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

Preprint ISSN 2198-5855


# Simulations of an ASA flow crystallizer with a coupled stochastic-deterministic approach


Clemens Bartsch[1], Volker John[1,2], Robert I. A. Patterson[1]



submitted: February 15, 2018



[1] Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: clemens.bartsch@wias-berlin.de
volker.john@wias-berlin.de
robert.patterson@wias-berlin.de

[2] Free University of Berlin
Department of Mathematics and Computer Science
Arnimallee 6
14195 Berlin, Germany



No. 2483
Berlin 2018



2010 *Mathematics Subject Classification.* 76T20.

*Key words and phrases.* Tubular flow crystallizer, population balance systems, stochastic particle method, Markov jump processes, deterministic CFD methods.

The work of V. John was supported by grant Jo329/10-3 within the DFG priority programme 1679: Dynamic simulation of interconnected solid processes.

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

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

# Simulations of an ASA flow crystallizer with a coupled stochastic-deterministic approach

Clemens Bartsch,Volker John, Robert I. A. Patterson

**Abstract**

A coupled solver for population balance systems is presented, where the flow, temperature, and concentration equations are solved with finite element methods, and the particle size distribution is simulated with a stochastic simulation algorithm, a so-called kinetic Monte-Carlo method. This novel approach is applied for the simulation of an axisymmetric model of a tubular flow crystallizer. The numerical results are compared with experimental data.

## 1 Introduction

A flow crystallizer consists mainly of a long, thin tube that is usually coiled up for practical reasons. A three-component dispersion is pumped into the tube at the inlet. The dispersion contains a solvent, a solute, and seed crystals, which consist of the same material as the solute. It is warm and highly saturated. While the dispersion flows through the tube, mechanisms that result in a change of crystal size are excited, like nucleation, surface attachment growth, particle collision growth or particle breakage. The occurrence and intensity of these processes are controlled by several parameters. Among these are the fluid composition, the fluid velocity, the shape and makeup of the tube, and the surrounding temperature. At the outlet, product crystals with properties different from the input crystals can be collected.

Flow crystallizers are a promising technology in pharmaceutical production. They allow for a regular particle growth that can be accurately controlled. Regular shape and size are desirable properties of crystals that are used in medicines, as more accurate crystal morphology influences medicinal effects. From a technical point of view, flow crystallizers are interesting for two more reasons. Firstly, they can be operated continuously. Secondly, knowledge transfer from the laboratory to the industry is relatively easy. Wherever fluids are involved, there are big differences between different scales, and so a typical development issue would be that a system behaves totally different at small scale (laboratory) and big scale (industry). This problem of scale-up does not apply to flow crystallizers: scaling up to the industrial measure does not mean bigger or longer tubes, it means more tubes, operated simultaneously. Thus, the conditions in the individual tube stay the same in experiment and application.

In a flow crystallizer, surface growth is largely responsible for the size gain of the crystals. A flow crystallizer is operated often at laminar conditions, the flow field trajectories are aligned and particle collisions happen less frequently, surface growth dominates. This effect is often desired, and it is boosted by cooling the flow crystallizer. As the temperature drops, so does the solubility of the material of the crystals. Supersaturation of the surrounding solution increases, and the need to reduce supersaturation is the driving force behind surface attachment growth. One must confine the above statement to tube crystallizers operated at low velocity and smooth curvature. As soon as either veloc-

DOI 10.20347/WIAS.PREPRINT.2483 Berlin 2018

ity or curvature hit a critical number, secondary flow structures (Dean vortices) begin to develop and particle coagulation becomes more likely.

Collision growth can be an obstacle to the operation of a tube crystallizer, as can “excessive primary nucleation” [10, p. 2249]. Both can easily lead to blockage of the typically very thin tube. A second issue is the goal to achieve a sharp size distribution of the product crystals. Collision growth runs contrary to this aim, because each collision event results in a sudden jump of the size of the involved crystals. Surface growth is much smoother in that sense, and therefore it is the preferable growth mechanism.

Numerical simulations have become an indispensable tool for the prediction and control of many processes in chemical engineering. The main purpose of this paper consists in presenting a novel coupled approach which solves equations for the flow, temperature, and concentration in a deterministic way, and the equation for the crystals with a stochastic method. Since in this paper a proof of concept for this approach is provided, the considered application was chosen such that it was possible to concentrate on the main new aspect of the method: the coupling of the deterministic CFD part and the stochastic particle solver.

The flow crystallizer that is in the focus of this paper was set up and operated by the group of Prof. Khinast at TU Graz. The crystalline model substance was acetylsalicylic acid (ASA), commonly known by its brand name aspirin. The solvent was almost pure ethanol (EtOH). Results were first reported in [10]. Modified setups were presented later in [9] and [11]. The first of these works contains a 1d ODE model of the experiment, and computational results that were gained with it. The work [5] contains a model and simulations for the setup in [11]. Recently, the method was applied to crystallization growth of the enzyme lysozyme in [31]. In [10], the authors found four different parameter setups for which they could operate the crystallizer up to fifteen minutes without blockage, maintaining almost steady-state conditions at the outlet.

The crystallization process in the flow crystallizer is modeled with a population balance system (PBS) consisting of equations for the flow, the temperature, the dissolved ASA, and a population balance equation (PBE) for the crystalline ASA. In this paper, the situation of a long straight tube is considered, which enables the derivation and use of a 2d axisymmetric model. For the sake of concentration to the main novel aspect of the method, only one internal coordinate (property coordinate) for the particle population is considered, although the stochastic particle method is designed for the simulation of complex particles. The restriction to one internal coordinate has the further advantage that the example can serve for comparison with other numerical methods, which is one of our future plans.

There is a rich literature on numerical methods for PBSs. A main difficulty comes from the fact that the PBE is defined in a higher dimensional domain than the other equations because of the internal coordinates. In the case of one and even two internal coordinates, direct methods for the higher dimensional equations have been applied, e.g., in [7, 22]. A popular alternative are moment-based schemes like the quadrature method of moment (QMOM) and variants of this method [28, 30].

A main limitation of the approaches mentioned so far is, in our opinion, that they assume from the modeling point of view that the aggregate of two particles is a particle of the same type whose properties are the sum of the properties of the two individual particles. Thus, often spherical particles are assumed and one internal coordinate is mass (or volume). However, models with two and more additive internal coordinates are quite rare in the literature.

In this paper, an approach is used for the simulation of the particle behavior that is suited inherently for complex particles, namely a stochastic simulation algorithm (SSA). SSAs in their many recent forms are the result of more than forty years of combined research effort. They have been improved and

Table 1: List of abbreviations.

| | | |
|---|---|---|
| ASA | – | acetylsalicylic acid |
| CFD | – | computational fluid dynamics |
| PBE | – | population balance equation |
| PBS | – | population balance system |
| PSD | – | particle size distribution |
| SPS | – | stochastic particle simulation |
| SSA | – | stochastic simulation algorithm |

extended by different research groups on different occasions, and with different scopes of application in mind. They have been developed further within different communities, and different names were coined for them, among them “Kinetic Monte Carlo”, “Dynamic Monte Carlo”, “Direct Simulation Monte Carlo”, “Population Balance Monte Carlo”, or “Gillespie SSA”. The last of these names gives credit to [17], where an SSA was introduced as a tool for the simulation of reactive chemical systems. The simulation algorithm used in this paper is closely based on that presented in [35] and various refinements are available for particular steps, see for example [24, 33]. It is based on a splitting between growth/coagulation and transport processes analogous to the splitting between elastic collisions and free streaming in the Bird algorithm for the Boltzmann equation [6]. The crystallizer is partitioned into cells and the physical particle population within each cell is approximated by an ensemble of computational particles. This approximation is ultimately justified by mathematical convergence proofs [34], but heuristically one may identify the computational particles with the physical particles in a small fraction of each cell in the spatial partition. In particular, the simulation dynamics can be expressed almost directly as a physical model for the behavior of individual particles. This particle approach lies behind the power and flexibility of the algorithm. Because one only ever has to deal with pairs of individual particles, high-dimensional integrals that arise when particles have many internal degrees of freedom can be avoided and reduced to sums over a list of particles. The SSA has therefore been used for very sophisticated particle models [38, 42] as well as in less complex settings [12, 18, 23].

The paper is organized as follows. In Section 2, the axisymmetric model of the tubular flow crystallizer and experimental results are described. The stochastic model of the growth processes is explained in Section 3. Section 4 describes the individual components of the coupled numerical method and the coupling strategy. Numerical studies are presented in Section 5. Finally, Section 6 provides a summary and gives an outlook.

## 2 The Model of an ASA Crystallizer

In this section, we will present a model of a flow crystallizer for aspirin. It is two-dimensional in space, a simplification which is achieved by assuming axisymmetry. First, we will describe the experiment to be modeled and simulated. Then, we list several general modeling assumptions, which include considerations on the fluid density, the axisymmetric geometry, and the particle model. In the following, we are going to use the terms *particles* and *crystals* interchangeably. The first term (“particles”) is closer to the mathematical model, the second term (“crystals”) more related to the actual physical system.

## 2.1 The Experiment

In the experimental setup, as described in [10], the crystallization takes place in a $15\ \mathrm{m}$ long polysiloxane tube that is coiled up in a box of dimensions $0.41\ \mathrm{m} \times 0.24\ \mathrm{m} \times 0.26\ \mathrm{m}$. The inner diameter of the tube is $2\ \mathrm{mm}$ and the outer diameter is $4\ \mathrm{mm}$, i.e., the tube has a wall thickness of $1\ \mathrm{mm}$. The fluid in the crystallizer is a mixture that is fed from two vessels. The first vessel contains a warm solution of ASA in ethanol, close to supersaturation. The second one contains a well-mixed ASA seed crystal suspension, consisting of ethanol, dissolved ASA and undissolved ASA in crystallized form. One peristaltic pump per vessel pumps the contents into a Y-fitting ("mixing zone"). and from there the mixture flows into the tube crystallizer. The temperature in the box that contains the tube is held at $24.3 \pm 1\ ^{\circ}\mathrm{C}$, which is cool compared to the temperature of the fluid at the inlet, see Table 2. The cooling results in a drop of supersaturation and thus in crystal surface growth. At inlet and outlet of the crystallizer, a microscope and a specialized camera have been installed, which allowed to gather data on the in- and outflowing crystals. An approximation of the crystal size distribution at in- and outflow could be determined from this data. In [10], four configurations were figured out to steadily operate the flow crystallizer, meeting several requirements at once:

- Comparing inflowing and outflowing crystals, the peak of the crystal size distribution should move significantly towards larger particles, i.e., significant surface growth should take place within the device.
- The peak of the product crystal size distribution should be sharp, i.e., the particles at the outlet should be rather evenly sized.
- The tube should not be blocked due to excessive crystal growth.
- Finally, as much crystalline material as possible, given the other requirements, should be produced within a certain operating time.

It is our goal to computationally reproduce the experimental results for all four parameter sets. The target variables of the simulation are the mean $\bar{d}$ and standard deviation $\sigma$ of the particle diameter at the outlet. There are several constraints that we will also use for the validation and verification of our computational results. These are:

- Mean and standard deviation of the crystal diameter at the inlet.
- Mass flow rate of crystalline ASA at the outlet.
- The conservation of the mass flux throughout the tube.

These target quantities of the experiments are also listed in Table 2. We use superscript $[i]$ with $i \in \{1, ..., 4\}$ for the distinction of the data. Often, when some general statement holds for all four setups, we skip the superscript.

