorithm has been developed. As for the original D2SD algorithm, the fast D2SD allows to detect non-homogeneous particle concentrations within a regular volume, relying solely on the information coming from the set of particles (without requiring other input parameters). The quick D2SD algorithm developed here combines two simplifications to reduce the computational costs associated with the original D2SD algorithm. First, a simplified decomposition is applied with an a-priori evaluation of the pdf-threshold that is used to classify regions according to their concentration level, which is taken here as $\hat{\lambda} = Q_{60}$. Second, a re-meshing criteria is used to avoid applying the spatial decomposition algorithm every time step. More precisely, the re-meshing criterion is composed of two tests to check if the spatial decomposition becomes too different from the one obtained at a previous time step: one test is based on the fluctuation of the displacement of the particles within the domain $\mathcal{D}$ and another test is based on the variation of the particle concentration.

The accuracy and efficiency of these adaptations of the original D2SD algorithm have been tested on a number of situations involving a range of initial distribution of particles (homogeneous or non-homogeneous) and various particle motion representing practical situations. These test-cases allowed to set-up the thresholds (ϵ_1, ϵ_2) that are required in the quick D2SD algorithm. In particular, the results obtained support the use of a re-meshing criteria based on two indicators coupled to a quick D2SD algorithm, since it accurately reproduces the results got with the original D2SD algorithm applied every time step even in complex situations. Besides, this algorithm has been tested on a practical 3D case, where Brownian particles are injected locally within a spherical domain. The results obtained with the D2SD algorithm have been compared to the results got with fine Lagrangian simulations (based on a Langevin model). It was shown in particular that the D2SD algorithm allows to significantly improve the prediction of the number of agglomeration events compared to a simple evaluation using PBE formulation on the whole domain (which severally underestimates the number of agglomeration events). As illustrated in [15], similar results are expected (underestimating or overestimating the number of agglomeration events) when an arbitrary mesh is chosen for the collision step.

These results support the need to apply spatial decomposition techniques when resorting to mean-field approaches (such as the

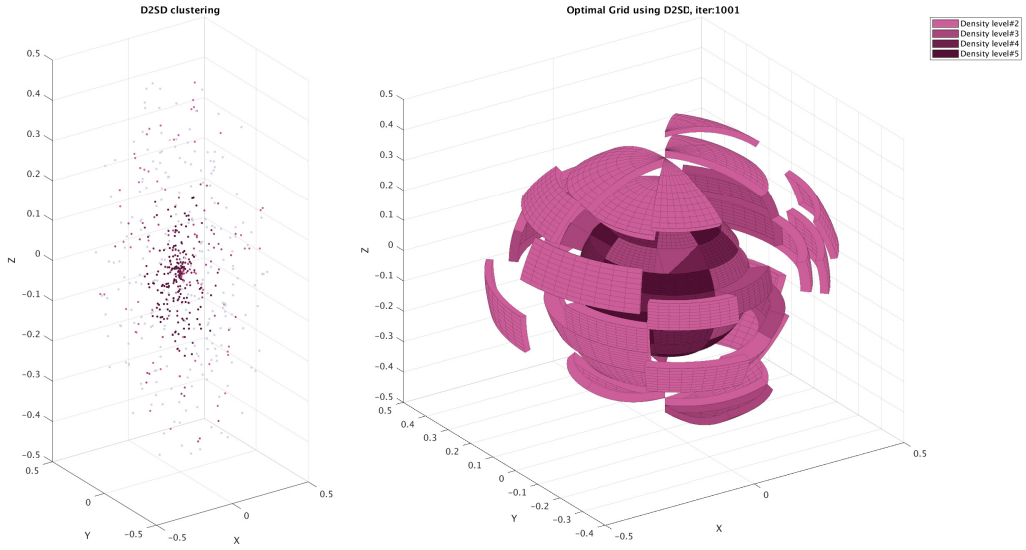


Figure 19: 3D plots showing the particles positions coloured according to their clustering region (left) and the spatial decomposition given by the quick D2SD algorithm using particle positions at a given time-iteration of the Langevin simulation (here 1001).

