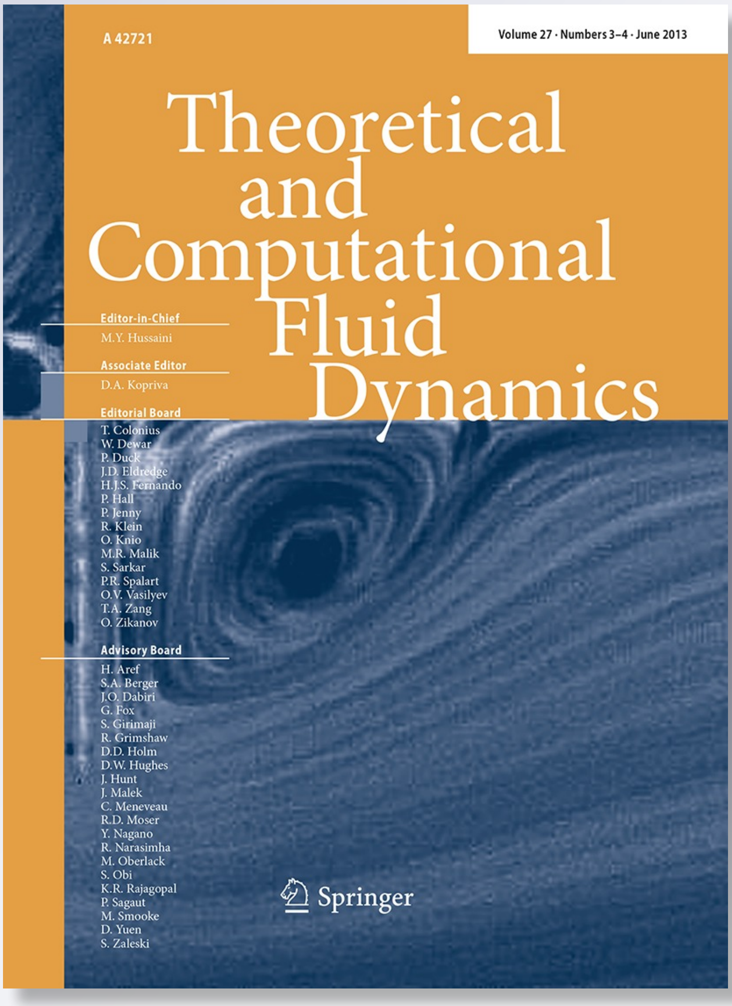
Numerical methods for the simulation of a coalescence-driven droplet size distribution

Róbert Bordás, Volker John, Ellen Schmeyer & Dominique Thévenin

Theoretical and Computational Fluid Dynamics

ISSN 0935-4964
Volume 27
Combined 3-4

Theor. Comput. Fluid Dyn. (2013)
27:253-271
DOI 10.1007/s00162-012-0275-9



Theor. Comput. Fluid Dyn. (2013) 27:253–271
DOI 10.1007/s00162-012-0275-9

ORIGINAL ARTICLE

**Róbert Bordás · Volker John · Ellen Schmeyer ·
Dominique Thévenin**

Numerical methods for the simulation of a coalescence-driven droplet size distribution

Received: 30 July 2011 / Accepted: 17 June 2012 / Published online: 7 July 2012

Abstract The 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 coalescence integrals. It will be shown that noticeable changes in the output of interest might occur. In addition, the computational efficiency of the studied methods is discussed.

1 Introduction

Population balance systems were derived to model the behavior of particle populations using distributions instead of considering individual particles. 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 droplets in flows [2–4], which is of importance, for example, for the simulation of processes in clouds.

The work of R. Bordás was supported by grant Th881/13-2 within the DFG priority programme 1276 MetStröm: Multiple Scales in Fluid Mechanics and Meteorology.
The work of E. Schmeyer was supported by grant Jo329/8-2 within the DFG priority programme 1276 MetStröm: Multiple Scales in Fluid Mechanics and Meteorology.

Communicated by R. Klein.