## 2.2 Some Modeling Assumptions

The particles represent ASA crystals of different sizes. We regard the entire computational domain to be continuously filled with homogeneous matter. Its suspension character and its microscopic features are ignored at first. The particles then get re-introduced as zero-dimensional objects, which interact

Table 2: *Top*: Target data at the inlet. Crystal mass flow rate $Q_{\text{in}}$, crystal mass flux $\dot{m}_{\text{in}}$, particle number mean diameter $\bar{d}_{\text{in}}$, particle diameter standard deviation $\sigma_{\text{in}}$, suspension temperature $T_{\text{in}}$. *Bottom*: Target data at the outlet. First number refers to the end time of the experiment (15, 11, 9, 9 minutes), number in parentheses to an intermediate time (9, 7, 6, 5 minutes).

| Quantity | $Q_{\text{in}}$ $[\mathrm{m}^3/\mathrm{s}]$ | $\dot{m}_{\text{in}}$ $[\mathrm{kg}/\mathrm{s}]$ | $\bar{d}_{\text{in}}$ $[\mu\mathrm{m}]$ | $\sigma_{\text{in}}$ $[\mu\mathrm{m}]$ | $T_{\text{in}}$ $[\mathrm{K}]$ |
|---|---|---|---|---|---|
| Setup 1 | 1.9e-7 | 0.0156 | 90 | 38 | 307.6 |
| Setup 2 | 2.9e-7 | 0.0258 | 81 | 26 | 312.9 |
| Setup 3 | 3.8e-7 | 0.0324 | 91 | 27 | 313.1 |
| Setup 4 | 4.2e-7 | 0.0378 | 85 | 29 | 313.7 |

| Quantity | $\dot{m}_{\text{out}}$ $[\mathrm{kg}/\mathrm{s}]$ | $\bar{d}_{\text{out}}$ $[\mu\mathrm{m}]$ | $\sigma_{\text{out}}$ $[\mu\mathrm{m}]$ | $T_{\text{out}}$ $[\mathrm{K}]$ |
|---|---|---|---|---|
| Setup 1 | 0.0888 | 243 (233) | 65 (62) | 297.5 |
| Setup 2 | 0.1512 | 214 (215) | 50 (43) | 297.5 |
| Setup 3 | 0.1932 | 192 (183) | 49 (41) | 297.5 |
| Setup 4 | 0.195 | 166 (186) | 44 (45) | 297.5 |

with the fluid only via a macroscopic observable, their particle size distribution (PSD). This approach comprises several assumptions on the particles: from the fluid point of view, the particles have a position but no extension. They are described by the PSD, which is a cumulated quantity. Additionally, the particles are assumed to follow the streamlines of the macroscopic velocity field and do not back-couple on the velocity field. As for the internal coordinates, we use a one-dimensional model: particle mass $m$ is the internal coordinate.

In order to stay in the framework of incompressible fluids with non-varying density, the density $\rho_{\text{susp}} = 916.87\,\mathrm{kg}/\mathrm{m}^3$ of the fluid is kept constant, see [4] for a detailed derivation of this value. The assumption of constant density means that surface growth, i.e., phase transition of ASA from dissolved to solid form, does not influence the density. Both phases are assumed to contribute to the overall density in the same way.

The computations will comprise only the $15\,\mathrm{m}$ long main portion of the tube, excluding the vessels, the Y-fitting and the mixing zone of the device. Also, the computational domain spans only the inner part of the tube, that part where the fluid flows. The tube wall is not part of the computational domain. This approach requires some assumptions on the boundary conditions for the energy balance equation, see below in this section. The geometry of the tube interior is simplified. The originally coiled up tube is straightened out, and an axisymmetric 2d approach is pursued. The meridian modeling domain is depicted in Figure 1. Here, $\Lambda$ is the 2d coupling domain, while $\Omega$ will denote the three-dimensional cylinder which is gained by rotation of $\Lambda$. The boundaries of $\Lambda$ are in- and outflow boundary $\Gamma_{\text{in}}$ and $\Gamma_{\text{out}}$. The wall boundary is $\Gamma_{\text{wall}}$ and $\Gamma_{\text{symm}}$ is the symmetry boundary in the center of the tube. As for the cylindrical variables, $z \in [0, 15]$ is the axial variable and $r \in [0, 0.001]$ is the radial variable, compare Figure 1.

## 2.3 The Population Balance System

The Reynolds and Dean numbers of the flows are given in Table 3, for their derivation the reader is referred to [4]. All numbers are small enough to expect a laminar flow without secondary vortices. A laminar flow in a tube or pipe develops a parabolic velocity profile. For this well-understood case, the analytic solution to the Navier–Stokes equations is given by the law of Hagen–Poiseuille. Note that in

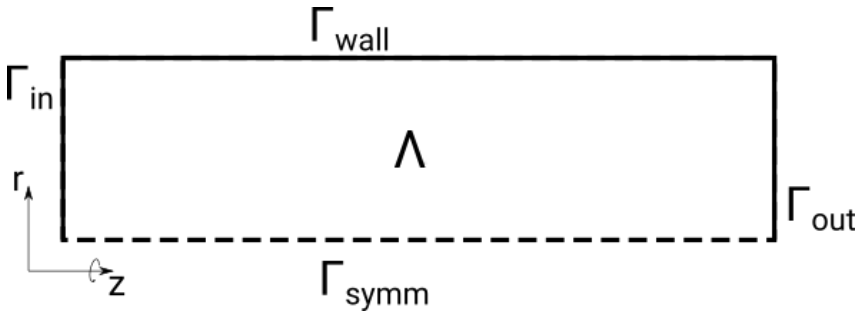



Figure 1: The modeling domain (symbolic), with denotations of the boundary pieces and the coordinates. The dashed line is the symmetry axis.

Table 3: Key figures of the flow, for the four different parameter setups. Total mass flow was calculated based on volumetric flow rate data in [10] and the density $\rho_{\text{susp}} = 916.87\ {}^{\text{kg}}/_{\text{m}^3}$.

| Quantity | Total mass flow $\left[{}^{\text{kg}}/_{\text{s}}\right]$ | Mean vel. $\left[{}^{\text{m}}/_{\text{s}}\right]$ | Reynolds no. | Dean no. |
|---|---|---|---|---|
| Setup 1 | 0.000174 | 0.0605 | 11.1 | 1 |
| Setup 2 | 0.000263 | 0.0912 | 16.7 | 1.5 |
| Setup 3 | 0.000348 | 0.1209 | 22.2 | 2 |
| Setup 4 | 0.000385 | 0.1337 | 24.5 | 2.2 |

an axisymmetric setup, this leads to a vanishing radial component of the flow, $u_r = 0$. Backcoupling from the particles, concentration or temperature is not contained in the flow model. The same approach was pursued, e.g., in [2, 19].

The population balance equation (PBE) for the one-dimensional particle model with internal coordinate $m\ [\text{kg}]$ has the form

$$\frac{\partial}{\partial t} f + \boldsymbol{u} \cdot \nabla f = \mathcal{C}(f) + \mathcal{G}(c, T, f) \quad \text{in } (0, t_{\text{end}}) \times \Omega \times [0, \infty), \tag{1}$$

where the sought function $f\ \left[{}^{1}/_{\text{m}^3\text{kg}}\right]$ is the particle number density. In (1), $\boldsymbol{u}\ \left[{}^{\text{m}}/_{\text{s}}\right]$ denotes the velocity field, $\mathcal{C}(f)$ the coagulation term, and $\mathcal{G}(c, T, f)$ the growth term. A stochastic particle scheme will be utilized for computing the right-hand side of (1). The corresponding Markov jump process formulation will be described in Section 4.3. The end time $t_{\text{end}}$ depends on the experimental setup.

The way how particles entering the flow domain are inserted into the simulation is described in detail in Section 4.3.

It is not necessary to formulate wall and symmetry axis boundary conditions at $\Gamma_{\text{wall}}$ and $\Gamma_{\text{symm}}$, since $u_r = 0$ means that no wall or axis collisions happen. At the outflow boundary, particles are removed from the computation when the transport step transports them to a point beyond $\Gamma_{\text{out}}$.

The coagulation term has the form

$$\begin{aligned} \mathcal{C}(f, t, \boldsymbol{x}, m) \;=\; & \frac{1}{2} \int_{\Omega_m} K(m-\mu, \mu) f(t, \boldsymbol{x}, m-\mu) f(t, \boldsymbol{x}, \mu)\, \mathrm{d}\mu \\ & - \int_{\Omega_m} K(m, \mu) f(t, \boldsymbol{x}, m) f(t, \boldsymbol{x}, \mu)\, \mathrm{d}\mu, \end{aligned}$$

where $\boldsymbol{x}\ [\text{m}]$ is the spatial coordinate and $m\ [\text{kg}]$ is the internal coordinate. There are several options for defining the coagulation kernel $K$. In our simulations, two "traditional" kernels were applied. The

first one is the additive shear flow kernel, which is usually suitable for the laminar flow regime, see [3]. In terms of particle mass, this coagulation kernel gives the probability that two "near" particles of masses $m_1$ and $m_2$ coagulate in an infinitesimal time interval. It has the form

$$K(m_1, m_2) = \kappa^{\text{add}}(m_1 + m_2) \quad [1/\text{s}]. \tag{2}$$

The scaling parameter $\kappa^{\text{add}}$ is unspecified so far. A second kernel is the (simpler) constant coagulation kernel

$$K(m_1, m_2) = \kappa^{\text{const}} \quad [1/\text{s}]. \tag{3}$$

The used stochastic particle method requires to model the growth rate in terms of particle surface reaction rather than as convection along the internal coordinate. This modeling and a comparison of both approaches are described in Section 3. Given a fixed mass growth increment $\mu$ [kg], the following form of the growth term

$$\mathcal{G}(c, T, f, m) = \frac{G(c, T, m-\mu)}{\mu} f(m-\mu) - \frac{G(c, T, m)}{\mu} f(m)$$

is used, which is well suited to be treated with a stochastic particle algorithm. For $G(\cdot)$, we stick to the semi-empirical model used in [5, 27]. It is based on the Arrhenius equation for the reaction speed constant of temperature dependent reactions, but multiplied with a monomial supersaturation term. In [27], the model is used for a diameter-based 1d particle description:

$$G_d(c, T) = k_{G_1} \exp\left(-\frac{k_{G_2}}{RT}\right) (c_{\text{sat}}(T) - c)^{k_{G_3}} \quad [\text{m}/\text{s}]. \tag{4}$$

It is formulated in terms of absolute supersaturation. In (4), $T$ [K] is the temperature, $c$ [mol/m$^3$] is the concentration, $c_{\text{sat}}$ [mol/m$^3$] the supersaturation, and $R = 8.314$ J/kg·K is the ideal gas constant. The parameters $k_G$ were experimentally determined, [27], to be

$$k_{G_1} = 3.21 \cdot 10^{-4} \text{ m/s}, \quad k_{G_2} = 2.58 \cdot 10^{-4} \text{ J/mol}, \quad k_{G_3} = 1.$$

For our mass-based approach we have to reformulate $G_d$ to $G_m$ [kg/s], the mass growth speed. Because the model assumes spherical particles, there is the following dependence of particle mass on particle diameter: $m = \frac{\pi}{6} d^3 \rho_{\text{ASA}}$. Using the chain rule and $G_d = \frac{\mathrm{d}}{\mathrm{d}t} d$, one computes

$$G_m(c, T, m) = \frac{\mathrm{d}}{\mathrm{d}t} m = \frac{\mathrm{d}}{\mathrm{d}d} m \frac{\mathrm{d}}{\mathrm{d}t} d = \frac{\pi}{2} d^2 \rho_{\text{ASA}} G_d(c, T).$$