PBE formulation) to compute particle agglomeration in complex situations. However, these results also shed light on the limitations of such mean-field approaches: PBE formulations indeed require that the population of particles is uniformly distributed in the volume considered and that other properties are constant (e.g. fluid velocity), while neglecting any spatial/temporal correlation between collisions. Despite the apparent simplicity of PBE formulations, these simulations show that complex algorithms have to be designed in order to use such formulations in CFD simulations of complex non-homogeneous flows.

Appendix A. Collision rate of Brownian particles

Let us call $N(t)$ the number of particles present in a spherical domain at time t for a given realisation of the initial positions and of the noises. The mean number of collisions n_{coll} between times t and $t + \Delta t$ is given by the average number of pairs of particles that approach each other within a distance equal to the sum of their radii, that is $2R_p$.

Suppose that there is a reference particle i at $\mathbf{X}_i(t)$ and another one, j, at $\mathbf{X}_j$ that approaches i with a velocity $\mathbf{V}_j(t) - \mathbf{V}_i(t)$ given by Eq 13. As displayed in Fig. A.20, particle j will collide with i between t and $t + \Delta t$ if two conditions are satisfied: (i) the radial component of its relative velocity along the line connecting the two spheres is negative $W = [\mathbf{V}_j(t) - \mathbf{V}_i(t)] \cdot \mathbf{r}/|\mathbf{r}|$ (with $\mathbf{r} = \mathbf{X}_j(t) - \mathbf{X}_i(t)$); (ii) particle j is located at a distance less than $2R_p + |W|\Delta t$ from i. Assuming that particles are uniformly distributed in the sphere $[0, R_s]$, the probability that particle j collides with i given its velocity difference $W < 0$ is given by

$$\Pr(\text{coll} \mid W) = \frac{4\pi\,(2R_p)^2\,|W|\,\Delta t}{\texttt{Vol}(\mathcal{D})}. \tag{A.1}$$

with $\texttt{Vol}(\mathcal{D}) = 4\pi R_s^3/3$ the volume of the domain.

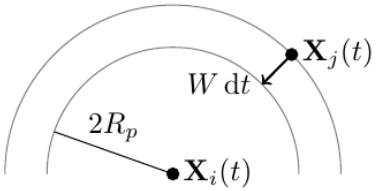


Figure A.20: 2D sketch of the criteria for a particle j to collide with a particle i fixed at the center.

The probability that particle i has a collision between t and $t + \Delta t$ can be written by summing over all particles $j \neq i$ present at time t and averaging with respect to the velocity difference W. Assuming that particle positions and velocities are completely independent, one obtains the probability $\mathcal{P}^i_{coll\in[t,t+\Delta t]}$ that particle i collides between t and $t + \Delta t$

$$\mathcal{P}^i_{coll\in[t,t+\Delta t]} = -\frac{16\,\pi\,R_p^2}{\texttt{Vol}(\mathcal{D})}\,(N(t)-1)\,\langle W\,\theta(W<0)\rangle\,\Delta t, \tag{A.2}$$

where θ denotes the Heaviside function and $\langle\cdot\rangle$ ensemble averages. For velocities following a Langevin process with friction γ and noise amplitude σ, the velocity difference W is a Gaussian process with variance σ^2/γ. One then obtains $\langle W\,\theta(W<0)\rangle = -\sigma/\sqrt{2\pi\gamma}$. As a result, the mean number of collisions $n_{coll\in[t,t+\Delta t]}$ occurring between t and $t + \Delta t$ reads

$$\begin{aligned} n_{coll\in[t,t+\Delta t]} &= \frac{1}{2}\sum_i \mathcal{P}^i_{coll\in[t,t+\Delta t]} \\ &= \frac{8\,\pi\,R_p^2\,\sigma}{\texttt{Vol}(\mathcal{D})\,\sqrt{2\pi\gamma}}\,N(t)\,(N(t)-1)\,\Delta t. \end{aligned} \tag{A.3}$$

