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

Preprint ISSN 0946 – 8633

Numerical methods for the simulation of an aggregation-driven droplet size distribution

Róbert Bordás[1],Volker John[2], Ellen Schmeyer[3] , Dominique Thévenin[1]

submitted: July 4, 2012

[1] University of Magdeburg
"Otto von Guericke"
Universitätsplatz 2
39106 Magdeburg
Germany
email: bordas@ovgu.de

[2] Weierstrass Institute
Mohrenstr. 39, 10117 Berlin, Germany and
Free University of Berlin
Department of Mathematics and Computer Science
Arnimallee 6, 14195 Berlin, Germany
email: Volker.John@wias-berlin.de

[3] Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
email: Ellen.Schmeyer@wias-berlin.de

No. 1716
Berlin 2012

2010 *Mathematics Subject Classification.* 76F65, 76T10.

Key words and phrases. population balance systems, TVD-ENO scheme, group FEM-FCT method, numerical computation of coalescence integrals.

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

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

Abstract

A droplet size distribution in a turbulent flow field is considered and modeled by means of a population balance system. This paper studies different numerical methods for the 4D population balance equation and their impact on an output of interest, the time-space-averaged droplet size distribution at the outlet which is known from experiments. These methods include different interpolations of the experimental data at the inlet, various discretizations in time and space, and different schemes for computing the aggregation integrals. It will be shown that notable changes in the output of interest might occur. In addition, the efficiency of the studied methods is discussed.

1 Introduction

Population balance systems model particle populations where not individual particles are of importance but the behavior of particle distributions. They are widely used for simulating processes in chemical engineering [1]. Another application, which will be considered in this paper, is the modeling of the behavior of droplet size distributions (DSD) in flows [2, 3], which is of importance, e.g., for the simulation of processes in clouds. The dominating mechanism for the evolution of the DSD in this application is the aggregation of droplets.

The basic form of a population balance system consists of an equation for the flow field and an equation for the DSD. Additional equations might be present, which model, e.g., energy and mass balances. The equation for the DSD possesses some properties which make its numerical simulation challenging. First, the DSD depends not only on time and space, as e.g. the velocity of the flow field, but also on properties of the droplets, the so-called internal coordinates. Consequently, this equation is defined in a higher-dimensional domain than the other equations of the system. Second, the modeling of aggregation leads to integral terms of convolution-type whose efficient numerical evaluation is difficult. And third, the equations are transport-dominated, such that the application of special discretization techniques is necessary.

There are many ways to treat the difficulties of simulating the equation for the DSD. An often used way to avoid the increase in the dimension consists in using methods of moments (MOM), see [4] for the first approach in this direction. In this technique, only the first moments of the DSD are simulated. This approach requires the closure of the system for the first moments. The currently most popular way is the quadrature MOM (QMOM) [5] which applies a numerical quadrature whose weights and nodes depend on the DSD (of a previous time). The direct QMOM (DQMOM) [6] simulates the weights and the nodes of the quadrature directly. However, it is well known that the reconstruction of a DSD from its first moments is a severely ill-posed problem and only a few reconstruction algorithms are available [7, 8].

This paper considers direct discretizations of the equation for the DSD, that means the equation is considered in the higher-dimensional domain. To our best knowledge, numerical comparisons of direct and moment-based methods are not yet available. But we think that the direct approach possesses the potential for more accurate results. The use of many different numerical methods is also possible in this approach. Some investigations of the effect of different methods in the direct approach for population balance systems on output quantities of interest can be found already in [9, 10]. For instance, it turned out in these studies that in certain situations different discretizations of the transport terms might lead to qualitatively different results.

In the numerical studies presented in this paper, the quantity of interest is a time-space-averaged DSD at the outlet of the flow domain. This quantity is among the most insensitive quantities of interest which can be considered. At any rate, it is less sensitive than the quantities studied in [9, 10]. The droplets are contained in a turbulent flow field. Experimental data are available for both the flow field and the DSD, see [11, 12]. With respect to the numerical methods for the DSD, different approximations of the inlet boundary condition, different discretizations for the temporal and transport-dominated spatial derivatives, and different methods for evaluating the aggregation term are studied.

The paper is organized as follows. The model for the DSD in a flow field is introduced in Section 2. Then, Section 3 describes shortly the numerical method for the simulation of the turbulent flow field. The numerical methods for the equation of the DSD are explained in detail in Section 4. Numerical studies are presented in Section 5. Finally, a summary and an outlook are given.

2 The Population Balance Model for the Simulation of the DSD

The droplets are considered in an incompressible flow field which is modeled by the incompressible Navier–Stokes equations

$$\begin{aligned} \rho\mathbf{u}_t - 2\mu\nabla\cdot\mathbb{D}(\mathbf{u}) + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p &= \mathbf{0} && \text{in } (0,t_e)\times\Omega, && (1)\\ \nabla\cdot\mathbf{u} &= 0 && \text{in } (0,t_e)\times\Omega, && (2) \end{aligned}$$

where $\mathbf{u}$ (m/s) is the fluid velocity vector, p (Pa) is the pressure, $\rho = 1.2041$ kg/m^3 is the density of air at 293.15 K, $\mu = 18.15\cdot 10^{-6}$ kg/(m s) is the dynamic viscosity of air at the same temperature, $\mathbb{D}(\mathbf{u}) = (\nabla\mathbf{u} + (\nabla\mathbf{u})^T)/2$ is the velocity deformation tensor, and t_e denotes the final time. Because of the low Mach numbers in the experiments, the density is assumed to be constant. The term describing the gravitational acceleration is included into the pressure.

Appropriate boundary conditions and an initial condition are needed to close the Navier–Stokes equations (1), (2). The experiments were conducted in a wind tunnel. The section for the measurements was of hexahedral shape and it was situated in the lower half of the wind tunnel. Because of some uncertainties concerning correct boundary conditions at the outlet, the computational domain $\Omega = (0, 0.5) \times (-0.225, 0.225) \times (-0.18, 0)$ m^3 is chosen to be somewhat longer than the section for the measurement, which had a length of 0.4 m. In this way, a slight incorrectness of the applied outlet boundary condition does not influence the computed values at the plane that corresponds to the outlet of the measurement section. Hence, an error source in the comparison of experimental and numerical data was eliminated.

Fully developed flow fields, which were computed in a pre-processing step, are used as initial conditions. The boundary conditions at the inflow boundary $\Gamma_{\text{in}} = \{0\} \times (-225, 225) \times (-180, 0)$ are based on experimental data. To model time–dependent inflows, a time-averaged experimental velocity $\mathbf{u}_{\text{exp}}(0, y, z)$ is disturbed with Gaussian noise

$$\mathbf{u}(t, 0, y, z) = \mathbf{u}_{\text{exp}}(0, y, z) + \text{rand}_{\text{normal}}(t, 0, y, z)\boldsymbol{\sigma}_{\text{exp}}(0, y, z) \quad \text{on } (0, t_e) \times \Gamma_{\text{in}},$$

where $\boldsymbol{\sigma}_{\text{exp}}(0, y, z)$ is the corresponding standard deviation, which is also known from the experiments, and $\text{rand}_{\text{normal}}(t, 0, y, z)$ denotes a normally distributed random number. The computation of the random number is performed with the Box–Muller scheme. Additional values at the inlet can be computed using bilinear interpolation. The second and third component of the inlet velocity are set to be zero for all flows.

Experimental data at the outlet $\Gamma_{\text{out}} = \{0.5\} \times (-225, 225) \times (-180, 0)$ are not available. In all numerical simulations, the so-called do-nothing condition

$$(2\nu\mathbb{D}(\mathbf{u}) - p\mathbb{I}) \cdot \mathbf{n} = \mathbf{0} \quad \text{on } (0, t_e) \times \Gamma_{\text{out}}$$

is used. From the location of the measurement section, it follows that the top plane $\Gamma_{\text{top}} = (0, 0.5) \times (-0.225, 0.225) \times \{0\}$ is a symmetry plane. This symmetry condition is given by

$$\begin{aligned} \mathbf{u} \cdot \mathbf{n} &= 0 \quad \text{on } (0, t_e) \times \Gamma_{\text{top}}, \\ \mathbf{n}^T (2\nu\mathbb{D}(\mathbf{u}) - p\mathbb{I})\, \boldsymbol{\tau}_k &= 0 \quad \text{on } (0, t_e) \times \Gamma_{\text{top}},\ k = 1, 2, \end{aligned}$$

where $\mathbf{n}$ denotes the unit normal vector on the boundary in outer direction and $\boldsymbol{\tau}_i, i \in 1, 2$, are two tangential vectors on the boundary. This condition describes free slip without penetration. On all other boundaries $\Gamma = \partial\Omega \setminus (\Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{top}})$, a free slip with penetration condition is used.