Setting $A_O(m) = \pi d(m)^2$, the surface area of a spherical ASA particle of mass $m$, one obtains

$$G_m(c, T, m) = \frac{1}{2} A_O(m) \rho_{\text{ASA}} G_d(c, T). \tag{5}$$

This particle mass growth rate depends linearly on the particle surface area. Another modeling decision concerns the supersaturation $c_{\text{sat}}(T)$. We use here a fitted Nyvilt model, see [10],

$$c_{\text{sat}}(T) = 10^{27.769 + \frac{-2500.906}{T} - 8.323 \log_{10}(T)}.$$

The concentration balance equation is modeled with a convection-diffusion equation. Its formulation in Cartesian coordinates is

$$\frac{\partial c}{\partial t} - D\Delta c + \boldsymbol{u} \cdot \nabla c = -\frac{1}{M_{\text{ASA}}} I_{\text{growth}}(c, T, f) \quad \text{on } \Omega \times [0, t_{\text{end}}). \tag{6}$$

Table 4: Material constants of ASA and ethanol (at $20\ ^\circ\mathrm{C}$) used in this paper.

| Quantity | $M_{\mathsf{ASA}}\ [\mathrm{kg/mol}]$ | $M_{\mathsf{EtOH}}\ [\mathrm{kg/mol}]$ | $\rho_{\mathsf{ASA}}\ [\mathrm{kg/m^3}]$ | $\rho_{\mathsf{EtOH}}\ [\mathrm{kg/m^3}]$ |
|---|---|---|---|---|
| Value | 0.18016 | 0.04607 | 1350 | 790 |

The unknown function $c\ [\mathrm{mol/m^3}]$ describes the molar concentration of dissolved ASA. The diffusion coefficient $D\ [\mathrm{m^2/s}]$ is the diffusion coefficient of dissolved ASA in EtOH. As we are not aware of an exact value in the literature, we use the diffusion coefficient of another model substance, urea in ethanol, see [2]. Since any numerical stabilization for convection-diffusion equations introduces numerical diffusion which exceeds the physical diffusion, it is only important to choose $D$ such that its order of magnitude is reasonable. We set $D = 1.35 \cdot 10^{-9}\ \mathrm{m^2/s}$. The precomputed velocity is $\boldsymbol{u}\ [\mathrm{m/s}]$, and $M_{\mathsf{ASA}}\ [\mathrm{kg/mol}]$ is the molar mass of ASA as given in Table 4.

On the right-hand side, the term $I_{\mathsf{growth}}\ [\mathrm{kg/m^3 s}]$ is the surface growth intensity of the ASA crystals, which measures the intensity of particle surface growth. Using the model (4) of the growth term, the growth intensity term takes the form

$$\begin{aligned} I_{\mathsf{growth}}(c,T,f,t,\boldsymbol{x}) &= \int_{\Omega_m} G(c,T,m) f(t,\boldsymbol{x},m)\,\mathrm{d}m \\ &= \frac{\rho_{\mathsf{ASA}}}{2} \int_{[0,\infty)} A_O(m) k_{G_1} \exp\left(-\frac{k_{G_2}}{RT}\right) \frac{(c_{\mathsf{sat}}(T)-c)}{c} f(t,\boldsymbol{x},m)\,\mathrm{d}m. \end{aligned} \tag{7}$$

The boundary conditions for the axisymmetric re-formulation of (7) are

$$\begin{cases} c = 1511.3\ \mathrm{mol/m^3} & \text{on } \Gamma_{\mathsf{in}}, \\ \frac{\partial c}{\partial n_\Gamma} = 0 & \text{on } \Gamma_{\mathsf{wall}} \cup \Gamma_{\mathsf{out}} \cup \Gamma_{\mathsf{symm}}. \end{cases}$$

Interestingly, the boundary conditions at outflow, wall, and symmetry boundary are all the same, although their interpretation differs. At $\Gamma_{\mathsf{wall}}$, the condition describes impermeability, at the outflow $\Gamma_{\mathsf{out}}$, it is a natural outflow condition, and at the symmetry axis $\Gamma_{\mathsf{symm}}$, it is the necessary symmetry condition. The derivation of the Dirichlet value at the inlet $\Gamma_{\mathsf{in}}$ from the experimental data is described in detail in [4].

Finally, the equation is closed with an initial condition. We assume that there is no ASA present in the crystallizer before the experiment starts, thus $c(0,\cdot) = 0\ \mathrm{mol/m^3}$ in $\Omega$.

The energy balance is modeled by the convection-diffusion equation formulation

$$\frac{\partial T}{\partial t} + \boldsymbol{u}\cdot\nabla T - \frac{\lambda_{\mathsf{EtOH}}}{\rho_{\mathsf{susp}} C_{\mathsf{EtOH}}} = \frac{\Delta h_{\mathsf{cryst}}}{\rho_{\mathsf{susp}} C_{\mathsf{EtOH}}} I_{\mathsf{growth}}(c,T,f) \text{ on } \Omega \times [0, t_{\mathsf{end}}). \tag{8}$$

The sought quantity $T\ [\mathrm{K}]$ is the temperature. On the right-hand side appears again the growth intensity $I_{\mathsf{growth}}$. The constants which scale the influence of the source term and the relation of diffusive and convective transport are

$$\begin{aligned} \lambda_{\mathsf{EtOH}} &= 0.1676\ \mathrm{W/m\cdot K} && \text{(thermal conductivity of ethanol)}, \\ C_{\mathsf{EtOH}} &= 2441.3\ \mathrm{J/kg\cdot K} && \text{(specific heat capacity of ethanol)}, \\ \Delta h_{\mathsf{cryst}} &= 1.6541 \cdot 10^5\ \mathrm{J/kg} && \text{(specific heat of crystallization)}. \end{aligned}$$

Note that since no other assumptions from [10] are available (thermal conductivity and specific heat capacity of the suspension), we used the values of the solvent instead. The constant $\Delta h_{\mathsf{cryst}}$ is the

specific heat of fusion of ASA. In [10], we found the molar heat of fusion of ASA to be $29800\ ^{\mathrm{J}}/_{\mathrm{mol}}$, which is assumed to be the same as the molar heat of crystallization. Dividing by $M_{\mathsf{ASA}}$, we get the specific heat of ASA crystallization as given above.

Equation (8) has to be transformed into its 2d axisymmetric formulation. For that formulation, the boundary conditions are

$$\begin{cases} T = T_{\mathsf{feed}}^{[i]} & \text{on } \Gamma_{\mathsf{in}}, \\ \frac{\partial T}{\partial n_\Gamma} = 0 & \text{on } \Gamma_{\mathsf{out}} \cup \Gamma_{\mathsf{symm}}, \\ T = T_{\mathsf{wall}}^{[i]} & \text{on } \Gamma_{\mathsf{wall}}. \end{cases}$$

The Dirichlet value at the inflow, $T_{\mathsf{feed}}$, depends on the parameter set $i$. The respective values come from measurements reported in [10] and they are given in Table 2. The Neumann conditions at outflow and symmetry axis are the same conditions that were used for the concentration balance equation. The wall boundary condition is special, since here we prescribed the heat loss through the tube wall by imposing a temperature profile. This is a concession to excluding the tube wall from the modeling domain.

The equation is closed with an initial condition. We assume that the entire tube finds itself at ambient temperature before the hot suspension is pumped into it: $T(0,\cdot) = 297.5\ K$ in $\Omega$.

# 3 Modeling of the Growth Term for the Stochastic Particle Method

For solving of the PBE (1), we want to apply the stochastic particle method that is presented in [35]. It is based essentially on two sources: G.A. Bird's direct simulation Monte–Carlo algorithm for the Boltzmann equation [6] and Gillespie's stochastic algorithm for the simulation of collision growth phenomena in clouds [14, 15]. While the Bird algorithm provides a way to deal with the convective transport part (by splitting), the Gillespie algorithm gives a tool to treat the coagulation part of the equation (by formulating a jump process). Particle growth, which is the third feature of the PBE (1), is included in neither of these original sources, nor is it considered in [35]. Yet in [36], particle surface reaction is included in model and simulation. The way it is done there is closely related to [16, 17], where a stochastic algorithm similar to that introduced in [14] is applied to chemical reactions. Our aim is to simulate growth as a stochastic jump process, and therefore we have to find an appropriate formulation. The key is to understand particle growth as a particle surface reaction rather than as convective transport along the internal coordinate axis. This section compares both approaches and establishes a link between them.

Let, for the sake of simplicity, the particle number density function $f : (0, t_{\mathsf{end}}) \times \Omega_m \to \mathbb{R}_0^+$ only depend on time $t$ and the internal (mass) coordinate $m$, i.e., we assume spatial homogeneity. Further, let particle growth by attachment be the only reason for $f$ to change, thus disregarding coagulation.

First, let us sketch the derivation of a population balance equation under the concept of "particle growth as transport along the internal coordinate axis". This sketch follows the presentation in [37, p. 16]. The main assumption is that for each $m \in \Omega_m$ there exists a (time-independent) growth rate $G(m)$ at which any particle of mass $m$ grows. Fix an arbitrary interval $[a, b] \subseteq \Omega_m$ in the property space. The particle number balance in the interval $[a, b]$ is then given by

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_a^b f(t,m)\,\mathrm{d}m = G(a)f(t,a) - G(b)f(t,b).$$

If one assumes smoothness of $f$ and $G$, both the time derivative and the right-hand side can be written under the integral. Because of the arbitrary choice of $[a, b]$, the equation can be reduced to the integrand, which gives the final PBE

$$\frac{\partial}{\partial t} f(t, m) = -\frac{\partial}{\partial m} G(m) f(t, m). \tag{9}$$

Assuming that $G(m) \equiv$ *const.*, and re-introducing the left-out features (convection, coagulation, dependence on $c$ and $T$), one regains a PBE as used, e.g., in [2, 19]. As result, particle growth appears as a transport term in this PBE.

In the second approach, which will allows for a more straightforward stochastic formulation, one starts with regarding particle surface growth as a chemical reaction. Each particle type $m \in \Omega_m$ acts as a chemical species and $f(t, m)$ can be interpreted as the "number concentrationöf the species at time $t$. We stick to the assumption of spatial homogeneity here. One states that each "growth reaction" for a particle means a mass gain of $\mu > 0$. The growth reactions are of the form

$$m \xrightarrow{r_m} (m + \mu),$$

where $r_m$ is the rate at which this growth reaction takes place. A standard ODE description of this reacting system gives for every $m$

$$\frac{\partial}{\partial t} f(t, m) = r_{m-\mu} f(t, m - \mu) - r_m f(t, m). \tag{10}$$

The first term on the right-hand side stands for gain of $m$-type particles by growth of particles of size $m - \mu$, the second term for loss of $m$-type particles. Comparing (9) and (10) yields that in order to link transport-based and reaction-based growth modeling, the relation

$$-\frac{\partial}{\partial m} G(m) f(t, m) \approx r_{m-\mu} f(t, m - \mu) - r_m f(t, m) \tag{11}$$

must be satisfied in some sense. Abbreviating $\varphi(t, m) := r_m f(t, m)$ for all $m \in \Omega_m$, we observe

$$\varphi(t, m - \mu) - \varphi(t, m) = -\mu \frac{\varphi(t, m - \mu) - \varphi(m)}{-\mu} = -\mu \left( \frac{\partial}{\partial m} \varphi(t, m) + \mathrm{o}(\mu) \right),$$

if $\varphi$ is smooth. Thus, up to first order,

$$r_{m-\mu} f(t, m - \mu) - r_m f(t, m) = -\mu \frac{\partial}{\partial m} \left( r_m f(t, m) \right)$$

holds, and one can interpret relation (11) in this way.