We consider the case where each collision leads to the removal of one particle from the simulation. This leads to the following

evolution equation for the average number of particles:

$$\frac{\mathrm{d}}{\mathrm{d}t}\langle N(t)\rangle = -\frac{\beta}{\mathtt{Vol}(\mathcal{D})}\,\langle N(t)\,(N(t)-1)\rangle \tag{A.4}$$

with the collision rate (or collision frequency) β defined by

$$\beta = \frac{8\pi R_p^2\,\sigma}{\sqrt{2\pi\gamma}}. \tag{A.5}$$

Authors' contributions

The work on the algorithm was initiated by CH and the study was conceived and designed by CH, MB and KMR. KMR has carried out the adaptation of the D2SD algorithm to CFD simulations as well as testing in simple cases. KMR and CH have carried out the 3D simulations (Lagenvin system) and the coupling with the quick D2SD algorithm. KMR, MB and CH performed the data analysis. KMR, CH and MB drafted the manuscript. All authors read and approved the manuscript.

Data access

The data that support the findings of this study are available from the corresponding author on request.

Acknowledgement

We acknowledge Jérémie Bec, Martin Ferrand and Jean Pierre Minier for useful and constructive discussions on the method validation. The authors are grateful to the OPAL infrastructure from Université Côte d'Azur for providing resources and support. KMR acknowledges partial support from ANID FONDECYT POSTDOCTORADO through grant N°321011.

References

[1] A. D. McNaught, A. Wilkinson, IUPAC: Compendium of chemical terminology, 2nd ed. (The "Gold Book"), online version (2019-) created by S. J. Chalk. Edition, Blackwell Scientific Publications, Oxford, 1997. doi:https://doi.org/10.1351/goldbook.

[2] M. Elimelech, J. Gregory, X. Jia, Particle deposition and aggregation: measurement, modelling and simulation, Butterworth-Heinemann, 2013. doi:https://doi.org/10.1016/B978-0-7506-7024-1.X5000-6.

[3] R. H. Meade, Transport and deposition of sediments in estuaries, Geological Society of America 133 (1) (1972) 91–120. doi:https://doi.org/10.1130/MEM133-p91.

[4] K. Gotoh, Y. Fujii, A fractal dimensional analysis on the cloud shape parameters of cumulus over land, Journal of Applied Meteorology 37 (10) (1998) 1283–1292. doi:https://doi.org/10.1175/1520-0450(1998)037<1283:AFDAOT>2.0.CO;2.

[5] G. Falkovich, A. Fouxon, M. Stepanov, Acceleration of rain initiation by cloud turbulence, Nature 419 (6903) (2002) 151. doi:https://doi.org/10.1038/nature00983.

[6] R. A. Shaw, Particle-turbulence interactions in atmospheric clouds, Annual Review of Fluid Mechanics 35 (1) (2003) 183–227. doi:https://doi.org/10.1146/annurev.fluid.35.101101.161125.

[7] J. Blum, G. Wurm, The growth mechanisms of macroscopic bodies in protoplanetary disks, Annu. Rev. Astron. Astrophys. 46 (2008) 21–56. doi:https://doi.org/10.1146/annurev.astro.46.060407.145152.

[8] M.-F. Pouet, A. Grasmick, Urban wastewater treatment by electrocoagulation and flotation, Water Science and Technology 31 (3-4) (1995) 275–283. doi:https://doi.org/10.1016/0273-1223(95)00230-K.

[9] J. Rubio, M. Souza, R. Smith, Overview of flotation as a wastewater treatment technique, Minerals Engineering 15 (3) (2002) 139–155. doi:https://doi.org/10.1016/S0892-6875(01)00216-3.

[10] M. Bartels, W. Lin, J. Nijenhuis, F. Kapteijn, J. R. Van Ommen, Agglomeration in fluidized beds at high temperatures: Mechanisms, detection and prevention, Progress in Energy and Combustion Science 34 (5) (2008) 633–666. doi:https://doi.org/10.1016/j.pecs.2008.04.002.