$$\begin{aligned} \mathbf{n}^T (2\nu\mathbb{D}(\mathbf{u}) - p\mathbb{I})\, \mathbf{n} &= 0 \quad \text{on } (0, t_e) \times \Gamma, \\ \mathbf{n}^T (2\nu\mathbb{D}(\mathbf{u}) - p\mathbb{I})\, \boldsymbol{\tau}_k &= 0 \quad \text{on } (0, t_e) \times \Gamma,\ k = 1, 2. \end{aligned}$$

The DSD is modeled by a population balance equation which includes the transport of the droplets along the streamlines, their growth in supersaturated air, and the aggregation of droplets

$$\frac{\partial f}{\partial t} + \mathbf{u}_{\text{drop}} \cdot \nabla f + \frac{\partial}{\partial d}\left(\frac{a}{d} f\right) = A_+ + A_- \quad \text{in } (0, t_e) \times \Omega \times (d_{\min}, d_{\max}). \tag{3}$$

In (3), f (no./m^4) is the DSD, d (m) is the droplet diameter with $d \in [d_{\min}, d_{\max}]$, $\mathbf{u}_{\text{drop}}$ (m/s) is the velocity of the droplets, a (m^2/s) is the growth rate, A_+ is the source of aggregation and A_- is the aggregation sink. The growth constant is given by $a = 5.0613 \cdot 10^{-12}$ m^2/s, see [12] for the derivation of this value. Experimental data for the droplet velocity are available at the planes $x = 0$ m, $x = 0.2$ m, and

$x = 0.4$ m. A linear interpolation in $[0, 0.4]$ m and a constant extrapolation in $[0.4, 0.5]$ m are used to define $\mathbf{u}_{\text{drop}}$ in Ω, see [12] for details.

The model for the aggregation is explained best by considering the volume of the droplets, e.g. see [4, 13]. Let f_V be the DSD and $A_{+,V}$, $A_{-,V}$ be source and sink, all with respect to the volume of the droplets. Then, the source term describes the amount of droplets of volume V which are created by the aggregation of two droplets with volume V' and $V - V'$, $V' \in (0, V)$. There are two realizations of this event, namely the first droplet is of volume V', the second one has volume $V - V'$, and vice versa. Hence, the model has the form

$$A_{+,V} = \frac{1}{2} \int_0^V \kappa_{\text{agg}}(V - V', V') f_V(V - V') f_V(V')\, dV'. \tag{4}$$

The sink term models the amount of droplets of volume V which disappear because they aggregate with other droplets of volume $V' \in (0, V_{\max})$

$$A_{-,V} = -\int_0^{V_{\max}} \kappa_{\text{agg}}(V, V') f_V(V) f_V(V')\, dV' = -f_V(V) \int_0^{V_{\max}} \kappa_{\text{agg}}(V, V') f_V(V')\, dV'. \tag{5}$$

Consequently, $A_{+,V} + A_{-,V}$ models the change of droplets of volume V due to the aggregation. In (4) and (5), the term κ_{agg} (m^3/s) is the aggregation kernel.

It is also possible to define the aggregation integrals in terms of the diameter, see [13],

$$\begin{aligned} A_+ &= \frac{d^2}{2} \int_{d_{\min}}^{d} \frac{\kappa_{\text{agg}}\left(\left(d^3 - (d')^3\right)^{1/3}, d'\right)}{(d^3 - d'^3)^{2/3}} f\left(\left(d^3 - (d')^3\right)^{1/3}\right) f(d')\, dd', \quad (6)\\ A_- &= -f(d) \int_{d_{\min}}^{d_{\max}} \kappa_{\text{agg}}(d, d') f(d')\, dd'. \quad (7) \end{aligned}$$

The aggregation kernel, which will be described in terms of volume, is the product of two factors $\kappa_{\text{agg}}(V, V') = p_{\text{col}}(V, V') p_{\text{eff}}(V, V')$. The first factor describes the probability of the collision of two droplets with volume V and V'. The efficiency of the collisions, that means the amount of collisions which actually lead to aggregations, is modeled by the second factor. This factor is chosen to be constant since other models are not available. This constant can be included into scaling factors for the individual terms of the following kernel, see [14, 15],

$$\begin{aligned} \kappa_{\text{agg}}(V, V') &= C_{\text{brown}} \frac{2k_B T}{3\mu} \left(\sqrt[3]{V} + \sqrt[3]{V'}\right) \left(\frac{1}{\sqrt[3]{V}} + \frac{1}{\sqrt[3]{V'}}\right) \\ &\quad + C_{\text{shear}} \sqrt{2 \nabla \mathbf{u}_{\text{drop}} : \nabla \mathbf{u}_{\text{drop}}} \left(\sqrt[3]{V} + \sqrt[3]{V'}\right)^3, \quad (8) \end{aligned}$$

where $k_B = 1.3806504\ 10^{-23}$ J/K is the Boltzmann constant. The first part in (8) is Brownian-motion-generated. It is of importance if small droplets are involved in the aggregation since in this case the last factor becomes large. The second term is shear-induced [16] and it is important if both droplets are large. The identification of the model parameters C_{brown} and C_{shear} was the main topic of the studies performed in [12]. Using one of the methods which is described below for simulating the population balance system, see Section 5, the values $C_{\text{brown}} \simeq 1.5 \cdot 10^6$ and $C_{\text{shear}} \simeq 0.1$ were obtained.

The boundary condition for the droplets at the inlet is based on experimental data. Measurements were performed at a grid of nodes at the inlet. A detailed description of the conversion of the experimental data to values for the DSD in these nodes, denoted by $f_{\text{in,exp}}(\mathbf{x}, d)$, and the corresponding standard deviation

$\sigma_f(\mathbf{x}, d)$ can be found in [12]. The values in the nodes are interpolated to the inlet boundary and the boundary condition reads as follows

$$f(t, \mathbf{x}, d) = \begin{cases} f_{\text{in,exp}}(\mathbf{x}, d) + \text{rand}_{\text{normal}}(t, \mathbf{x})\sigma_f(\mathbf{x}, d), & \mathbf{x} = (0, y, z) \in \Gamma_{\text{in}} \\ 0, & \text{at } d = d_{\text{min,art}}. \end{cases}$$

Here, $d_{\text{min,art}} = 0$ m is an artificial smallest diameter for the droplets which is introduced to define the necessary boundary conditions because of the positive growth rate, see [12] for a discussion of this topic. In $[d_{\text{min,art}}, d_{\text{min}})$, the DSD is set to be $f(t, \mathbf{x}, d) = 0$. The initial condition is given by $f(0, \mathbf{x}, d) = 0$ in $\Omega \times (d_{\text{min,art}}, d_{\text{max}})$.

The general solution strategy for the population balance system is as follows. In each discrete time, first the Navier–Stokes equations (1), (2) are solved and the flow field is computed. Then, the equation for the DSD (3) is solved, where the aggregation terms are always treated explicitly with respect to the DSD. With this approach, the problem in 4D becomes linear.

3 A Finite Element Variational Multiscale (FEVMS) Method for the Simulation of Turbulent Flows

The turbulent flow field is simulated with a projection-based FEVMS method. VMS methods control the influence of the turbulence model in an appropriate way. To this end, so-called small resolved scales are defined to which the direct influence of the turbulence model is restricted. In the projection-based FEVMS method, standard finite element spaces are used to model all resolved scales. An additional finite element space is necessary to perform the decomposition of the resolved scales. The large scales are defined by a variational projection into this large scale space, which is given explicitly as an additional equation in the considered method. Together with the unresolved scales, a three-scale decomposition of the flow field is defined.

VMS methods are based on a variational formulation of the underlying equation. Let $V^h \times Q^h$ be the pair of inf-sup stable, conforming finite element spaces for the velocity and the pressure. The additional large scale finite element space is a space of symmetric 3×3 tensor-valued functions $L^H \subset \{\mathbb{L} \in (L^2(\Omega))^{3\times 3}, \mathbb{L}^T = \mathbb{L}\}$. The semi-discrete (continuous in time) projection-based FEVMS method reads as follows: Find $\mathbf{u}^h : [0, t_e] \to V^h$, $p^h : (0, t_e] \to Q^h$, and $\mathbb{G}^H : [0, t_e] \to L^H$ such that

$$\begin{aligned} (\mathbf{u}_t^h, \mathbf{v}^h) + (2\nu\mathbb{D}(\mathbf{u}^h), \mathbb{D}(\mathbf{v}^h)) + ((\mathbf{u}^h \cdot \nabla)\mathbf{u}^h, \mathbf{v}^h) & & \\ -(p^h, \nabla \cdot \mathbf{v}^h) + (\nu_T(\mathbb{D}(\mathbf{u}^h) - \mathbb{G}^H), \mathbb{D}(\mathbf{v}^h)) &= (\mathbf{f}, \mathbf{v}^h), & \forall \mathbf{v}^h \in V^h, \\ (q^h, \nabla \cdot \mathbf{u}^h) &= 0, & \forall q^h \in Q^h, \\ (\mathbb{D}(\mathbf{u}^h) - \mathbb{G}^H, \mathbb{L}^H) &= 0, & \forall \mathbb{L}^H \in L^H. \end{aligned} \quad (9)$$

The parameter $\nu_T \geq 0$ in (9) is the so-called turbulent viscosity. By definition, the large scales of $\mathbb{D}(\mathbf{u}^h)$ are defined by $\mathbb{G}^H$, the L^2-projection of $\mathbb{D}(\mathbf{u}^h)$ into L^H. Consequently, the resolved small scales are given by $\mathbb{D}(\mathbf{u}^h) - \mathbb{G}^H$. Thus, the additional viscous term $(\nu_T(\mathbb{D}(\mathbf{u}^h) - \mathbb{G}^H), \mathbb{D}(\mathbf{v}^h))$ in the momentum equation of (9) acts directly only on the small resolved scales.