R. Bordás · D. Thévenin
University of Magdeburg “Otto von Guericke”, LSS/ISUT, Universitätsplatz 2, 39106 Magdeburg, Germany
E-mail: bordas@ovgu.de

D. Thévenin
E-mail: thevenin@ovgu.de

V. John (✉) · E. Schmeyer
Weierstrass Institute for Applied Analysis and Stochastics Leibniz Institute in Forschungsverbund Berlin e. V. (WIAS),
Mohrenstr. 39, 10117 Berlin, Germany
E-mail: volker.john@wias-berlin.de

E. Schmeyer
E-mail: ellen.schmeyer@wias-berlin.de

V. John
Department of Mathematics and Computer Science, Free University of Berlin, Arnimallee 6, 14195 Berlin, Germany

The dominating physical mechanism for the evolution of the droplet size distribution (DSD) in this application is the coalescence 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, for example, energy and mass balances. The equation for the DSD possesses some properties that make its numerical simulation challenging. First, the DSD depends not only on time and space, as for example 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 coalescence 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 specific discretization techniques is necessary.

There are several ways to treat these difficulties. In order to avoid the increase in dimension, a technique often employed consists in using methods of moments (MOM), see [5] for the first approach in this direction. In this technique, only the first moments of the DSD are simulated. This approach usually requires the closure of the system for the first moments. The currently most popular way is the quadrature MOM (QMOM) [6], which applies a numerical quadrature whose weights and nodes depend on the moments (of a previous time). The direct QMOM (DQMOM) [7] 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 [8,9].

Another approach consists in addressing directly the equation for the DSD. For the numerical simulation of this equation, operator-splitting techniques or direct discretizations can be applied. In operator-splitting techniques, equations defined on the spatial domain and on the domain for the internal coordinates are solved sequentially. However, this approach introduces a splitting error whose magnitude is unknown. Altogether, we think that a direct discretization of the equation for the DSD in the higher-dimensional domain possesses the highest potential for accurate results, compared with moment-based methods and operator-splitting schemes. This is the main motivation for considering such direct discretizations in this paper. The use of different numerical methods is also possible in the class of direct discretizations. 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 [10,11]. 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 the present paper, the output quantity of interest is a time-space-averaged DSD at the outlet of the flow domain. This global quantity is less sensitive to numerical and modeling techniques than the quantities studied in [10,11]. The droplets move in a turbulent flow field, and quantitative experimental data are available for both the flow field (air) and the DSD (water droplets), see http://www.ovgu.de/isut/lss/metstroem for the data and [12,13] for the descriptions of the experiments. These experiments were designed such that selected properties of flows in clouds are reproduced, in particular the droplet number density per unit volume and the liquid water content (2 g/kg). At the same time, other properties, like gravitational collisions of droplets or the integral scale of turbulence, are not covered by the experiments. 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 coalescence term are studied.

The paper is organized as follows. The model of the considered process is introduced in Sect. 2. Then, Sect. 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 Sect. 4. Numerical studies are presented in Sect. 5. Finally, a summary and an outlook are given. Appendix A discusses one of the numerical methods in more detail at a model problem to provide an explanation of the results obtained with this method in the simulations of the DSD.

2 The population balance model for the simulation of the DSD

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

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

$$\nabla\cdot\mathbf{u} = 0 \quad \text{in } (0,t_e)\times\Omega, \tag{2}$$

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/(ms) is the dynamic viscosity of air at the same temperature, $\mathbb{D}(\mathbf{u}) =$

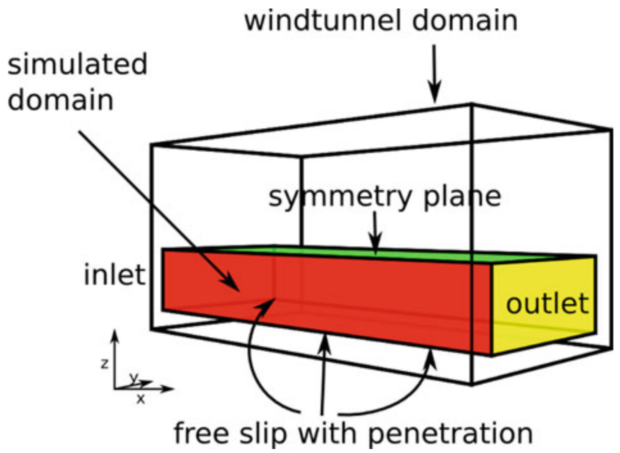


Fig. 1 Sketch of the wind tunnel, the computational domain, and the boundary conditions for the flow field