This leads to the following conclusion. If one wants both approaches to model the same physical process from a macroscopic point of view, the identity $G(m) = \mu r_m$ must hold.

# 4 Numerical Methods

This section describes the numerical methods that were used to simulate the ASA flow crystallizer. In particular, the algorithms for the stochastic particles are presented in some detail: the convective transport of these particles and the Markov jump processes, which simulate their growth. At the end of this section, the coupling strategy for the deterministic CFD solver and the stochastic particle solver, as well as the steps for achieving an appropriate coupling are explained.

## 4.1 Convection-Diffusion Equations

The concentration balance equation (6) and the energy balance equation (8) are of convection-diffusion type. Since the diffusion parameters are very small, convection dominates. It is well known that in this situation standard discretizations lead to numerical solutions that are globally polluted with spurious oscillations. Instead, so-called stabilized discretizations have to be used.

For the physical fidelity of the computed results, the solutions obtained with a stabilized discretization must not possess spurious oscillations at all. Competitive numerical studies in [20, 21] showed that first, there are only very few stabilized finite element schemes which lead to oscillation-free solutions. And second, the linear finite element method flux-corrected transport scheme with Crank–Nicolson time discretization (linear CN-FEM-FCT), which is a special variant of a scheme proposed in [25], is a good compromise between accuracy of the solutions and efficiency of the simulations. This scheme was used in our simulations. For a detailed description, the reader is referred to [20, 21].

## 4.2 Spatial Inhomogeneity and Convective Transport of Stochastic Particles

Stochastic simulation algorithms have been extended recently to cover systems with spatial inhomogeneity, and especially convective transport by an external flow field, e.g., see [26, 35, 43]. For our brief presentation of the key points, we stick to [35], though the main idea is the same in all cases. It goes back to the Bird algorithm for the Boltzmann equation [6] and consists essentially in a splitting scheme, splitting the particle transport ("free streaming") and the particle interaction step. Consider a constant splitting time $\Delta t_{\text{split}}$. In the original Boltzmann application, the interaction step meant computing particle collisions, where particles exchanged momentum. In our application, the interaction step means computing particle coagulations and growth.

Some fundamental alterations must be conducted in order to adapt a standard SSA to spatial inhomogeneity. The particle state space is complemented by a bounded spatial domain $\Omega \subset \mathbb{R}^d, d \in \{1, 2, 3\}$, which must then be discretized into a finite number of physically sensible and computationally feasible compartments $K_j, j = 1, ..., M$. In [35], this spatial domain is one-dimensional, and the compartments are just equally spaced sub-intervals.

Each computational particle gets tagged with a spatial coordinate $\boldsymbol{x}_i \in \Omega$. All particles which belong to the same cell $K_j$ are regarded as one *ensemble*, and during the particle interaction step only particles within the same ensemble may interact. The state of the entire process at time $t$ is now

$$X^{\text{SSA}}(t) = \bigcup_{j \in 1,...,M} (m_i^j, \boldsymbol{x}_i^j)_{i=1,...,N_j(t)}.$$

An external flow field $\boldsymbol{u}$ is responsible for the transport of particles. In the free streaming step, each computational particle is moved as if following the trajectories of the numerical velocity field $\boldsymbol{u}$

$$\boldsymbol{x}_i^j \longrightarrow \boldsymbol{x}_i^j + \Delta t_{\text{split}} \boldsymbol{u}(\boldsymbol{x}_i^j).$$

The transport step is performed for each particle, and a check must be performed, whether the relocated particle does still belong to the same cell. If not, then the particle must be removed from its current ensemble and inserted into the ensemble of the new cell.

The particle interaction step poses the difficulty that whenever two particles coagulate, the position of the new particle must be determined, without unwanted side effects. An example for such side-effects was noted in [35]. If inserting the new particle, e.g., at the midpoint in between the two former positions,

the particles collect at the midpoint of the cell. To avoid this side-effect, in [35] it is proposed to choose a new particle position $\boldsymbol{y}$, of a particle which emerged from coagulation of the particles $\left(m_I, \boldsymbol{x}_I\right)$ and $\left(m_J, \boldsymbol{x}_J\right)$ stochastically, distributed according to the elementary probabilities

$$\mathbb{P}(\boldsymbol{y} = \boldsymbol{x}_I) = \frac{m_I}{m_I + m_J}, \quad \mathbb{P}(\boldsymbol{y} = \boldsymbol{x}_J) = \frac{m_J}{m_I + m_J}.$$

The necessary changes when combining stochastic weighting and spatial inhomogeneity are addressed also in [35].

### 4.3 Markov Jump Processes

The stochastic jump process in each cell will be described. Our presentation follows closely that one in [36], see also [32] for an introduction to Markov jump processes. For the sake of simplicity, we give the jump process in terms of the "direct simulation algorithm" described in [35, Sec. 2.1]. It has the advantage to spare some stochastic subtleties of the actually used "stochastic weighted algorithm" [35, Sec. 2.2].

Fix a spatial cell and the belonging particle ensemble, $(K, \mathcal{E})$. Each particle in $\mathcal{E}$ is represented by a spatial and an internal coordinate,

$$e_i = (\boldsymbol{x}_i, m_i),$$

with $\boldsymbol{x}_i \in K \subseteq \Omega_{\boldsymbol{x}}$ and $m_i \in \Omega_m$. The entire ensemble, consisting of $N_{\mathcal{E}}$ particles, is thus

$$\mathcal{E} = (e_1, ..., e_{N_{\mathcal{E}}}).$$

The state of the ensemble can change by particle growth jumps and by particle coagulation jumps. Starting at some time $t$, the system persists in state $\mathcal{E}(t)$ for an exponentially distributed waiting time $\tau$,

$$P(\tau \geq s) = \exp(-\lambda(\mathcal{E})s).$$

The waiting time parameter $\lambda(\mathcal{E})$ is the sum of the individual rates of all jumps that are possible in $\mathcal{E}(t)$. Assembling these in a growth jump rate $\lambda_{\text{grow}}$ and a coagulation jump rate $\lambda_{\text{coag}}$ gives:

$$\lambda(\mathcal{E}) = \lambda_{\text{grow}}(\mathcal{E}) + \lambda_{\text{coag}}(\mathcal{E}).$$

**Particle growth jumps** A particle growth jump by a fixed growth height $\mu$ changes the state of a certain particle $e_i$ in $\mathcal{E}$ according to

$$e_i = (\boldsymbol{x}_i, m_i) \longrightarrow (\boldsymbol{x}_i, m_i + \mu) =: \tilde{e}_i.$$

Growth jumps in $\mathcal{E}$ happen at the total rate

$$\lambda_{\text{grow}}(\mathcal{E}) = \sum_{i=1}^{N_{\mathcal{E}}} \frac{G(c, T, m_i)}{\mu}.$$

The particle $e_i$ for which the next growth jump occurs is chosen with probability

$$\frac{G(c, T, m_i)}{\mu} \left(\lambda_{\text{grow}}(\mathcal{E})\right)^{-1}.$$

In this expression, $c$ and $T$ are assumed to be constant (in space) within the cell $K$.

**Particle coagulation jumps** Coagulation jumps affect two particles, denote them by $e_i$ and $e_j$ (with $i < j$). A coagulation jump has the form

$$e_i, e_j \longrightarrow (\xi(\boldsymbol{x}_i, \boldsymbol{x}_j), m_i + m_j) =: \tilde{e}_i$$

The particle $e_j$ is removed from the ensemble. The placement of the new particle $\tilde{e}_i$ can be done in several ways. It is already mentioned in Section 4.2 that a numerically stable approach consists in using the center of mass.

The total rate of coagulation jumps is the sum of all individual coagulation jump rates of particle pairs. It is calculated as

$$\lambda_{\text{coag}}(\mathcal{E}) = \frac{1}{2N_{\mathcal{E}}} \sum_{i,j=1}^{N_{\mathcal{E}}} K(m_i, m_j).$$

As in the particle growth case, the choice of two particles for a coagulation jump is made at random with the probabilities

$$P(e_i \text{ and } e_j \text{ chosen for coagulation}) = \frac{K(m_i, m_j)}{2N_{\mathcal{E}}}.$$

With these definitions of jumps and jump rates, the stochastics of the process are well defined. Evidently, correlations between particles are neglected.

**Insertion of particles entering the flow domain** At the inflow boundary $\Gamma_{\text{in}}$, particles are inserted into the simulation domain, i.e., into the particle ensembles of those cells, which border the inlet boundary. Particle inceptions are simulated as inception jumps, meaning that each ensemble $\mathcal{E}$ in contact to $\Gamma_{\text{in}}$ gets equipped with an additional jump rate $\lambda_{\text{in}}(\mathcal{E})$ and a corresponding jump, which adds a new particle to the ensemble. The inception jump rates are chosen such that in the very first layer of cells a certain ASA crystal mass concentration is achieved on average. The jump rates, superscripted with the respective parameter setup number, are

$$\lambda_{\text{incept}}^{[1]} = 395 \cdot 10^6 u_z \ {}^{\#\text{particles}}/_{\text{m}^2\text{s}}, \quad \lambda_{\text{incept}}^{[2]} = 661 \cdot 10^6 u_z \ {}^{\#\text{particles}}/_{\text{m}^2\text{s}},$$
$$\lambda_{\text{incept}}^{[3]} = 486 \cdot 10^6 u_z \ {}^{\#\text{particles}}/_{\text{m}^2\text{s}}, \quad \lambda_{\text{incept}}^{[4]} = 552 \cdot 10^6 u_z \ {}^{\#\text{particles}}/_{\text{m}^2\text{s}}.$$

Note that the velocity component $u_z$, which is orthogonal to the inception boundary, must be included in the formulation of the inception jump rate here because the target particle concentration in a cell is proportional to the velocity in that cell. The differences of the inception jump rates are due to the different input particle size distributions, see Table 2. The position within a cell and the amount of ASA of which a newly incepted particle consists must be determined stochastically. For details of this process, see [4].

## 4.4 Coupled Simulation with a Splitting Scheme

For the numerical simulation, we used two in-house code bases: The finite element CFD package PARMOON [13, 41] and the stochastic particle simulation code BRUSH [36]. Our goal was to combine these two codes for the simulation of PBSs. To this end, an appropriate numerical scheme had to be used and the necessary extensions to the software had to be developed and implemented.