The first parameter in (9) is the turbulent viscosity. In the simulations presented below, the static Smagorinsky model [17] $\nu_T = C_S\delta^2 \|\mathbb{D}(\bar{\mathbf{u}})\|_F$ is applied. The second parameter is the large scale space L^H. This space must be in some sense a coarser space than the finite element space for the velocity V^h, which represents all resolved scales. Provided that V^h is a higher order space, L^H can be chosen on the same grid as V^h, see [18] for a discussion of this topic. Numerical studies [19, 20, 21, 22] revealed that the choice of L^H possesses a significant influence on the computed results. For efficiency reasons, L^H must be a discontinuous finite element space, see [19]. The first implementations of the projection-based FEVMS method used a space L^H with the same polynomial degree on all mesh cells. Recently, in [23, 24], an approach for an adaptive, a posteriori choice of L^H was presented which allows to use different polynomial degrees in different mesh cells K. The aim of this approach consists in adjusting the local influence of the turbulence model according to the local turbulent character of the flow by an appropriate local scale separation.

As inf-sup stable pair of velocity-pressure finite element spaces, the pair Q_2/P_1^{disc} is used. This finite element is a popular choice [25] as it combines a high accuracy and the possibility of solving the arising saddle point problems efficiently [26, 27]. The adaptive FEVMS uses the following measure of the small resolved scales as indicator of the amount of local turbulence

$$\eta_K = \frac{\|\mathbb{G}^H - \mathbb{D}(\mathbf{u}^h)\|_{L^2(K)}}{\|1\|_{L^2(K)}} = \frac{\|\mathbb{G}^H - \mathbb{D}(\mathbf{u}^h)\|_{L^2(K)}}{|K|^{1/2}}, \quad K \in \mathcal{T}^h,$$

where $\{K\}$ are the mesh cells of the triangulation $\mathcal{T}^h$. This indicator assumes that in regions with high turbulence, where the size of the unresolved scales is large, also the size of the adjacent class of scales (the small resolved scales) is large. The local contributions η_K are compared to the arithmetic mean $\overline{\eta}$ of the local indicators in order to decide which local space $L^H(K)$ is chosen: given $0 \le C_1 \le C_2 \le C_3$, define $\eta := \eta_K/\overline{\eta}$, then

1. for cells K with $\eta \le C_1$: $L^H(K) = P_2^{\mathrm{disc}}(K)$, $\nu_T(K) = 0$,
2. for cells K with $C_1 < \eta \le C_2$: $L^H(K) = P_1^{\mathrm{disc}}(K)$,
3. for cells K with $C_2 < \eta \le C_3$: $L^H(K) = P_0(K)$,
4. for cells K with $C_3 < \eta$: $L^H(K) = P_{00}(K)$.

The space $P_{00}(K)$ consists only of the zero tensor. In this case, the turbulence model is applied locally to all resolved scales. In the simulations presented below, the space L^H is updated at each discrete time. The parameters in the adaptive choice of the large scale space are chosen to be $C_1 = 0.3$, $C_2 = 1$, and $C_3 = 2$.

4 Numerical Methods for Equation (3) for the DSD

4.1 Obtaining an Inlet Boundary Condition

Measurements were performed in the nodes of a grid with the y-z coordinates $(-0.225, -0.215, \ldots, 0.225)$ m × $(-0.18, -0.17, \ldots, 0)$ m at the plane $x = 0$ m for the obtaining boundary conditions at the inlet and at the plane $x = 0.4$ m for the comparison of experimental data and numerical results. Experimental data for the droplets and for the standard deviation, both in (no./cm^3), are available for the diameters $1\ \mu$m$, 3\ \mu$m$, \ldots, 171\ \mu$m. In [12], a way is described for converting the experimental data in values for the DSD at the grid points in (no./m^4). This way is used in the numerical simulations presented below.

An interpolation of the DSD at the inlet is necessary if a grid for the internal coordinate is used whose nodes do not coincide with the measurement points with respect to the internal coordinate. In the simulations presented below, two kinds of interpolation are used. The first one is the continuous piecewise linear interpolation between the given values at the diameters for which measurement data are available, both for the DSD and the standard deviation.

A second way is the use of a prescribed form of the DSD which obeys a certain function. The experimental data suggest a log-normal interpolation of the form

$$f_{\mathrm{in,exp}}(\mathbf{x}, d; \sigma, \mu, c_f) = \frac{c_f}{d\sigma\sqrt{2\pi}} e^{-\frac{(\ln d - \mu)^2}{\sigma^2}}, \ d > 0, \tag{10}$$

with the parameters σ, μ and c_f. Exemplary, Fig. 1 shows a representative result for the approximation in an arbitrary grid point at the inlet plane. The computation of the parameters in (10) was performed with a damped Gauss–Newton method for each grid point at the inlet in a pre-processing step. In the simulations presented below, all values for diameters larger or equal than 125 μm are set to be zero. This approach prevents the creation of nonphysically large droplets which else would arise in the evaluation of the aggregation terms. The standard deviation at the inlet is approximated with the ansatz (10).

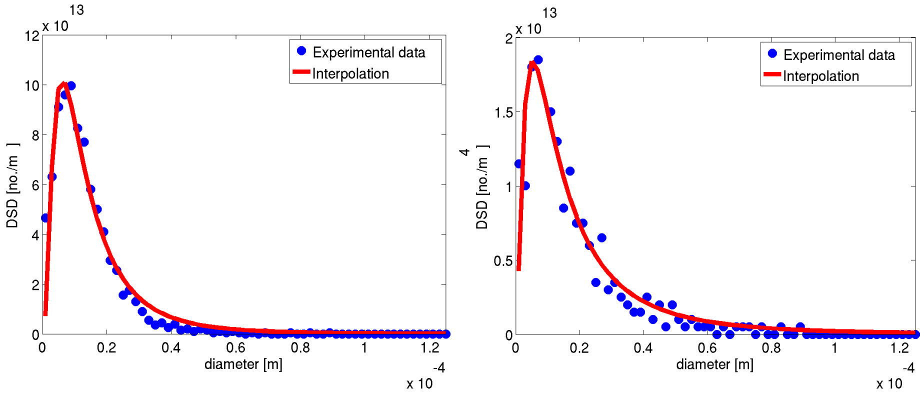

Figure 1: Log-normal interpolation of the DSD (left) and the standard deviation (right) at the inlet.

4.2 Discretization of the Left Hand Side

Two finite difference methods in combination with explicit time stepping schemes are considered. The tensor-product form of the 4D domain allows an easy application of finite difference methods. Finite element methods are much more flexible with respect to the form of the domain and therefore they are of great interest for many applications. A finite element method in combination with an implicit time stepping scheme will be considered, which has been proved to be among the best performing finite element methods for transport-dominated equations in recent studies. For this method, a standard realization and a more efficient non-standard realization are included in the studies presented below. This section describes the methods in some detail. The current discrete time is denoted by t_n, the length of the time step by Δt, and functions at t_n possess the subscript n.

The Forward Euler Finite Difference Upwind (FWE-UPW) Method

The Forward Euler scheme applied to (3) has the form: Find f_{n+1} such that

$$f_{n+1} = f_n - \Delta t \left(\frac{\partial}{\partial d} \left(\frac{a}{d} f_n \right) + \mathbf{u}_{\text{drop,n+1}} \cdot \nabla f_n - A_{+,n} - A_{-,n} \right). \qquad (11)$$

Note that the velocity field at t_{n+1} is used, also in the evaluation of the aggregation kernel in $A_{+,n}$ and $A_{-,n}$. For the sake of stability, the convective terms at the right hand side of (3) require a special discretization. Consider a node (x_i, y_i, z_i, d_i), then the (simple) upwind discretization approximates, e.g., the term with the x-derivative in the form

$$\Big(\mathbf{u}_1\partial_x f_n\Big)(x_i,y_i,z_i,d_i) \approx \begin{cases} \dfrac{\big(\mathbf{u}_1 f_n\big)(x_i,y_i,z_i,d_i)-\big(\mathbf{u}_1 f_n\big)(x_{i-1},y_i,z_i,d_i)}{x_i-x_{i-1}} & \text{if } \mathbf{u}_1(x_i,y_i,z_i,d_i)\ge 0 \\ \dfrac{\big(\mathbf{u}_1 f_n\big)(x_{i+1},y_i,z_i,d_i)-\big(\mathbf{u}_1 f_n\big)(x_i,y_i,z_i,d_i)}{x_{i+1}-x_i} & \text{if } \mathbf{u}_1(x_i,y_i,z_i,d_i)< 0 \end{cases}.$$

Here $\mathbf{u}_1$ denotes the first component of $\mathbf{u}_{\text{drop,n+1}}$, and (x_{i-1},y_i,z_i,d_i), (x_{i+1},y_i,z_i,d_i) are the neighbor nodes of (x_i,y_i,z_i,d_i) with respect to the x-coordinate.