[11] L. Gallen, A. Felden, E. Riber, B. Cuenot, Lagrangian tracking of soot particles in LES of gas turbines, Proceedings of the Combustion Institute 37 (4) (2019) 5429–5436. doi:https://doi.org/10.1016/j.proci.2018.06.013.

[12] N. Maximova, O. Dahl, Environmental implications of aggregation phenomena: current understanding, Current Opinion in Colloid & Interface Science 11 (4) (2006) 246–266. doi:https://doi.org/10.1016/j.cocis.2006.06.001.

[13] D. Henning, R. Baer, A. Hassan, R. Dave, Major advances in concentrated and dry milk products, cheese, and milk fat-based spreads, Journal of Dairy Science 89 (4) (2006) 1179–1188. doi:https://doi.org/10.3168/jds.S0022-0302(06)72187-7.

[14] C. Henry, J.-P. Minier, G. Lefèvre, Towards a description of particulate fouling: From single particle deposition to clogging, Advances in colloid and interface science 185 (2012) 34–76. doi:https://doi.org/10.1016/j.cis.2012.10.001.

[15] K. M. Rodríguez, M. Bossy, R. Maftei, S. Shekarforush, C. Henry, New spatial decomposition method for accurate, mesh-independent agglomeration predictions in particle-laden flows, Applied Mathematical Modelling 90 (2020) 582–614. doi:https://doi.org/10.1016/j.apm.2020.08.064.

[16] M. Chen, K. Kontomaris, J. McLaughlin, Direct numerical simulation of droplet collisions in a turbulent channel flow. Part I: collision algorithm, International Journal of Multiphase Flow 24 (7) (1999) 1079–1103. doi:https://doi.org/10.1016/S0301-9322(98)00007-X.

[17] H. Sigurgeirsson, A. Stuart, W.-L. Wan, Algorithms for particle-field simulations with collisions, Journal of Computational Physics 172 (2) (2001) 766–807. doi:https://doi.org/10.1006/jcph.2001.6858.

[18] J. Bec, S. S. Ray, E. W. Saw, H. Homann, Abrupt growth of large aggregates by correlated coalescences in turbulent flow, Physical Review E 93 (3) (2016) 031102. doi:https://doi.org/10.1103/PhysRevE.93.031102.

[19] J.-P. Minier, E. Peirano, The pdf approach to turbulent polydispersed two-phase flows, Physics Reports 352 (1-3) (2001) 1–214. doi:https://doi.org/10.1016/S0370-1573(01)00011-4.

[20] J.-P. Minier, On Lagrangian stochastic methods for turbulent polydisperse two-phase reactive flows, Progress in Energy and Combustion Science 50 (2015) 1–62. doi:https://doi.org/10.1016/j.pecs.2015.02.003.

[21] J.-P. Minier, Statistical descriptions of polydisperse turbulent two-phase flows, Physics Reports 665 (2016) 1–122. doi:https://doi.org/10.1016/j.physrep.2016.10.007.

[22] S. Subramaniam, Lagrangian–Eulerian methods for multiphase flows, Progress in Energy and Combustion Science 39 (2-3) (2013) 215–245. doi:https://doi.org/10.1016/j.pecs.2012.10.003.

[23] C. Henry, J.-P. Minier, M. Mohaupt, C. Profeta, J. Pozorski, A. Tanière, A stochastic approach for the simulation of collisions between colloidal particles at large time steps, International Journal of Multiphase Flow 61 (2014) 94–107. doi:https://doi.org/10.1016/j.ijmultiphaseflow.2014.01.007.

[24] C. Henry, J.-P. Minier, J. Pozorski, G. Lefèvre, A new stochastic approach for the simulation of agglomeration between colloidal particles, Langmuir 29 (45) (2013) 13694–13707. doi:https://doi.org/10.1021/la403615w.

[25] P. J. O'Rourke, Collective drop effects on vaporizing liquid sprays, Tech. rep., Los Alamos National Lab., NM (USA) (1981).

[26] M. Mezhericher, A. Levy, I. Borde, Probabilistic hard-sphere model of binary particle–particle interactions in multiphase flow of spray dryers, International Journal of Multiphase Flow 43 (2012) 22–38. doi:https://doi.org/10.1016/j.ijmultiphaseflow.2012.02.009.