A sketch of the numerical scheme for coupling the two codes is presented in Figure 2. In each discrete time, the computation is split into two steps. First, the deterministic CFD equations are solved, thereby

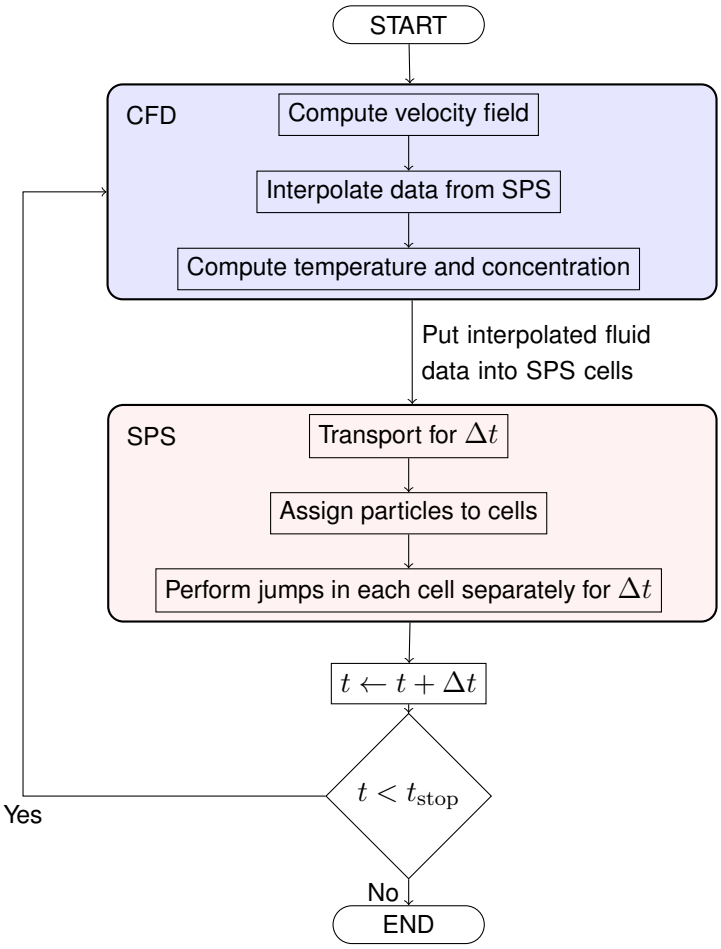



Figure 2: Schematic sketch of the coupled simulation via a splitting scheme.

using the data from the stochastic particle simulation (SPS) from the previous time instance. Then, the CFD data are interpolated via a custom C++ interface between PARMOON and BRUSH. As the second main step of the splitting scheme, the behavior of the particles is simulated, using the algorithms described above.

Besides the interface between the codes, major extensions of the basic codes included the interpolation of the SPS data to be used in the CFD simulations, the simulation of the transport of the particles, and their assignment to mesh cells, see [4] for details of the implementations.

## 5 2d Coupled Simulations of a Tubular Flow Crystallizer

The 2d axisymmetric computational domain was discretized regularly into $5 \times 150$ rectangles. This grid is used for all parts of the simulation, i.e., it is at the same time the finite element mesh for $u$, $c$, and $T$, and the grid of ensemble cells for the stochastic particle simulation (SPS).

The flow equations were discretized with inf-sup stable $Q_2/Q_1$ elements (Taylor–Hood) and for the convection-diffusion equations $Q_1$ elements (piecewise bilinear) were used. This meant a problem size of 7528 d.o.f. for the flow equations and of 906 d.o.f for both convection-diffusion equations. They were solved with the sparse direct solver UMFPACK [8]. Some performance profiling showed that

the time spent in those solvers was sufficiently small such that investing further optimization effort here was unnecessary. In the SPS, computing time was dominantly determined by the number of computational particles per ensemble. We restricted the number of computational particles per cell to $128$, which proved to be a good compromise of computing time and accuracy. We also fixed the maximal number concentration of ASA particles to $1.2 \cdot 10^{11}\ {}^{1}/_{\mathrm{m}^3}$. This meant that each computational particle, upon insertion to the simulation domain, stood representative for $937.5$ physical particles per cubic centimeter. These numbers were kept constant in space and time, and were also the same for all four parameter setups.

Information transfer between SPS and CFD was done with $L^2$ projection operators. In fact, the $Q_2$ velocity was projected to $Q_0$ for both applications. The reason was that otherwise (projection to $Q_1$ for the convection-diffusion equations, $Q_0$ for the SPS), an unwanted numeric effect took place. Since $c$ and $T$ get transported with that velocity which is found at the respective quadrature points, they are transported with a different velocity than $f$, which is transported only by the $Q_0$ velocity in the ensemble cell. Such a lag between fluid quantities and particle transport resulted in the situation that the conservation of mass flux in the tube was lost.

The chosen time step was $\Delta t = 0.025$. With a coarser time step, the CFL-like condition of the CN-FEM-FCT-scheme for the convection-diffusion equations was violated, and a finer time step did not produce significantly better results, as several studies showed (not presented here for the sake of brevity). The same time step was used for CFD and SPS, i.e., there was exactly one transport step plus one process step per $\Delta t$ in the SPS.

The simulations were performed on widely available computing workstations (HP BL460c Gen9 2xXeon, Fourteen-Core 2600MHz). We used eight cores per run, to make use of the inherent shared memory parallelism in the SPS. One simulation run took between 20 and 45 minutes, depending on the parameter setup and the choice of the coagulation parameter.

We ran simulations for all four parameter sets. In all cases, the simulated time was the reported operating time of the experimental tube crystallizer from [10] plus 100 seconds. Those additional $100\,\mathrm{s}$ were the “recording time”, from which the results that are given in this section were computed.

For the SPS, we made use of the “linear process deferrement algorithm” (LPDA), that was proposed in [33] and is implemented in BRUSH. Instead of recomputing jump rates and updating particle properties after each occurrence of a “ linear” process (surface growth), the updates are deferred until the occurrence of a coagulation event and then performed all at once. Due to the relative rareness of coagulation events compared to growth events, this led to five to ten times faster computations in our setup.

An opportunity for speeding up the computations offered itself in caching data that was necessary for transferring functions between the two simulation platforms. We can re-use the geometric multigrid implementation of PARMOON, as long as the grids for SPS and CFD stood in a hierarchical relation to each other. In the current example, where the grids were identical, this was trivially the case.

Simulations were conducted for all four parameter sets, with the goal to reproduce the average particle diameter $\bar{d}$ and its standard deviation $\sigma$ as they were observed at the outlet of the experimental flow crystallizer, see Table 2. The coagulation kernel $K$ and the coagulation intensity parameter $\kappa$ had to be determined. In a first, unsuccessful attempt we tried the additive coagulation kernel (2). The simulations performed with this kernel led to an excessive coagulation of large particles, resulting in an almost unchanged median but unphysically large outliers of the PSD, when increasing the value of $\kappa$. Therefore, we opted for the constant coagulation kernel (3), where the rate of coagulation is unaffected by the size of the particles. With this kernel, we were able to produce physically reasonable

Table 5: Computational results for all four parameter sets at the outflow boundary, averaged over the given time interval. (Number) mean particle diameter $\bar{d}$, standard deviation $\sigma$, maximal observed particle diameter $d_{\text{max}}$, crystalline ASA mass flux $\varphi_m$ and mass flow rate $\dot{m}$, scaled so as to easily compare with the experimental data in Table 2.

| Setup | Time [s] | $\bar{d}$ [$\mu$m] | $\sigma$ [$\mu$m] | $d_{\text{max}}$ [$\mu$m] | $\varphi_m$ [$^{\text{kg}}/_{\text{m}^2\text{s}}$] | $\dot{m}$ [$^{\text{g}}/_{\text{min}}$] |
|---|---|---|---|---|---|---|
| 1 | [900,1000] | 238 | 85 | 953 | 7.443 | 1.40 |
| 2 | [660,760] | 215 | 73 | 858 | 11.571 | 2.18 |
| 3 | [540,640] | 189 | 56 | 682 | 15.088 | 2.84 |
| 4 | [540,640] | 176 | 54 | 645 | 16.499 | 3.12 |

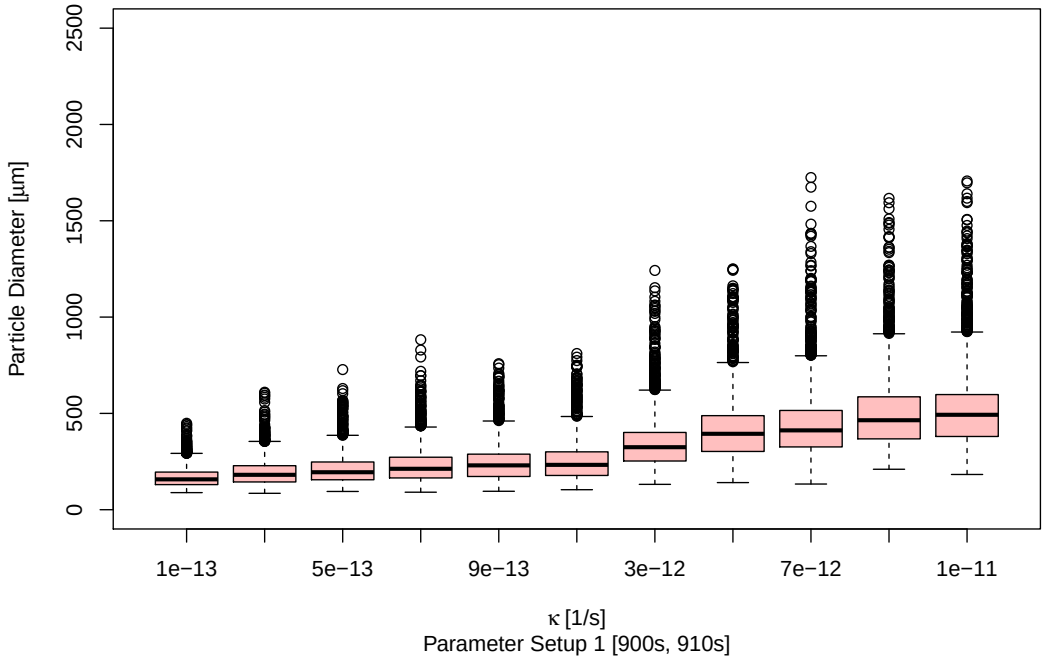
results. Values for $\kappa$ were chosen between $10^{-13}$ and $10^{-11}$ $^1/_{\text{s}}$ in order to determine that value which allowed for the best fitting of the experimental $\bar{d}$ at the outlet. The outlet PSDs obtained with this parameter study are shown in Figure 3. These results were gained by recording the properties, especially the particle mass and diameter, of each particle that left the simulation domain at the outlet, over 10 seconds after the end time of the experiment. The properties of each "measured" stochastic particle were weighted with the product of its stochastic weight and the volume of the ensemble cell it belonged to last. Thus, the boxplots in Figure 3 are weighted boxplots of the "raw" output data of the computational particles. As the results are qualitatively the same for all four parameter sets, only the slowest and fastest setup are presented.

It can be seen that increasing the coagulation parameter leads to a moderate increase of the median distribution, which is as expected. On the other hand, it also leads to an increased variance, as can be seen by the stretch of the boxes and whiskers. This effect was expected, too, since each collision growth event moves one particle far to the right of the median particle size. Collision growth is, due two its big jump height and relative sparseness, less uniform than surface growth, which happens often and results in small size increases only. One can also see that the number and the range of outliers increase, but by far not as severely as was the case for the additive kernel.

The dependence of the mean particle diameter on $\kappa$, disregarding standard deviation, is given in Figure 4 for all four parameter sets.

Comparing the results of the parameter studies to the experimental results, best-fitting values of $\kappa$ could be determined. These values range between $5 \cdot 10^{-13}$ and $10^{-12}$ $^1/_{\text{s}}$. The simulation results proved to be rather sensitive to the parameter. For the two slow setups 1 and 2, $\kappa_1 = 9 \cdot 10^{-13}$ $^1/_{\text{s}}$ and $\kappa_2 = 10^{-12}$ $^1/_{\text{s}}$ proved to be the best choices. For the faster flowing setups, the best parameters were half an order of magnitude smaller, $\kappa_3 = 6 \cdot 10^{-13}$ $^1/_{\text{s}}$ and $\kappa_4 = 5 \cdot 10^{-13}$ $^1/_{\text{s}}$. The results (number mean diameter $\bar{d}$ and standard deviation $\sigma$) which were obtained with these $\kappa$ are given in Table 5, one should compare them to the experimental values in Table 2. One notices that the average could be hit quite well. The standard deviation of the computed particle size distribution is about 1.1 to 1.5 times as high as the standard deviation of the experimental data. The values were gained by averaging in time over all computational particles which left the computational domain at the outlet, within the $100$ s "recording time". We also list the diameter of the largest observed particle. Those values are somewhat too large, and if they would appeared in the actual experiment (tube diameter is $2000$ $\mu$m), blockage would be likely. Indeed, in [10] the largest observed particles were of diameter $500$ $\mu$m and blockage was not observed. Finally, the mass flux and mass flow rate of crystalline ASA (averaged over $100$ s) are given in Table 5. These values are remarkably close to experiments, see Table 2.