The properties of the FWE-UPW scheme are well known. This scheme is of first order in space and time, its solutions do not possess spurious oscillations but layers are strongly smeared.

In [12], also a backward Euler scheme

$$f_{n+1}+\Delta t\left(\frac{\partial}{\partial d}\left(\frac{a}{d}f_{n+1}\right)+\mathbf{u}_{\text{drop,n+1}}\cdot\nabla f_{n+1}\right)=f_n+\Delta t\left(A_{+,n}+A_{-,n}\right)$$

with upwind discretization was studied. The results obtained with this scheme showed only very slight differences to the results from FWE-UPW but the simulations using the backward Euler scheme were more time-consuming. For this reason, the backward Euler scheme is not included in the studies presented below.

A Total Variation Diminishing Essentially Non-Oscillatory (TVD-ENO) Finite Difference Method

A serious drawback of FWE-UPW is its low accuracy. However, many higher order methods in time and space possess the disadvantage that the computed solutions show non-negligible spurious oscillations. There exist only few exceptions. One of them is the combination of a TVD scheme in time and an ENO finite difference scheme in space. In general, the appearance of spurious oscillations cannot be excluded with this method. But if such oscillations appear, they are generally small. In the numerical studies presented below, the combination of a third order TVD Runge–Kutta scheme and a third order ENO scheme is included.

Write equation (3) in the generic form $\partial f/\partial t = F(t,f)$. Then, an optimal third order TVD Runge–Kutta method has the form [28]

$$\begin{aligned} k_1 &= F(t_n,f_n), \\ k_2 &= F(t_n+\Delta t, f_n+\Delta t k_1), \\ k_3 &= F\left(t_n+\frac{\Delta t}{2}, f_n+\frac{\Delta t}{4}k_1+\frac{\Delta t}{4}k_2\right), \\ f_{n+1} &= f_n+\Delta t\left(\frac{k_1}{6}+\frac{k_2}{6}+\frac{4k_3}{6}\right). \end{aligned} \tag{12}$$

The first order derivatives on the right hand side of (12) are approximated by finite differences. The construction of a higher order scheme than simple upwinding requires the use of higher order information. The basic idea of ENO schemes consists in constructing several interpolation polynomials of a certain order and using as approximation the smoothest polynomial. Here, smoothness is measured in terms of the absolute value of the second order divided differences, which is proportional to the curvature.

Consider for simplicity of notation a one-dimensional situation. For a third order ENO scheme, the seven nodes $x_{i-3} < x_{i-2} < \ldots < x_{i+3}$ are used. Let $P_1(x)$ be the polynomial that interpolates the function f at the nodes $\{x_i, x_{i+1}, x_{i+2}, x_{i+3}\}$, $P_2(x)$

be the polynomial based on the nodes $\{x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$, $P_3(x)$ be the polynomial based on the nodes $\{x_{i-2}, x_{i-1}, x_i, x_{i+1}\}$, and finally $P_4(x)$ be the polynomial based on the nodes $\{x_{i-3}, x_{i-2}, x_{i-1}, x_i\}$. If necessary, Dirichlet boundary conditions are extended off the domain to define values for the interpolations. Then $f_x(t_n, x_i)$ will be approximated by one of the values $(P_j)_x(x_i)$, $j = 1, \cdots, 4$. Depending on the direction of the convection at x_i, one of the polynomials $P_1(x)$ and $P_4(x)$ is not needed. Hence, this scheme has a stencil with the six nodes $\{x_{i-3}, \ldots, x_{i+2}\}$ or $\{x_{i-2}, \ldots, x_{i+3}\}$.

Let the convection at x_i be non-negative. The ENO strategy tries to find the smoothest approximation. As first smoothness indicator, a quantity is compared which is proportional to the absolute value of the second derivative of the second order polynomials through the nodes $\{x_{i-2}, x_{i-1}, x_i\}$ and $\{x_{i-1}, x_i, x_{i+1}\}$, respectively. Let the interpolation polynomial through $\{x_{i-1}, x_i, x_{i+1}\}$ be the smoother one. Then, in the second step, a quantity is compared which is proportional to the absolute value of the third derivative at x_i of the polynomials $P_2(x)$ and $P_3(x)$. The smaller value gives the index j which defines the approximation $(P_j)_x(x_i)$ of $f_x(t_n, x_i)$. All other cases are treated in the same way, see [29] for a detailed algorithm.

A Linear Crank–Nicolson Finite Element Flux-Corrected Transport (CN-FCT) Method

A linear finite element FCT scheme in combination with the Crank–Nicolson scheme has been shown the best ratio of accuracy and efficiency in numerical studies of stabilized finite element methods for transport-dominated problems in [30, 31]. For this reason, two variants of this method will be included in the numerical studies. This section describes the basic ideas of this method and a standard realization.

Consider a continuous 4-linear finite element space Q_1 with the basis $\{\varphi_i\}_{i=1}^N$. The starting point of CN-FCT is the discretization of (3) with the Crank-Nicolson scheme in time and the Galerkin finite element method in space. This step leads to an algebraic equation of the form

$$\left(M_C + \frac{\Delta t}{2} A\right) \mathbf{f}_{n+1} = \left(M_C - \frac{\Delta t}{2} A\right) \mathbf{f}_n + \frac{\Delta t}{2} \left(\mathbf{a}_n + \mathbf{a}_{n+1}\right). \qquad (13)$$

Here, $(M_C)_{ij} = (\varphi_j, \varphi_i)_{i,j=1}^N$ is the consistent mass matrix and A is the matrix of the Galerkin discretization of the transport terms in (3). In our implementation, A is assembled with $\mathbf{u}_{\mathrm{drop},n+1}$. The DSD is a Q_1 finite element function and $\mathbf{f}_n = (f_{1,n}, \ldots, f_{N,n})^T$ denotes the vector of nodal unknowns, i.e. $f(t_n, \mathbf{x}, d) = \sum_{j=1}^N f_{j,n} \varphi_j(\mathbf{x}, d)$. The values for the aggregation are computed in the vertices of the mesh cells. For this reason, the aggregation can be interpreted also as a Q_1 finite element function. That means, there is a representation of the form $A_{+,n} + A_{-,n} = \sum_{j=1}^N \tilde{a}_{j,n} \varphi_j(\mathbf{x}, d)$. With this representation, the finite element right hand side becomes

$$(A_{+,n} + A_{-,n}, \varphi_i)_{i=1}^N = \left(\sum_{j=1}^N \tilde{a}_{j,n} (\varphi_j, \varphi_i)\right)_{i=1}^N =: \mathbf{a}_n.$$

Discretization (13) is instable for transport-dominated problems. The FCT methodology changes in the first step the matrix to a stable matrix. This step introduces a lot of diffusion. In the second step, this diffusion is removed where it is not needed by modifying the right hand side of (13). For the definition of the stable matrix, let

$$\begin{aligned} L &= A + D, \\ D &= \begin{cases} -\max\{0, a_{ij}, a_{ji}\} = \min\{0, -a_{ij}, -a_{ji}\} & \text{for } i \neq j, \\ -\displaystyle\sum_{j=1, j\neq i}^{N} d_{ij}, & \text{for } i = j, \end{cases} \\ M_L &= \operatorname{diag}(m_i), \quad m_i = \sum_{j=1}^{N} m_{ij}. \end{aligned}$$

The diagonal matrix M_L is called lumped mass matrix. The over-diffusive discretization after the first step has the form

$$\left(M_L + \frac{\Delta t}{2} L\right) \mathbf{f}_{n+1} = \left(M_L - \frac{\Delta t}{2} L\right) \mathbf{f}_n + \frac{\Delta t}{2} \left(\mathbf{a}_n + \mathbf{a}_{n+1}\right). \tag{14}$$

The system matrix possesses properties of an M-matrix.