[27] D. P. Schmidt, C. Rutland, A new droplet collision algorithm, Journal of

Computational Physics 164 (1) (2000) 62–80. doi:https://doi.org/10.1006/jcph.2000.6568.

[28] G. Bird, Approach to translational equilibrium in a rigid sphere gas, Physics of Fluids 6 (1963) 1518–1519. doi:https://doi.org/10.1063/1.1710976.

[29] G. A. Bird, J. Brady, Molecular gas dynamics and the direct simulation of gas flows, Vol. 42, Clarendon press Oxford, 1994.

[30] F. J. Alexander, A. L. Garcia, The direct simulation Monte Carlo method, Computers in Physics 11 (6) (1997) 588–593. doi:https://doi.org/10.1063/1.168619.

[31] C. Henry, K. K. Norrfors, M. Olejnik, M. Bouby, J. Luetzenkirchen, S. Wold, J.-P. Minier, A refined algorithm to simulate latex colloid agglomeration at high ionic strength, Adsorption 22 (4-6) (2016) 503–515. doi:https://doi.org/10.1007/s10450-015-9714-4.

[32] D. Ramkrishna, The status of population balances, Reviews in Chemical Engineering 3 (1) (1985) 49–95. doi:https://doi.org/10.1515/REVCE.1985.3.1.49.

[33] D. Ramkrishna, Population balances: Theory and applications to particulate systems in engineering, Elsevier, 2000.

[34] D. L. Marchisio, R. O. Fox, Solution of population balance equations using the direct quadrature method of moments, Journal of Aerosol Science 36 (1) (2005) 43–73. doi:https://doi.org/10.1016/j.jaerosci.2004.07.009.

[35] D. Ramkrishna, M. R. Singh, Population balance modeling: current status and future prospects, Annual Review of Chemical and Biomolecular Engineering 5 (2014) 123–146. doi:https://doi.org/10.1146/annurev-chembioeng-060713-040241.

[36] G. Kasper, On the coagulation rate of aerosols with spatially inhomogeneous particle concentrations, Journal of Colloid and Interface Science 102 (2) (1984) 560–562. doi:https://doi.org/10.1016/0021-9797(84)90261-3.

[37] D. P. Schmidt, C. J. Rutland, Reducing grid dependency in droplet collision modeling, J. Eng. Gas Turbines Power 126 (2) (2004) 227–233. doi:https://doi.org/10.1115/1.1564066.

[38] S. Hou, D. P. Schmidt, Adaptive collision meshing and satellite droplet formation in spray simulations, International Journal of Multiphase Flow 32 (8) (2006) 935–956. doi:https://doi.org/10.1016/j.ijmultiphaseflow.2006.02.013.

[39] P. Pischke, D. Cordes, R. Kneer, A collision algorithm for anisotropic disperse flows based on ellipsoidal parcel representations, International Journal of Multiphase Flow 38 (1) (2012) 1–16. doi:https://doi.org/10.1016/j.ijmultiphaseflow.2011.09.002.

[40] J. Zhang, J. Mi, H. Wang, A new mesh-independent model for droplet/particle collision, Aerosol Science and Technology 46 (6) (2012) 622–630. doi:https://doi.org/10.1080/02786826.2011.649809.

[41] P. Pischke, R. Kneer, D. P. Schmidt, A comparative validation of concepts for collision algorithms for stochastic particle tracking, Computers & Fluids 113 (2015) 77–86. doi:https://doi.org/10.1016/j.compfluid.2015.01.018.

[42] D. Freedman, P. Diaconis, On the histogram as a density estimator: L2 theory, Probability Theory and Related Fields 57 (4) (1981) 453–476. doi:https://doi.org/10.1007/BF01025868.

[43] J. R. Berrendero, A. Cuevas, F. Vázquez-grande, Testing multivariate uniformity: The distance-to-boundary method, Canadian Journal of Statistics 34 (4) (2006) 693–707. doi:https://doi.org/10.1002/cjs.5550340409.