Figures 5 and 6 show more results in the same spirit, supporting physical plausibility. In Figure 5, the



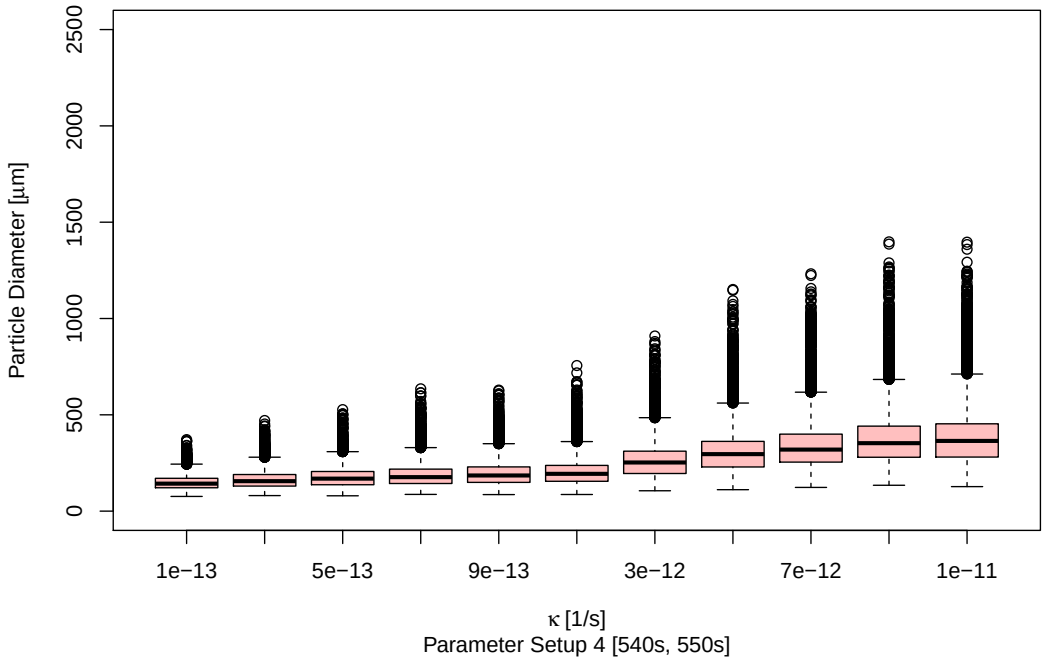




Figure 3: Boxplots of PSD at exit for fastest and slowest parameter set, with different choices of coagulation parameter $\kappa$ in constant coagulation kernel. Box whiskers stretch to 1.5 interquartile range of lower/upper quartile, X-axis scale is only ordinal.

ASA mass flux throughout the tube is shown, averaged over $50\ \mathrm{s}$ "recording time", when stationary conditions have already been reached for a while. One can see that the total ASA mass flux (crystalline plus dissolved ASA) is constant in $z$. This is as expected, since the velocity field is divergence-free

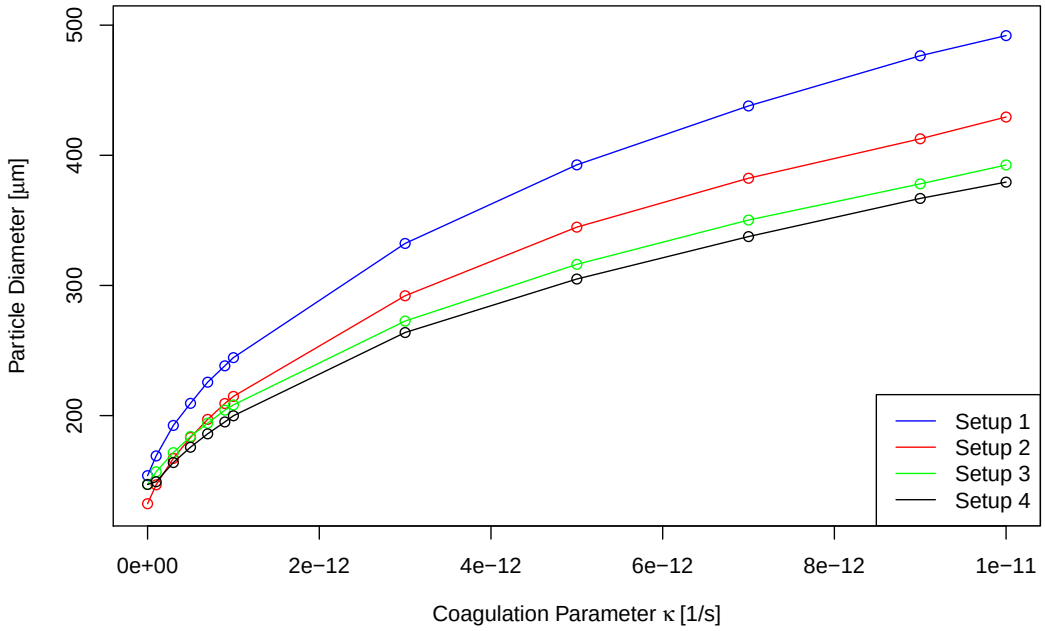



Figure 4: Mean particle diameter at the outlet, in dependence on $\kappa$, constant coagulation kernel, all four parameter sets. Data is averaged over 100 seconds past the end time of the respective experiment.

and ASA mass must neither be lost nor gained. The green and blue lines show how dissolved ASA is consumed by surface growth of the crystals. This process takes place, until supersaturation is zero and equilibrium of dissolved and crystalline ASA is reached. One can see that surface growth is quickest in the slowest setup (1), where equilibrium is reached essentially within the first 3 meters of the tube. The faster the flow, the farther in the tube equilibrium is reached. In the fastest setup (4), it is not reached before meter 8.

Figure 6 shows the first 10 meters of the tube in a zoom. It illustrates an expected and interesting effect: there is a growth delay by coagulation. In Figure 6, the mass flux for four different, well apart coagulation parameters $\kappa$ is shown. One observes that the higher the coagulation parameter is, the shallower is the slope of the mass flux curve, i.e., the slower is the transition of ASA from dissolved to crystalline state by surface growth. The reason for this effect is the surface dependence of the growth intensity $G_m$, see (5). Collision growth maintains mass, but the total crystal surface area is reduced by each coagulation event. Therefore, after coagulation, less crystal surface is available for new material to attach, and surface growth is slowed down. This effect is clearly visible in Figure 6.

Finally, Figure 7 shows the net effect of the (simulated) operation of the flow crystallizer. For all four parameter setups, the initial (at $z = 0$) and final (at $z = 15$) distribution is given, in terms of probability. The data was gained the same way as described before, but applying the same procedure at both outlet and inlet. It can be seen that in all four cases the peak of the distribution moved towards the right, and additionally the initial sharpness is somewhat smeared. This effect is the stronger, the slower the flow is, i.e., the more time the particles have to form larger aggregates by coagulation. One can also observe that the histograms of the slowest setup are somewhat “jagged”. This is due to the lower number of in- or outflowing computational particles per second. The effect could be attenuated somewhat by collecting data for an even longer period of time.

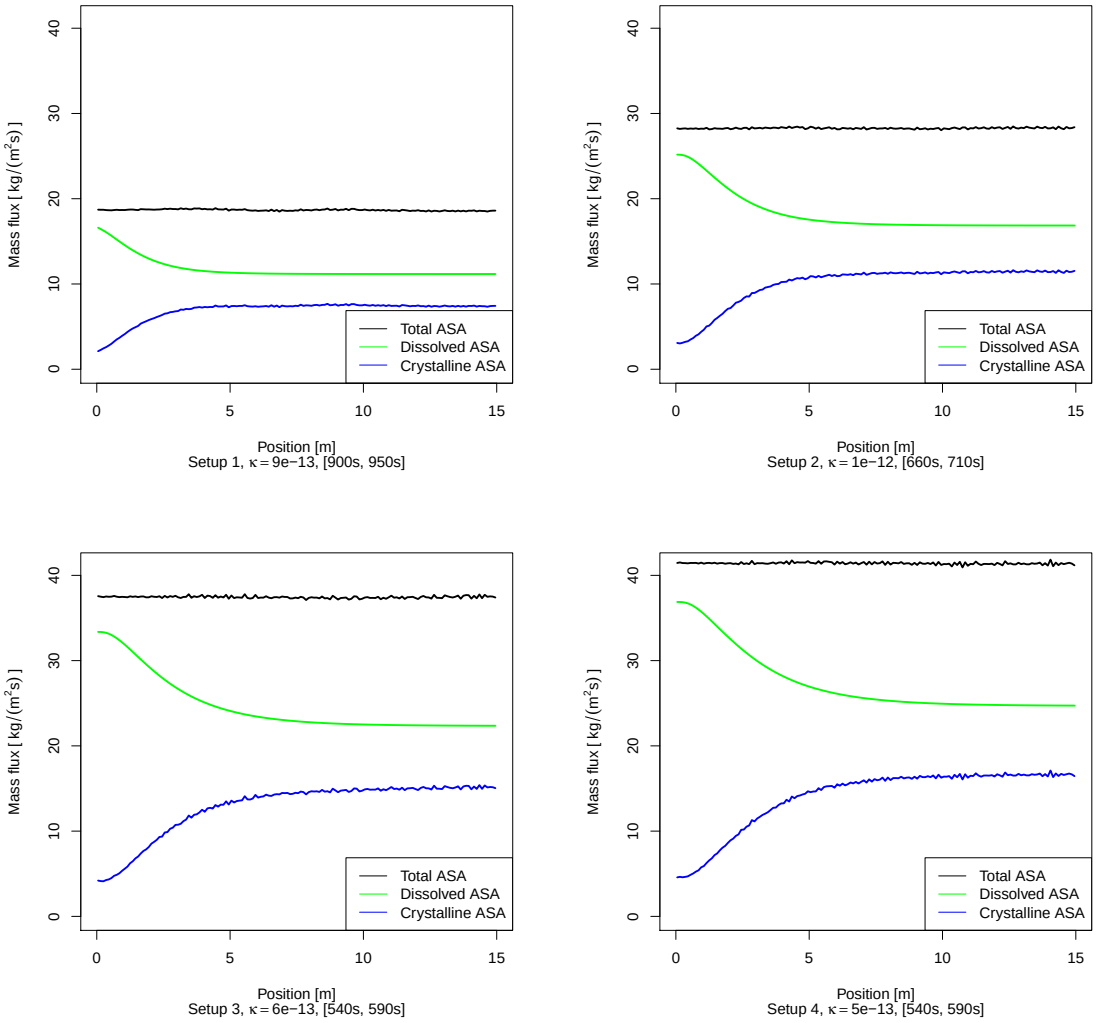



Figure 5: Time-averaged development of mass flux of crystalline and dissolved ASA along the tube, all four setups, best coagulation parameter. In all cases, stationary conditions were reached and the total ASA mass flux is constant (in space, on time average) along the tube.