The linear FCT scheme adds to the right hand side of (14) an anti-diffusive contribution $\mathbf{a}^*(\mathbf{f}_n)$. This term is defined with the help of the residual vector, which is the difference of (14) and (13). A straightforward calculations leads to the following representation of this vector

$$\mathbf{r} = (M_L - M_C)(\mathbf{f}_{n+1} - \mathbf{f}_n) + \frac{\Delta t}{2} D(\mathbf{f}_{n+1} + \mathbf{f}_n).$$

Then, the modification of the right hand side of (14) is given by

$$\mathbf{a}^*(\mathbf{f}_n) = \left(\sum_{j=1}^{N} \alpha_{ij} r_{ij}\right)_{i=1}^{N}, \tag{15}$$

with the weights $\alpha_{ij} \in [0,1]$. FCT methods determine these weights in such a way that they become close to one in smooth regions (this recovers the Galerkin finite element method) and that they are close to zero at layers (this recovers the stable low order scheme). The so-called fluxes are defined by a decomposition of the residual vector $r_i = \sum_{j=1}^{N} r_{ij}$ with

$$\begin{aligned} r_{ij} &= \Big[m_{ij}\left[(f_{n+1,i} - f_{n+1,j}) - (f_{n,i} - f_{n,j})\right] \\ &\quad - \frac{\Delta t}{2} d_{ij}\left[(f_{n+1,i} - f_{n+1,j}) + (f_{n,i} - f_{n,j})\right]\Big] \qquad (16) \\ &= 2m_{ij}\left[(f_{n+1/2,i} - f_{n,i}) - (f_{n+1/2,j} - f_{n,j})\right] - \Delta t d_{ij}\left[(f_{n+1/2,i} - f_{n+1/2,j})\right], \end{aligned}$$

where $\mathbf{f}_{n+1/2} = (\mathbf{f}_{n+1} + \mathbf{f}_n)/2$.

The linear finite element FCT scheme, which will be used, is a special case of one of the schemes presented in [32]. In this scheme, the vector $\mathbf{f}_{n+1/2}$ in the flux r_{ij} is replaced by an approximation that is computed with an explicit scheme. To this end, the forward Euler scheme with the length of the time step $\Delta t/2$ is used, which is applied in combination with the low order method (14). The result obtained with this scheme is inserted into (17). Note that this predictor step is performed with an explicit method such that a CFL condition applies, see [32].

The computation of the weights uses Zalesak's algorithm [33]. This algorithm is described and discussed also in [34]. In the simulations presented below, the version of Zalesak's algorithm is used which is given in [29].

The linear system (14) with the modification (15) of the right hand side is solved with a BiCGSTAB method with an SSOR preconditioner [35]. Very few iterations were always necessary for decreasing the Euclidean norm of the residual vector below 10^{-12}.

The matrix A depends on the time-dependent velocity field $\mathbf{u}_{\text{drop}}$. Hence, a new matrix is needed in every discrete time. The standard approach consists in assembling this matrix over and over again using the bilinear form

$$\sum_{k=1}^{4}\sum_{j=1}^{N}\Big(\mathbf{v}_k\partial_k\varphi_j,\varphi_i\Big) \tag{17}$$

for the matrix entry (i,j). Here, $\mathbf{v}_k$ stands for the convection, i.e. $\mathbf{u}_{\text{drop}}$ or the growth. In our implementation, a two-point Gaussian quadrature rule in each dimension is used to keep the quadrature errors sufficiently small. This quadrature rule possesses 16 quadrature points in each 4D mesh cell. In the simulations, it turned out that the assembling of A took much longer than all other operations which were performed in the simulation of the DSD.

A Linear Crank–Nicolson Group Finite Element FCT (CN-GFCT) Method

The most serious drawback of CN-FCT is the computational cost which arises in the assembling of the matrices. This section presents an alternative approach to obtain the matrices, which is much more economical and which leads to very similar results. It is called group finite element method and it was proposed, e.g., in [36]. The group finite element method was applied in combination with FCT schemes already, e.g., in [32, 34].

The starting point of the group finite element method is the divergence formulation of the transport term of (3)

$$\mathbf{u}_{\text{drop}}\cdot\nabla f+\frac{\partial}{\partial d}\left(\frac{a}{d}f\right)=:\nabla\cdot(\mathbf{v}f), \tag{18}$$

where for simplicity of notation the fourth component of $\mathbf{v}$ is the growth rate a/d and the divergence is defined in four dimensions. The velocity $\mathbf{u}_{\text{drop}}$ is considered to be divergence-free. In the simulations, $\mathbf{u}_{\text{drop}}$ is the sum of interpolations of experimental data and a finite element approximation of a turbulent flow field. Hence, the divergence-free constraint will be violated somewhat but it can be assumed that this violation is small.

The basic idea of the group finite element method consists in not only using f as variable in (3) but to use for (18) the group $(\mathbf{v}f)$ as finite element variable $(\mathbf{v}f)^h=\sum_{j=1}^{N}(\mathbf{v}_j f_j)\varphi_j$, where $\mathbf{v}_j$ are the degrees of freedom of the convection and f_j are the unknown degrees of freedom of the DSD. Inserting this ansatz in (18) gives

$$(\mathbf{v}f)\approx(\mathbf{v}f)^h=\sum_{k=1}^{4}\left(\sum_{j=1}^{N}(\partial_k\varphi_j,\varphi_i)(\mathbf{v}_j)_k f_j\right). \tag{19}$$

The matrices $C_k=(\partial_k\varphi_j,\varphi_i)_{i,j=1}^N$, $k=1,\ldots,4$, have to be assembled only once. To obtain the approximation of the transport matrices from (19), each row of C_k has to be multiplied with the k-th component of the convection.

Instead of applying numerical quadrature, an approximation of the transport matrices is obtained by some multiplications of pre-computed matrices and the current convection vectors in the group finite element method. Comparing (17) with (19), one can see that in the group finite element method the value of the convection at the node j is used instead of the values at the quadrature points around the node j in the standard approach.

4.3 Computation of the Aggregation Term

Let $0 = d_0 < d_1 = d_{\min} < \ldots < d_M = d_{\max}$ be the grid points with respect to the internal coordinate. As explained above, the grid point d_0 is introduced for the application of boundary conditions with respect of the internal coordinate and $f(d)$ vanishes in (d_0, d_1). The DSD is assumed to be continuous within each interval.

Standard Numerical Quadrature

The simplest approach consists in using quadrature formulas for the evaluation of the integrals (4) – (7). In our simulations, Gaussian quadrature formulas are used with three, four, or five quadrature points for each interval $[d_i, d_{i+1}]$. The results were always rather similar.

The use of standard quadrature formulas possesses disadvantages. The source term is of convolution type and the application of a time-consuming double loop over the intervals of the internal coordinates is necessary. If the diameter formulation (6) is used, then the term in the integral is almost singular for d' close to d. Because the DSD f is set to zero for diameters smaller than $d_{\min}$, it is prevented that the term in the integral becomes in fact singular.

Pre-Computation of Certain Integrals

The basic idea of this approximation of the integrals consists in approximating the DSD within each interval $[d_i, d_{i+1}]$ by a single value. Then, this value of the DSD can be written outside the integrals and integration of only the kernel is necessary. Since the kernel is known, these integrals can be computed in a pre-processing step.

This approach will be illustrated exemplary for the sink term (7) where the dependency of the DSD on time and space is neglected for simplicity of presentation. The sink term at the diameter d_j, $j \in \{1, \ldots, M\}$, is approximated as follows

$$\begin{aligned}
f(d_j) \int_{d_{\min}}^{d_{\max}} \kappa_{\mathrm{agg}}(d_j, d) f(d') \, dd' &= f(d_j) \sum_{i=1}^{M-1} \int_{d_i}^{d_{i+1}} \kappa_{\mathrm{agg}}(d_j, d') f(d') \, dd' \\
&\approx \quad f(d_j) \sum_{i=1}^{M-1} \frac{f(d_{i+1}) + f(d_i)}{2} \int_{d_i}^{d_{i+1}} \kappa_{\mathrm{agg}}(d_j, d') \, dd'.
\end{aligned}$$

For the integral of the kernel holds

$$\begin{aligned}
\int_{d_i}^{d_{i+1}} \kappa_{\mathrm{agg}}(d_j, d') \, dd' &= C_{\mathrm{brown}} \frac{2 k_B T}{3\mu} \int_{d_i}^{d_{i+1}} (d + d') \left(\frac{1}{d} + \frac{1}{d'} \right) \, dd' \\
&\quad + C_{\mathrm{shear}} \sqrt{2 \nabla \mathbf{u}_{\mathrm{drop}} : \nabla \mathbf{u}_{\mathrm{drop}}} \int_{d_i}^{d_{i+1}} (d + d')^3 \, dd',
\end{aligned}$$

$i = 1, \ldots, M-1$. Since the grid with respect to the internal coordinate is given, the integrals on the right hand side can be computed in a pre-processing step. A similar approach can be performed for the source term.

For the pre-computation of the integrals, the package MAPLE was used because it was not possible to evaluate all integrals analytically. The numerical computation of the integrals in MAPLE was performed with the option to be exact for 14 digits.