$(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)/2$ is the velocity deformation tensor, and t_e denotes the final time. Because of the low Mach number 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, see Fig. 1. The section for the measurements was of hexahedral shape and it was situated in the lower half of the wind tunnel test section. Because of some uncertainties concerning the most appropriate boundary conditions at the outlet, the computational domain $\Omega = (0, 0.5) \times (-0.225, 0.225) \times (-0.18, 0)\ \mathrm{m}^3$ is chosen to be somewhat longer than the measurement section, 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 corresponding to the outlet of the measurement section. Hence, a possible error source in the comparison of experimental and numerical data was eliminated.

A fully developed flow field, which was computed in a preprocessing step, is used as initial condition. 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 $\mathbf{u}_{\text{air,exp}}(0, y, z)$ is disturbed with white noise

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

where $\boldsymbol{\sigma}_{\text{air,exp}}(0, y, z)$ 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.

Experimental data at the outlet of the computational domain $\Gamma_{\text{out}} = \{0.5\} \times (-225, 225) \times (-180, 0)$ 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. The 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

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

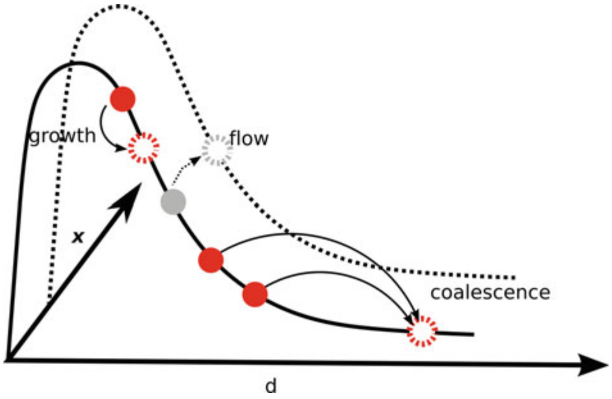


Fig. 2 Sketch of the mechanisms acting on the droplets along internal (d) and external ($\mathbf{x}$) coordinates

The Reynolds number of the flow, based on the kinematic viscosity $\nu = \mu/\rho = 1.5073 \times 10^{-5}$ m^2/s, the hydraulic diameter of the wind tunnel $L = 0.5454$ m, and the integral mean value of the velocity at the inlet of the computational domain $U = 2.4491$ m/s is Re $= 88,618$.

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

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

see Fig. 2 for a schematic sketch of the process. 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, C_+ is the source of coalescence, and C_- is the coalescence sink. The growth constant is given by 5.0613×10^{-10} m^2/s, see [13] for the derivation of this value. In all simulations, a supersaturation of 1% is assumed. Hence, the growth rate, which is the product of the growth constant and the supersaturation, is given by $a = 5.0613 \times 10^{-12}$ m^2/s.

The experimental data for the air velocity and for the droplet velocity at the plane $x = 0$ m are extrapolated constantly into Ω, for example, $\mathbf{u}_{\text{drop,exp}}(x, y, z) = \mathbf{u}_{\text{drop,exp}}(0, y, z)$ for all $x \in [0, 0.5]$ m. The same procedure is applied to the experimental data for the air flow $\mathbf{u}_{\text{air,exp}}(x, y, z) = \mathbf{u}_{\text{air,exp}}(0, y, z)$ for all $x \in [0, 0.5]$ m. Then, the first component of the droplet velocity is defined by

$$(\mathbf{u}_{\text{drop}})_1(t, x, y, z) := \left(\mathbf{u}_{\text{drop,exp}} - \mathbf{u}_{\text{air,exp}}\right)(x, y, z) + (\mathbf{u}_{\text{air,sim}})_1(t, x, y, z),$$

where $(\mathbf{u}_{\text{air,sim}})_1$ is the first component of the computed velocity field from (1), (2). The other components of the droplet velocity are set to be the velocity components coming from the solution of (1), (2).