Several further simulations and postprocessing steps were performed in order to confirm the reliability of our results. All gave positive results. First, we checked the stochastic stability of the final results of Table 5. This we did by sectioning the $100\ \mathrm{s}$ "recording time" at the end of each experiment into four $25\ \mathrm{s}$ intervals and comparing mean diameter and standard deviation of those particles leaving the tube within these intervals. The results are consistent, see Table 6. The values are closer together for the faster flowing setups, which is a direct consequence of the higher number of computational particles leaving the domain. We then made sure that our approach of simulating a single trajectory of the stochastic process was sufficient, by running ten independent realizations of one example. We picked the fastest flowing setup with optimal coagulation parameter. Averaging the results of these runs (first $10\ \mathrm{s}$ of the recording time) gave a mean of $175\ \mu\mathrm{m}$ and, and a mean standard deviation of $53\ \mu\mathrm{m}$, compare with Table 5. The fluctuations of the individual runs about those mean values can be quantified by their standard deviation: it is $0.62$ for the mean values and $0.64$ for the individual standard deviations. We consider both standard deviations sufficiently small to justify the one-trajectory approach.

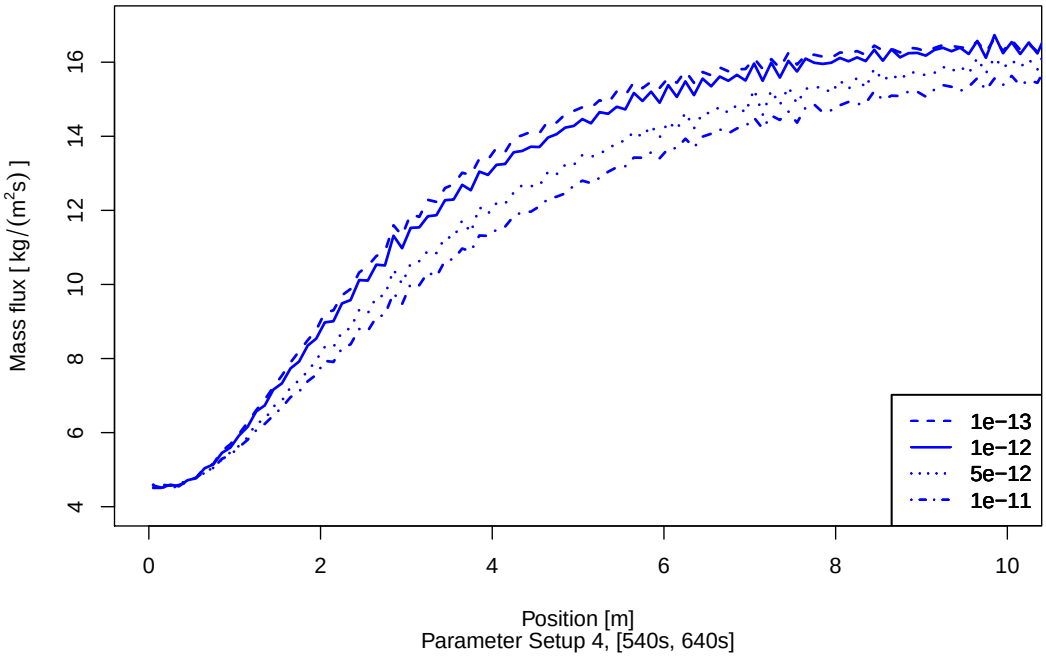



Figure 6: Time-averaged mass flux for four different coagulation parameters $\kappa$, in the first $10\ \mathrm{m}$ of the tube crystallizer after steady-state conditions have been reached.

Table 6: Number mean particle diameter in $\mu\mathrm{m}$ and standard deviation (in brackets), for all four parameter setups and the particular optimal coagulation parameter. The given values refer to the first, second, third and fourth $25\ \mathrm{s}$ of the $100\ \mathrm{s}$ "recording" time interval at the end of each simulation.

| Setup | First $25\ \mathrm{s}$ | Second $25\ \mathrm{s}$ | Third $25\ \mathrm{s}$ | Fourth $25\ \mathrm{s}$ |
|---|---|---|---|---|
| 1 | 240.53 (85.92) | 239.60 (85.24) | 235.61 (83.51) | 237.57 (87.15) |
| 2 | 213.60 (73.11) | 215.05 (73.94) | 215.87 (72.90) | 214.51 (73.62) |
| 3 | 189.05 (56.46) | 188.29 (56.81) | 189.60 (56.55) | 189.16 (56.26) |
| 4 | 175.17 (53.06) | 176.63 (53.43) | 175.12 (53.22) | 175.95 (54.47) |

For the fastest flowing parameter setup, we performed grid refinement tests. The grid was refined only in flow direction, thus reducing the aspect ratio somewhat. The results for the best coagulation parameter showed a slight, but systematic dependence on the grid size, see Table 7. The finer the grid, the smaller was the number mean diameter, and the higher the standard deviation. Additionally, we performed the same grid refinement tests for a no-coagulation setup in order to exclude the SPS treatment as a source for this dependency. The dependency was qualitatively and quantitatively the same for that setup.

In summary, with our new coupled method, we were able to reproduce the experimentally number mean diameter quite well, with a similar coagulation parameter for all four setups. The results are physically plausible. There is an overestimation of the standard deviation, the experimental PSD showed less variance. We can identify two sources of variance. One is particle coagulation, the other is the spread in residence time in the tube crystallizer, that gets introduced by the parabolic Hagen–Poiseuille flow profile. The first source of variance could be reduced by re-modeling the proportion of surface growth and collision growth, favoring surface growth even more. Surface growth does not introduce

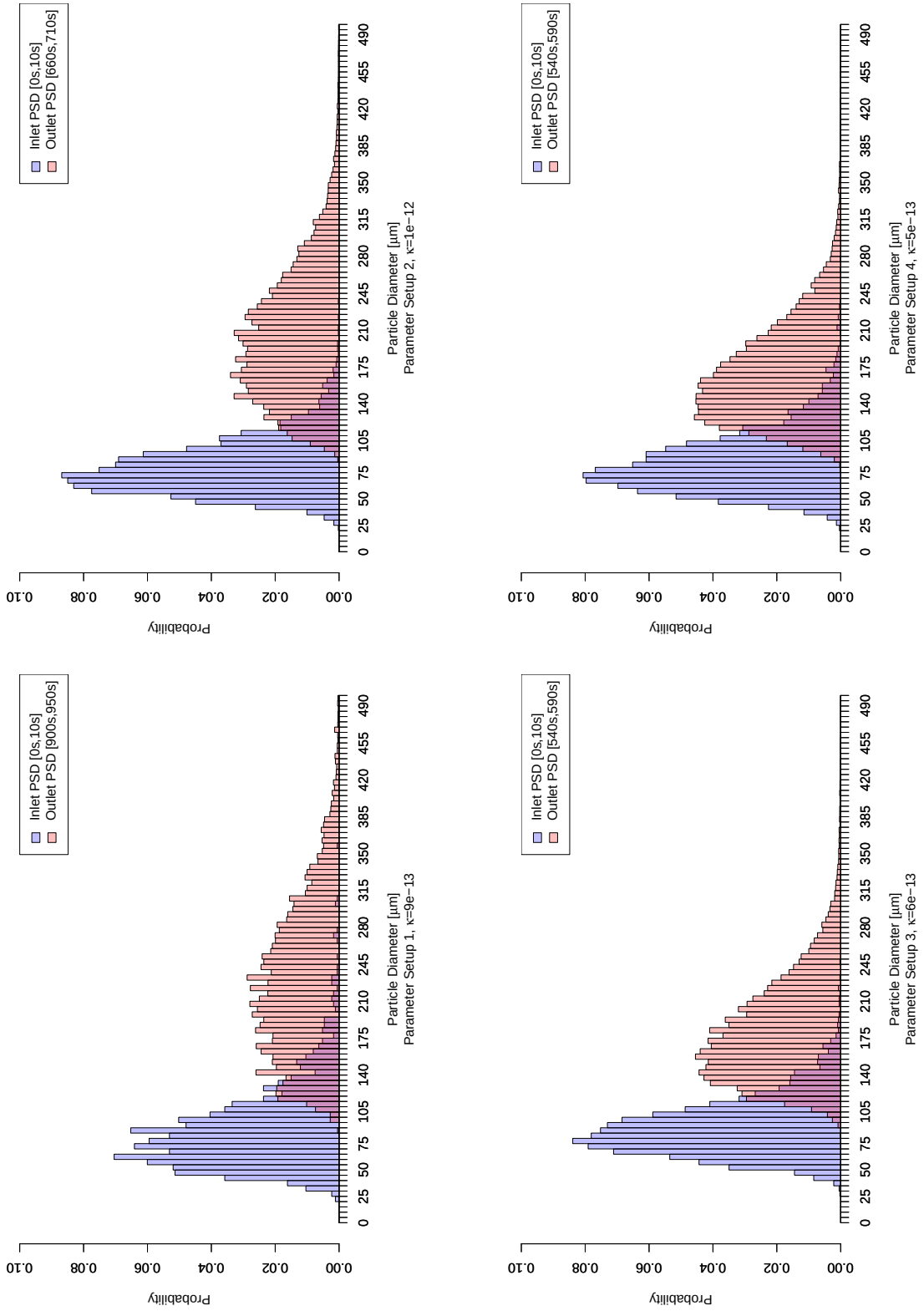



Figure 7: Simulation results for all four setups, best coagulation parameter $\kappa$. Particle size distribution at inlet and outlet is shown, measured over time interval $[0\,\mathrm{s}, 10\,\mathrm{s}]$ (inlet) and the following intervals at the outlet: $[900, 950]$ (setup 1), $[660, 710]$ (setup 2), $[540, 590]$ (setup 3), $[540, 590]$ (setup 4).

Table 7: Results of grid refinement tests. Parameter setup 4, two choices of coagulation parameter $\kappa$, five refinement levels. Values are particle number mean diameter and standard deviation in parentheses, both in $\mu\mathrm{m}$. One observes a slight decline of the mean value and a slight increase of the standard deviation with refinement.

| N cells | 150×5 | 300×5 | 600×5 | 1200×5 | 1500×5 |
|---|---|---|---|---|---|
| $\kappa = 5 \cdot 10^{-13}\ {}^{1}/_{\mathrm{s}}$ | 176 (54) | 175 (54) | 175 (54) | 174 (57) | 174 (58) |
| $\kappa = 0\ {}^{1}/_{\mathrm{s}}$ | 140 (31) | 140 (30) | 140 (31) | 139 (33) | 138 (34) |

as much variance as collision growth does. In addition, we think that replacing the simple coagulation kernel (3) by a more appropriate kernel, which is planned for future work, might lead to a narrower variance. The second source of variance could be dealt with by either taking a step back towards a 1d-model and transporting all particles with the same velocity, or by augmenting the 2d model. Experimental and numerical results from [40] suggest that smaller particles travel slower through a tube crystallizer, since they follow the flow microstructure, thus spending more time in the crystallizer, having more time to grow. If this conjecture is applicable to the ASA crystallizer, too, then capturing this effect in the model could counterbalance the variance introduced with the different streaming layers, thus keeping the standard deviation even closer to the experimental data.

# 6 Summary and Outlook

The newly developed coupling method was successfully applied to the axisymmetric 2d simulation of a flow crystallizer. Experimental results could be reproduced well for four different operating conditions of the crystallizer, the simulation results are physically plausible in all cases, and the computing times were within reasonable bounds. To conduct these simulations, it was necessary to find a coupled formulation of the population balance system and to choose accurate and efficient numerical schemes for each sub-problem. Transfer of information between the CFD part and the stochastic particle simulation had to be implemented and made efficient. The stochastic simulation had to be adapted to two spatial dimensions, since preceding simulations that included convection focused on the 1d case only.