A Mass-Conserving Method for Computing Convolution Integrals

The method, which was developed in [37, 38, 39], will be applied to the volume formulations of the aggregation terms (4) and (5). One of its main properties is the conservation of mass. For this feature, the use of special grids with respect to the internal coordinate is necessary. This method was used in the simulations of the urea synthesis in [40].

In the studied situations, much more small droplets are contained in the flow than large droplets. For this reason, it might be advantageous to use a grid for the DSD, that is refined towards the smallest droplets. The requirement of the used method is that this grid is locally uniform with respect to the internal coordinate, which is here the volume of the droplets. A grid with the following property fulfills this requirement: $|V_{i+1} - V_i|/|V_i - V_{i-1}| = l$, $l \in \{1, 2\}$, $i = 1, \dots, M-1$, where $|\cdot|$ denotes the length of an interval. On such a grid, a piecewise linear ansatz space $\mathcal{S}$ for the DSD is chosen.

A property of the kernel, which is exploited in the method, is its separability, i.e. the aggregation kernel can be written in the form $\kappa_{\mathrm{agg}}(V, V') = \sum_{i=1}^{k} a_i(V) b_i(V')$. Then, the aggregation term becomes

$$\begin{aligned}
&A(V) = A_+(V) + A_-(V) = \\
&= \frac{1}{2}\int_0^V \kappa_{\mathrm{agg}}(V - V', V') f_V(V - V') f_V(V')\, dV' - f_V(V)\int_0^{V_{\max}} \kappa_{\mathrm{agg}}(V, V') f_V(V')\, dV' \\
&= \sum_{i=1}^{k}\left[\frac{1}{2}\int_0^V a_i(V - V') b_i(V') f_V(V - V') f_V(V') dV' \right. \\
&\qquad \left. - f_V(V) a_i(V)\int_0^{V_{\max}} b_i(V') f_V(V')\, dV'\right].
\end{aligned}$$

The difficult part is the evaluation of the source term since it is the sum of convolutions $\varphi_i * \psi_i$, where $\varphi_i = a_i f$ and $\psi_i = b_i f$. The functions $a_i(V)$ and $b_i(V)$ are approximated by piecewise constants on the same grid as $f_V(V)$. Legendre polynomials are used as an orthonormal basis of $\mathcal{S}$. Special properties of these polynomials allow the computation of the convolution with the complexity $\mathcal{O}(M \log M)$. Essentially, some discrete convolutions with the fast Fourier transform have to be computed. For details of the rather involved algorithm, see [37, 38, 39].

The exact convolution $\omega_{\mathrm{exact}} = \sum_{i=1}^{M} \varphi_i * \psi_i$ does not belong to the space $\mathcal{S}$. In the simulations presented below, the L^2 projection ω_{comp} of ω_{exact} into the ansatz space is used. Since $\omega_{\mathrm{exact}} - \omega_{\mathrm{comp}}$ is L^2 orthogonal to all piecewise linear functions, one gets for all intervals $[V_i, V_{i+1}]$

$$\mathrm{mass}_i(\omega_{\mathrm{exact}}) = \int_{V_i}^{V_{i+1}} V' \omega_{\mathrm{exact}}(V')\, dV' = \int_{V_i}^{V_{i+1}} V' \omega_{\mathrm{comp}}(V')\, dV' = \mathrm{mass}_i(\omega_{\mathrm{comp}}),$$

i.e. the mass (volume) is locally preserved. However, a change in total mass might occur from the fact that the support of the convolution is larger than the support of the convolved functions. In the case of aggregation, only non-negative contributions will be neglected by not considering the complete support of the convolution such that the mass will always decrease. To avoid the decrease of mass, the following correction to the aggregation term is applied

$$A(V) := A(V) - \mathrm{mass}(A(V)) \frac{2}{V_{\max}^2 - V_{\min}^2}.$$

Then, although local mass conservation is violated, the total mass of the computed aggregation term is zero, which is in accordance with the physics. Other forms of corrections to obtain mass conservation are possible. But the presented way turned out to be the best in our studies, see also the discussion of this topic in [40].

5 Numerical Studies

The numerical studies presented below were performed with the code MooNMD [41]. They are based on non-dimensional equations using the following reference values

$$l_\infty = 1 \text{ m}, \ u_\infty = 1 \text{ m/s}, \ t_\infty = \frac{l_\infty}{u_\infty} \text{ s}, \ p_\infty = \rho u_\infty^2 \text{ Pa}, \ f_\infty = 10^{12} \text{ no./m}^4.$$

The minimal diameter of the droplets is set to be $d_{\min} = 10^{-6}$ m, which corresponds to the smallest droplets in the experiments.

A hexahedral grid consisting of $51 \times 46 \times 19$ nodes, which is equi-distant in each direction, was used for the triangulation of Ω. With this grid, the positions of the measurement points were located at nodes. For the internal coordinate, two types of grids are considered. The first one is essentially uniform with the 89 nodes at $(0, 1, 3, 5, \ldots, 175)$ μm. Hence, the nodes coincide with the diameters for which experimental data are available. The number of nodes of the 4D grid for the DSD is 3 967 086, which corresponds to the number of degrees of freedom for the Q_1 finite element method. For this grid, the value $d_\infty = d_{\max} = 175 \cdot 10^{-6}$ m is used. The second grids is refined towards small diameters. It consists of 94 nodes and the local refinement is performed as described in the presentation of the mass-conserving integration scheme. Because quite long intervals are used for large particles, the simulations are performed on a somewhat larger domain with respect to the internal coordinate using $d_\infty = d_{\max} = 249 \cdot 10^{-6}$ m. With the used grid, $[124.5 \cdot 10^{-6}, 249 \cdot 10^{-6}]$ m is just covered with three intervals. This grid possesses 4 189 956 nodes.

All simulations are performed in the time interval $[0, 1]$ s and the data are averaged in $[0.5, 1]$ s. The length of the time step is set to be $\Delta t = 10^{-3}$ s.

A snap shot of the flow field is presented in Figure 2. The Q_2/P_1^{disc} leads to 1 020 201 velocity degrees of freedom and to 162 000 pressure degrees of freedom. The number of degrees of freedom for the projection space changes during the adaption process. It was around 125 000 for each component of the symmetric tensor. Because of the imposed noise at the inlet Γ_{in}, the temporal variation of the minimal and maximal value of the stream-wise velocity is quite large. The mean value of the stream-wise velocity shows that the average residence time of a droplet in the measurement section is less than 0.25 s. Hence, a fully developed process is present in the time interval for computing the averaged data.

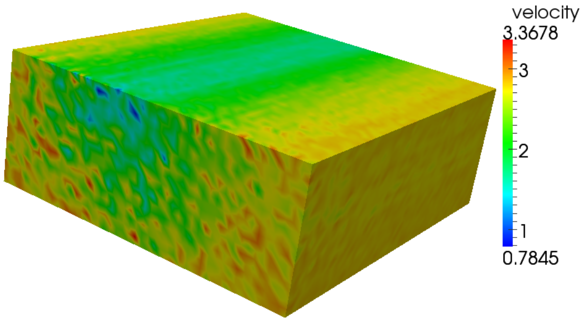

Figure 2: Snap shot of the stream-wise component of the velocity field.

Results for different discretizations for the temporal derivative and the spatial derivatives of the transport terms in Equation (3) for the DSD are presented in Figure 3 and Table 1. The curves for the finite difference schemes FWE-UPW and TVD-ENO fit very well to the time-space-averaged experimental data. This is not surprising since the fit of the model parameters C_{brown} and C_{shear} in the kernel (8) was performed with FWE-UPW in [12]. The results of CN-FCT and CN-GFCT are rather similar. They predict too few small droplets. This is a hint that the parameter C_{brown} is too large. In fact, we could observe that with the parameter $C_{\text{brown}} = 1 \cdot 10^6$ a better agreement of the results obtained with CN-FCT and CN-GFCT to the experimental data is obtained. These observation demonstrates the influence of numerical methods on the calibration of model parameters.

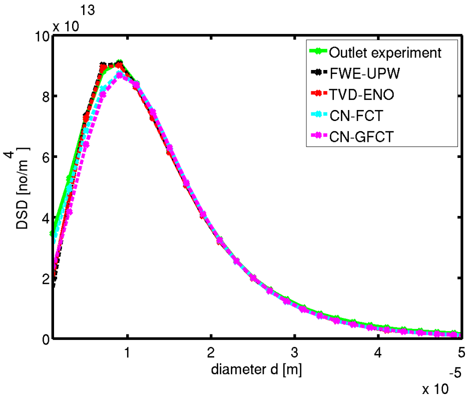

Figure 3: Comparison of discretizations for temporal and transport-dominated spatial derivatives, uniform grid for DSD, pre-computed integrals for aggregation.

Table 1: Mode (value where maximum is taken), maximum, and computing time per time step for different discretizations of the temporal and spatial derivatives, uniform grid for DSD, pre-computed integrals for aggregation.

method	mode (m)	maximum (no./m^4)	comp. time (s)
experimental data outlet	$9 \cdot 10^{-6}$	$9.0987 \cdot 10^{13}$	
FWE-UPW	$9 \cdot 10^{-6}$	$9.0641 \cdot 10^{13}$	127
TVD-ENO	$9 \cdot 10^{-6}$	$9.0006 \cdot 10^{13}$	140
CN-FCT	$9 \cdot 10^{-6}$	$8.7379 \cdot 10^{13}$	995
CN-GFCT	$9 \cdot 10^{-6}$	$8.6794 \cdot 10^{13}$	322

Figure 4 and Table 2 present the effect on the time-space-averaged data of using a log-normal interpolation of the inlet boundary condition for the DSD in each measurement point. A shift of the peak towards smaller droplets can be observed. This shift is still present in the computed results at the outlet, see Figure 5. The approximations of the DSD for droplets with diameter in $[10, 25]$ μm is considerably worse for the log-normal interpolation compared with using the data and nodes of the experiment directly. Also in this case, we could observe that it is possible to obtain results for the log-normal interpolation that are closer to the experiments by using different model parameters C_{brown} and C_{shear}.

Table 2: Mode and maximum for different approximations of the space-averaged inlet boundary condition for the DSD.

method	mode (m)	maximum (no./m^4)
experimental data inlet	$7 \cdot 10^{-6}$	$1.1560 \cdot 10^{14}$
log-normal interpolation, adaptive grid	$6.0688 \cdot 10^{-6}$	$1.1767 \cdot 10^{14}$

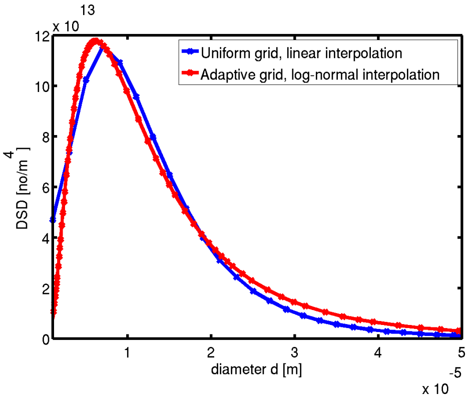

Figure 4: Time-space averaged inlet condition for the DSD.

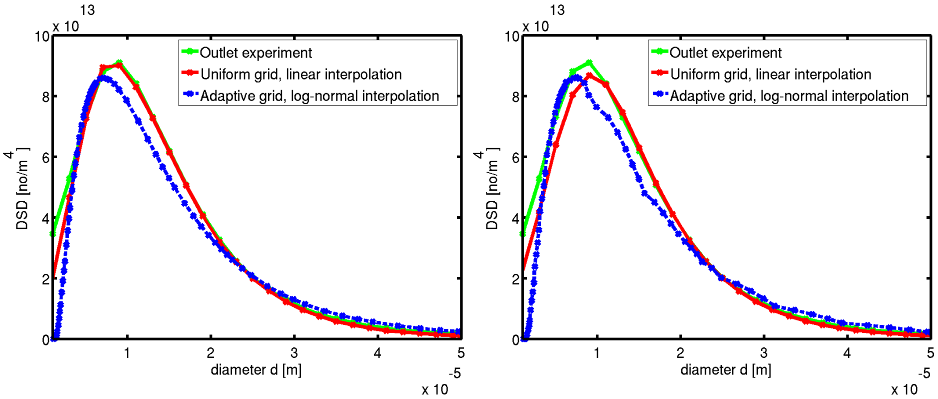

Figure 5: Comparison of different approximations of the inlet boundary condition for the DSD with TVD-ENO (left) and CN-GFCT (right), pre-computed integrals.

The effect of applying different methods for computing the aggregation integrals is studied in Figures 6, 7 and in Tables 3, 4. It can be seen that the choice of this method possesses only little influence on the time-space-averaged DSD at the outlet.

Table 3: Mode, maximum, and computing time per time step for different methods for computing the aggregation integrals, TVD-ENO, log-normal interpolation.

method	mode (m)	maximum (no./m^4)	comp. time (s)
Gauss3	$7.0697 \cdot 10^{-6}$	$8.6010 \cdot 10^{13}$	271
Gauss4	$7.0697 \cdot 10^{-6}$	$8.5887 \cdot 10^{13}$	309
Gauss5	$7.0697 \cdot 10^{-6}$	$8.6096 \cdot 10^{13}$	338
pre-computed integrals	$7.0697 \cdot 10^{-6}$	$8.5981 \cdot 10^{13}$	174
mass-conserving method	$7.0697 \cdot 10^{-6}$	$8.5434 \cdot 10^{13}$	215

Computing times per time step for different methods are given in Tables 1, 3, and 4. The simulation of the flow field takes around 100 s. As expected, the finite difference methods with explicit time stepping schemes are faster than the finite element methods with implicit temporal discretization. Among the first class, FWE-UPW is a little bit faster than TVD-ENO. CN-GFCT takes more than twice as long as TVD-ENO. The slowest scheme is CN-FCT, whose simulation time is

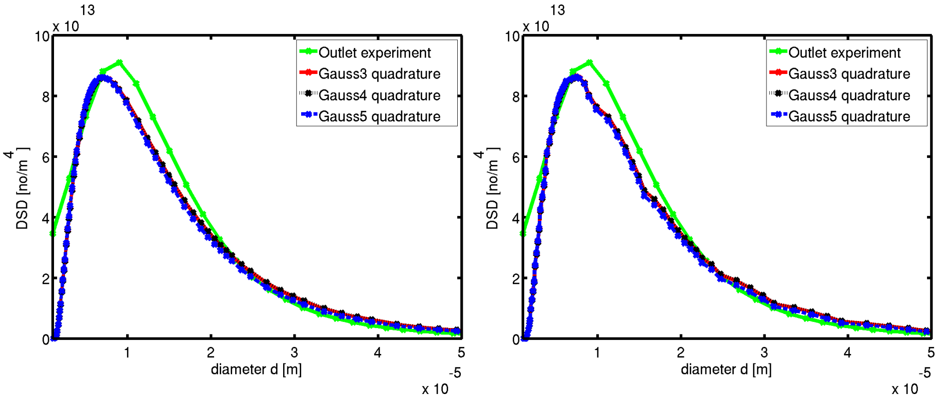

Figure 6: Comparison of Gaussian quadrature methods for computing the aggregation with TVD-ENO (left) and CN-GFCT (right), adaptive grid with log-normal interpolation for DSD.

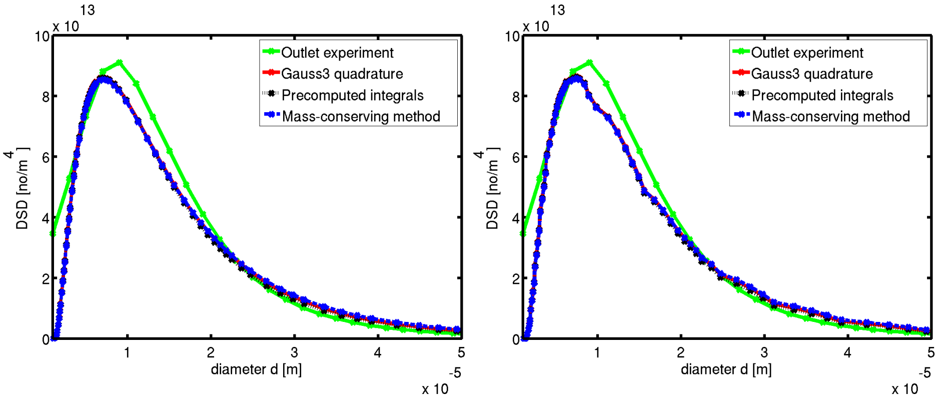

Figure 7: Comparison of different methods for computing the aggregation with TVD-ENO (left) and CN-GFCT (right), adaptive grid with log-normal interpolation for DSD.

three times longer than CN-GFCT. With respect to the method for computing the aggregation, it can be seen that Gaussian quadrature is the least efficient approach. The application of the method with pre-computed integrals is fastest.

6 Summary and Outlook

The numerical studies presented in Section 5 reveal that the use of different numerical methods for solving Equation (3) for the DSD might possess a non-negligible impact on the computed time-space-averaged DSD at the outlet. Unlike [9, 10], where more sensitive outputs of interest were studied, the changes are rather quantitative than qualitative. For the considered problem, the interpolation of the inlet boundary condition and the discretization of the temporal and transport-dominated first order spatial derivatives have the most significant influence on the numerical results. Different choices of the method for computing the aggregation integrals lead to very similar results. However, this situation might change if long-term simulations are performed where, e.g., the loss of mass in some methods might become crucial.