The obtained achievements form a good basis for the extension of the method to higher dimensions. “Higher dimensions” here refers to both the external and the internal coordinates. As for the external coordinates, the method should be used within a 3d setting. From the CFD perspective, this leads to more degrees of freedom and denser matrices, which necessitates more sophisticated solvers. Adapting the SPS to be performed in a 3d flow domain, on the other hand, will require great attention to details, and these adaptations are the main challenges to overcome.

The first issue are wall boundary conditions. We will have to formulate and implement reflection conditions, which mimic particle-wall collisions. A computational issue is the search for particles after the transport step. If a particle is moved only within its cell or to a neighboring cell, locating all $N$ computational particles after the transport step can be done in $\mathcal{O}(N)$ operations with a rather naive approach. But if particles move farther than to a neighbor cell, a more refined technique is required to keep the computational effort within bounds.

Several of the described difficulties might have appeared in 2d already, but were not apparent in the ASA crystallizer example, because there all stream lines of the flow are aligned.

As for the internal coordinates, the severity of complications connected to an extension depends on the used particle model. The SPS is a very natural framework for complicated particle models, due

to the presence of computational particles. These can contain further information much easier than non-stochastic models, which rely on averaged particle properties.[1] Other inner dimensions could be a second or third parameter for the geometry (needle shaped, elliptical, hexagonal particles, ...) or further chemical substances. The main challenge here is to define a sensible addition on the inner coordinate space, which represents coagulation, and likewise a coagulation kernel $K$.

The simulations that were presented in this paper were performed in the spirit of a "proof-of-concept" example. This example is well suited for classical methods, such as Direct Discretization Simulation [39], Method Of Moments [29], or an Operator Splitting approach [1]. Therefore, the example would be fit for a direct comparison of their accuracy and efficiency. This undertaking is not in the scope of this work and will be the subject of further studies.

# References

[1] Naveed Ahmed, Gunar Matthies, and Lutz Tobiska. Finite element methods of an operator splitting applied to population balance equations. *J. Comput. Appl. Math.*, 236(6):1604–1621, 2011.

[2] Felix Anker, Sashikumaar Ganesan, Volker John, and Ellen Schmeyer. A comparative study of a direct discretization and an operator-splitting solver for population balance systems. *Computers & Chemical Engineering*, 75:95 – 104, 2015.

[3] G Barthelmes, S.E Pratsinis, and H Buggisch. Particle size distributions and viscosity of suspensions undergoing shear-induced coagulation and fragmentation. *Chemical Engineering Science*, 58(13):2893–2902, 2003.

[4] Clemens Bartsch. *Embedding a Stochastic Population Balance Solver into a Finite Element Flow Simulation*. PhD thesis, Freie Universität Berlin, Department of Mathematics and Computer Science, 2018. in preparation.

[5] M. O. Besenhard, R. Hohl, A. Hodzic, R. J. P. Eder, and J. G. Khinast. Modeling a seeded continuous crystallizer for the production of active pharmaceutical ingredients: Modeling a seeded continuous crystallizer. *Crystal Research and Technology*, 49(2-3):92–108, 2014.

[6] G. A. Bird. Direct Simulation and the Boltzmann Equation. *Physics of Fluids*, 13(11):2676, 1970.

[7] Róbert Bordás, Volker John, Ellen Schmeyer, and Dominique Thévenin. Numerical methods for the simulation of a coalescence-driven droplet size distribution. *Theor. Comput. Fluid Dyn.*, 27(3-4):253–271, 2013.

[8] Timothy A. Davis. Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Software*, 30(2):196–199, 2004.

[9] R. J. P. Eder, E. K. Schmitt, J. Grill, S. Radl, H. Gruber-Woelfler, and J. G. Khinast. Seed loading effects on the mean crystal size of acetylsalicylic acid in a continuous-flow crystallization device. *Crystal Research and Technology*, 46(3):227–237, 2011.

[1] Averaged properties are of importance for the SPS, too, but only in postprocessing and information transfer, not essentially.

[10] Rafael J. P. Eder, Stefan Radl, Elisabeth Schmitt, Sabine Innerhofer, Markus Maier, Heidrun Gruber-Woelfler, and Johannes G. Khinast. Continuously Seeded, Continuously Operated Tubular Crystallizer for the Production of Active Pharmaceutical Ingredients. *Crystal Growth & Design*, 10(5):2247–2257, 2010.

[11] Rafael J. P. Eder, Simone Schrank, Maximilian O. Besenhard, Eva Roblegg, Heidrun Gruber-Woelfler, and Johannes G. Khinast. Continuous Sonocrystallization of Acetylsalicylic Acid (ASA): Control of Crystal Size. *Crystal Growth & Design*, 12(10):4733–4738, 2012.

[12] Y. Efendiev and M.R. Zachariah. Hybrid monte carlo method for simulation of two-component aerosol coagulation and phase segregation. *Journal of Colloid and Interface Science*, 249(1):30 – 43, 2002.

[13] S. Ganesan, V. John, G. Matthies, R. Meesala, S. Abdus, and U. Wilbrandt. An object oriented parallel finite element scheme for computing pdes: Design and implementation. In *IEEE 23rd International Conference on High Performance Computing Workshops (HiPCW) Hyderabad*, pages 106–115. IEEE, 2016.

[14] Daniel T. Gillespie. The Stochastic Coalescence Model for Cloud Droplet Growth. *Journal of the Atmospheric Sciences*, 29(8):1496–1510, 1972.

[15] Daniel T. Gillespie. An Exact Method for Numerically Simulating the Stochastic Coalescence Process in a Cloud. *Journal of the Atmospheric Sciences*, 32(10):1977–1989, 1975.

[16] Daniel T Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434, 1976.

[17] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[18] Dirk Grosschmidt, Peter Habisreuther, and Henning Bockhorn. Calculation of the size distribution function of soot particles in turbulent diffusion flames. *Proceedings of the Combustion Institute*, 31(1):657 – 665, 2007.

[19] Wolfgang Hackbusch, Volker John, Aram Khachatryan, and Carina Suciu. A numerical method for the simulation of an aggregation-driven population balance system. *International Journal for Numerical Methods in Fluids*, 69(10):1646–1660, 2012.

[20] Volker John and Julia Novo. On (essentially) non-oscillatory discretizations of evolutionary convection-diffusion equations. *J. Comput. Phys.*, 231(4):1570–1586, 2012.

[21] Volker John and Ellen Schmeyer. Finite element methods for time-dependent convection-diffusion-reaction equations with small diffusion. *Comput. Methods Appl. Mech. Engrg.*, 198(3-4):475–494, 2008.

[22] Volker John and Carina Suciu. Direct discretizations of bi-variate population balance systems with finite difference schemes of different order. *Chem. Eng. Sci.*, 106(0):39 – 52, 2014.

[23] M. Kostoglou, A.G. Konstandopoulos, and S.K. Friedlander. Bivariate population dynamics simulation of fractal aerosol aggregate coagulation and restructuring. *Journal of Aerosol Science*, 37(9):1102 – 1115, 2006.

[24] G. Kotalczyk and F.E. Kruis. A monte carlo method for the simulation of coagulation and nucleation based on weighted particles and the concepts of stochastic resolution and merging. *Journal of Computational Physics*, 340:276 – 296, 2017.

[25] Dmitri Kuzmin. Explicit and implicit FEM-FCT algorithms with flux linearization. *J. Comput. Phys.*, 228(7):2517–2534, 2009.

[26] Kok Foong Lee, Robert I.A. Patterson, Wolfgang Wagner, and Markus Kraft. Stochastic weighted particle methods for population balance equations with coagulation, fragmentation and spatial inhomogeneity. *Journal of Computational Physics*, 303:1–18, 2015.

[27] Christian Lindenberg, Martin Krättli, Jeroen Cornel, Marco Mazzotti, and Jörg Brozio. Design and Optimization of a Combined Cooling/Antisolvent Crystallization Process. *Crystal Growth & Design*, 9(2):1124–1136, 2009.

[28] Daniele L. Marchisio and Rodney O. Fox. Solution of population balance equations using the direct quadrature method of moments. *Journal of Aerosol Science*, 36(1):43 – 73, 2005.

[29] Daniele L. Marchisio, Jesse T. Pikturna, Rodney O. Fox, R. Dennis Vigil, and Antonello A. Barresi. Quadrature method of moments for population-balance equations. *AIChE Journal*, 49(5):1266–1276, 2003.

[30] Robert McGraw. Description of aerosol dynamics by the quadrature method of moments. *Aerosol Science and Technology*, 27(2):255–265, 1997.

[31] Peter Neugebauer and Johannes G. Khinast. Continuous Crystallization of Proteins in a Tubular Plug-Flow Crystallizer. *Crystal Growth & Design*, 15(3):1089–1095, 2015.

[32] J. R. Norris. *Markov chains*. Cambridge series on statistical and probabilistic mathematics. Cambridge University Press, Cambridge, UK ; New York, 1st pbk. ed edition, 1998.

[33] R. I. A. Patterson, J. Singh, M. Balthasar, M. Kraft, and J. R. Norris. The Linear Process Deferment Algorithm: A new technique for solving population balance equations. *SIAM Journal on Scientific Computing*, 28(1):303–320, 2006.

[34] Robert I. A. Patterson. Convergence of Stochastic Particle Systems Undergoing Advection and Coagulation. *Stochastic Analysis and Applications*, 31(5):800–829, 2013.

[35] Robert I. A. Patterson and Wolfgang Wagner. A stochastic weighted particle method for coagulation-advection problems. *SIAM Journal on Scientific Computing*, 34:B290–B311, 2012.

[36] Robert I.A. Patterson, Wolfgang Wagner, and Markus Kraft. Stochastic weighted particle methods for population balance equations. *Journal of Computational Physics*, 230(19):7456–7472, 2011.

[37] Doraiswami Ramkrishna. *Population balances: theory and applications to particulate systems in engineering*. Academic Press, San Diego, CA, 2000.

[38] Shraddha Shekar, Alastair J. Smith, William J. Menz, Markus Sander, and Markus Kraft. A multidimensional population balance model to describe the aerosol synthesis of silica nanoparticles. *Journal of Aerosol Science*, 44:83 – 98, 2012.

[39] O. C. Suciu. *Numerical methods based on direct discretizations for uni- and bi-variate population balance systems*. PhD thesis, Freie Universität Berlin, 2013.

[40] Viktoria Wiedmeyer, Felix Anker, Clemens Bartsch, Andreas Voigt, Volker John, and Kai Sundmacher. Continuous Crystallization in a Helically Coiled Flow Tube: Analysis of Flow Field, Residence Time Behavior, and Crystal Growth. *Industrial & Engineering Chemistry Research*, 56(13):3699–3712, 2017.

[41] Ulrich Wilbrandt, Clemens Bartsch, Naveed Ahmed, Najib Alia, Felix Anker, Laura Blank, Alfonso Caiazzo, Sashikumaar Ganesan, Swetlana Giere, Gunar Matthies, Raviteja Meesala, Abdus Shamim, Jagannath Venkatesan, and Volker John. ParMooN—A modernized program package based on mapped finite elements. *Computers & Mathematics with Applications*, 2017.

[42] Edward K.Y. Yapp, Robert I.A. Patterson, Jethro Akroyd, Sebastian Mosbach, Erin M. Adkins, J. Houston Miller, and Markus Kraft. Numerical simulation and parametric sensitivity study of optical band gap in a laminar co-flow ethylene diffusion flame. *Combustion and Flame*, 167:320 – 334, 2016.

[43] Haibo Zhao and Chuguang Zheng. A population balance-Monte Carlo method for particle coagulation in spatially inhomogeneous systems. *Comput. Fluids*, 71:196–207, 2013.