For the assessment of the methods, two situations will be distinguished. The

Table 4: Mode, maximum, and computing time per time step for different methods for computing the aggregation integrals, CN-GFCT, log-normal interpolation.

method	mode (m)	maximum (no./m^4)	comp. time (s)
Gauss3	$7.4425 \cdot 10^{-6}$	$8.6318 \cdot 10^{13}$	376
Gauss4	$7.4425 \cdot 10^{-6}$	$8.6165 \cdot 10^{13}$	414
Gauss5	$7.4425 \cdot 10^{-6}$	$8.6285 \cdot 10^{13}$	442
pre-computed integrals	$7.4425 \cdot 10^{-6}$	$8.6196 \cdot 10^{13}$	276
mass-conserving method	$7.4425 \cdot 10^{-6}$	$8.5768 \cdot 10^{13}$	303

first one is the case of a tensor-product domain Ω where finite difference methods can be applied easily. Then, from the point of view of efficiency, FWE-UPW and the use of pre-computed integrals for the aggregation are the best methods. But, because of their better mathematical properties, we would recommend the use of TVD-ENO and the mass-conserving quadrature method. Their overhead with respect to the fastest methods is modest. Note that the mass-conserving quadrature method requires special grids with respect to the internal coordinate. In the second situation, that the flow domain is not of tensor-product form, our method of choice for the discretization of the temporal and spatial derivatives will be CN-GFCT.

An interesting extension of the studies is the consideration of more sensitive outputs of interest. It can be expected that in such situations larger differences can be observed if different methods are used. The difficulty consists in assessing the computational results, see also [9, 10], since it will be hard to obtain experimental data for such outputs. Another important topic is the inclusion of moment-based methods into the studies since these methods are often used in practice. This topic will be pursued next in our future work.

References

[1] D. Ramkrishna. *Population Balances: Theory and Applications to Particulate Systems in Engineering.* Academic Press, San Diego, 2000.

[2] R.R. Rogers and M.K. Yau. *Cloud Physics.* Butterworth–Heinemann, 3rd edition, 1989.

[3] R.A. Shaw. Particle-turbulence interactions in athmospheric clouds. *Annu. Rev. Mech.*, 35:183 – 227, 2003.

[4] H.M. Hulburt and S. Katz. Some problems in particle technology – a statistical mechanical formulation. *Chem. Engrg. Sci.*, 19:555 – 574, 1964.

[5] R. McGraw. Description of aerosol dynamics by the quadrature method of moments. *Aerosol Science and Technology*, 27:255 – 265, 1997.

[6] D.L. Marchisio and R.O. Fox. Solution of population balance equations using the direct quadrature method of moments. *J. Aero. Sci.*, 36:43 – 73, 2005.

[7] V. John, I. Angelov, A. A. Öncül, and D. Thévenin. Techniques for the reconstruction of a distribution from a finite number of its moments. *Chem. Engrg. Sci.*, 62:2890 – 2904, 2007.

[8] L.G.M. de Souza, G. Janiga, V. John, and D. Thévenin. Reconstruction of a distribution from a finite number of moments with an adaptive spline-based algorithm. *Chem. Engrg. Sci.*, 65:2741 – 2750, 2010.

[9] V. John, M. Roland, T. Mitkova, K. Sundmacher, L. Tobiska, and A. Voigt. Simulations of population balance systems with one internal coordinate using finite element methods. *Chem. Engrg. Sci.*, 64:733 – 741, 2009.

[10] V. John and M. Roland. On the impact of the scheme for solving the higher dimensional equation in coupled population balance systems. *Int. J. Numer. Meth. Engng.*, 82:1450 – 1474, 2010.

[11] R. Bordás, T. Hagemeier, B. Wunderlich, and D. Thévenin. Droplet collisions and interaction with the turbulent flow within a two-phase wind tunnel. *Phys. Fluids*, 2011. accepted for publication.

[12] R. Bordás, V. John, E. Schmeyer, and D. Thévenin. Measurement and simulation of a droplet population in a turbulent flow field. Preprint 1590, Weierstrass Institute for Applied Analysis and Stochastics Berlin, 2011.

[13] A. Mersmann. *Crystallization Technology Handbook.* Marcel Dekker Inc., New York, 2nd edition, 2001.

[14] K.A. Berglund. Analysis and measurment of crystallization utilizing the population balance. In A. Myerson, editor, *Handbook of Industrial Crystallization.* Butterworth-Heinemann, Oxford, second edition, 2002.

[15] C. Lindenberg, J. Schöll, M. Vicum, and J. Brozio. L-glutamic acid precipitation: Agglomeration effects. *Crystal Growth & Design*, 8:224 – 234, 2008.

[16] M.V. Smoluchowski. Versuch einer mathematischen Theorie der Koagulationskinetik kolloider Lösungen. *Z. Phys. Chem.*, 92:129 – 168, 1917.

[17] J.S. Smagorinsky. General circulation experiments with the primitive equations. *Mon. Weather Review*, 91:99 – 164, 1963.

[18] V. John, S. Kaya, and W. Layton. A two-level variational multiscale method for convection-dominated convection-diffusion equations. *Comp. Meth. Appl. Math. Engrg.*, 195:4594 – 4603, 2006.

[19] V. John and S. Kaya. A finite element variational multiscale method for the Navier-Stokes equations. *SIAM J. Sci. Comp.*, 26:1485 – 1503, 2005.

[20] V. John and A. Kindl. Variants of projection-based finite element variational multiscale methods for the simulation of turbulent flows. *Int. J. Numer. Meth. Fluids*, 56:1321 – 1328, 2008.

[21] V. John and A. Kindl. Numerical studies of finite element variational multiscale methods for turbulent flow simulations. *Comput. Methods Appl. Mech. Engrg.*, 199:841 – 852, 2010.

[22] V. John and M. Roland. Simulations of the turbulent channel flow at $Re_\tau = 180$ with projection-based finite element variational multiscale methods. *Int. J. Numer. Meth. Fluids*, 55:407 – 429, 2007.

[23] V. John and A. Kindl. A variational multiscale method for turbulent flow simulation with adaptive large scale space. *J. Comput. Phys.*, 229:301 – 312, 2010.

[24] V. John, A. Kindl, and C. Suciu. Finite element LES and VMS methods on tetrahedral meshes. *J. Comput. Appl. Math.*, 233:3095 – 3102, 2010.

[25] P.M. Gresho and R.L. Sani. *Incompressible Flow and the Finite Element Method.* Wiley, Chichester, 2000.

[26] V. John and G. Matthies. Higher order finite element discretizations in a benchmark problem for incompressible flows. *Int. J. Num. Meth. Fluids*, 37:885 – 903, 2001.

[27] V. John. Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier-Stokes equations. *Int. J. Num. Meth. Fluids*, 40:775 – 798, 2002.

[28] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.*, 77:439 – 471, 1988.

[29] V. John and J. Novo. On (essentially) non-oscillatory discretizations of evolutionary convection-diffusion equations. Preprint, WIAS Berlin, 2011.

[30] V. John and E. Schmeyer. Stabilized finite element methods for time-dependent convection-diffusion-reaction equations. *Comput. Methods Appl. Mech. Engrg.*, 198:475 – 494, 2008.

[31] V. John and E. Schmeyer. On finite element methods for 3d time-dependent convection-diffusion-reaction equations with small diffusion. In A. Hegarty et al., editor, *BAIL 2008 – Boundary and Interior Layers*, number 69 in Lecture Notes in Computational Science and Engineering, pages 173 – 181. Springer, 2009.

[32] D. Kuzmin. Explicit and implicit FEM-FCT algorithms with flux linearization. *J. Comput. Phys.*, 228:2517 – 2534, 2009.

[33] S.T. Zalesak. Fully multi-dimensional flux corrected transport algorithms for fluid flow. *J. Comput. Phys.*, 31:335 – 362, 1979.

[34] D. Kuzmin and M. Möller. Algebraic flux correction I. Scalar conservation laws. In D. Kuzmin, R. Löhner, and S. Turek, editors, *Flux-Corrected Transport: Principles, Algorithms and Applications*, pages 155 – 206. Springer, 2005.

[35] Y. Saad. *Iterative methods for sparse linear systems.* SIAM Philadelphia, 2nd edition, 2003.

[36] C.A.J. Fletcher. The group finite element formulation. *Int. J. Numer. Methods Fluids*, 37:225 – 243, 1983.

[37] W. Hackbusch. On the efficient evaluation of coalescence integrals in population balance models. *Computing*, 78:145 – 159, 2006.

[38] W. Hackbusch. Approximation of coalescence integrals in population balance models with local mass conservation. *Numer. Math.*, 106:627 – 657, 2007.

[39] W. Hackbusch. Convolution of hp-functions on locally refined grids. *IMA Journal of Numerical Analysis*, 29:960 – 985, 2009.

[40] W. Hackbusch, V. John, A. Khachatryan, and C. Suciu. A numerical method for the simulation of an aggregation-driven population balance system. *Int. J. Numer. Methods Fluids*, 2011. in press.

[41] V. John and G. Matthies. MooNMD - a program package based on mapped finite element methods. *Comput. Visual. Sci.*, 6:163 – 170, 2004.