The model for the coalescence is explained best by considering the volume of the droplets, see [5,14]. Let f_V be the DSD and $C_{+,V}$, $C_{-,V}$ 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 coalescence 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' and the second one has volume $V - V'$, and vice versa. Hence, the model has the form

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

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

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

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

It is also possible to define the coalescence integrals in terms of the diameter, see [14],

$$C_+ = \frac{d^2}{2} \int_{d_{\min}}^{d} \frac{\kappa_{\mathrm{col}}\left(\left(d^3-(d')^3\right)^{1/3}, d'\right)}{(d^3-d'^3)^{2/3}} f\left(\left(d^3-(d')^3\right)^{1/3}\right) f(d')\, dd', \tag{6}$$

$$C_- = -f(d) \int_{d_{\min}}^{d_{\max}} \kappa_{\mathrm{col}}(d, d') f(d')\, dd'. \tag{7}$$

Collisions of droplets can occur because

- very small droplets bump against each other due to Brownian motion,
- droplets are pushed to other droplets moving in a slower fluid layer (shear-induced),
- or faster moving droplets overtake slower droplets.

The last mechanism is particularly important for the simulation of real clouds [15]. It has been modeled by gravitational kernels, like the Hall kernel [16] or, in the presence of turbulence, the Ayala kernel [17,18]. A gravitational collision kernel is not considered here, since its effect compared with that of the turbulent collision kernel is negligible in case of strong turbulence with large dissipation rate [19,20], as found in the present experiment [12].

The collision kernel, which will be described in terms of volume, is the product of two factors $\kappa_{\mathrm{col}}(V, V') = p_{\mathrm{col}}(V, V')p_{\mathrm{eff}}(V, V')$. The first factor models the probability of the collision of two droplets with volume V and V'. The efficiency of the collisions, which means the amount of collisions actually leading to a coalescence of droplets, is modeled by the second factor. This factor is chosen to be constant since other models are not available. The constant can be included into scaling factors for the individual terms of the following kernel, see [21,22],

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

where $k_B = 1.38 \times 10^{-23}$ 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 collision since in this case, the last factor becomes large. The second term is shear-induced [23] 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 [13]. Using one of the methods described below for simulating the population balance system, see Sect. 5, the values $C_{\mathrm{brown}} \simeq 1.5 \times 10^6$ and $C_{\mathrm{shear}} \simeq 0.1$ were obtained.

The boundary condition for the droplets at the inlet is based on experimental data. Measurements were taken 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 $f_{\mathrm{in,exp}}(\mathbf{x}, d)$, and the corresponding standard deviation $\sigma_f(\mathbf{x}, d)$ can be found in [13]. The values in the nodes are extrapolated to the inlet boundary and the boundary condition reads as follows

$$f(t, \mathbf{x}, d) = \begin{cases} f_{\mathrm{in,exp}}(\mathbf{x}, d) + \mathrm{rand}_{\mathrm{normal}}(t, \mathbf{x})\sigma_f(\mathbf{x}, d), & \text{for } d \in [d_{\min}, d_{\max}], \\ 0, & \text{for } d \in [d_{\min,\mathrm{art}}, d_{\min}) \end{cases}$$

with $\mathbf{x} = (0, y, z) \in \Gamma_{\mathrm{in}},\ t \in [(0, t_e]$. Here, $d_{\min,\mathrm{art}} = 0$ 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 [13] for a discussion of this topic. The initial condition is given by $f(0, \mathbf{x}, d) = 0$ in $\Omega \times (d_{\min,\mathrm{art}}, d_{\max})$.

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 coalescence 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 try to control the influence of the turbulence model in a better way than other approaches for the simulation of turbulent flows. 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

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

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 [24] $\nu_T = C_S\delta^2\ \|\mathbb{D}\ (\bar{\mathbf{u}})\|_F$ 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 [25] for a discussion of this topic. Numerical studies [26–29] 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 [26]. 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 [30,31], 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.

An inf-sup stable pair of velocity–pressure finite element spaces is used, namely the pair Q_2/P_1^{disc}. This pair is a popular choice [32] as it combines high accuracy and the possibility of solving the arising saddle point problems efficiently [33,34]. 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 $\{K\}$ is the set of 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 with the arithmetic mean $\overline{\eta}$ of the local indicators in order to decide which local space $L^H(K)$ is chosen: given $0 \leq C_1 \leq C_2 \leq C_3$, define $\eta := \eta_K/\overline{\eta}$, then

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

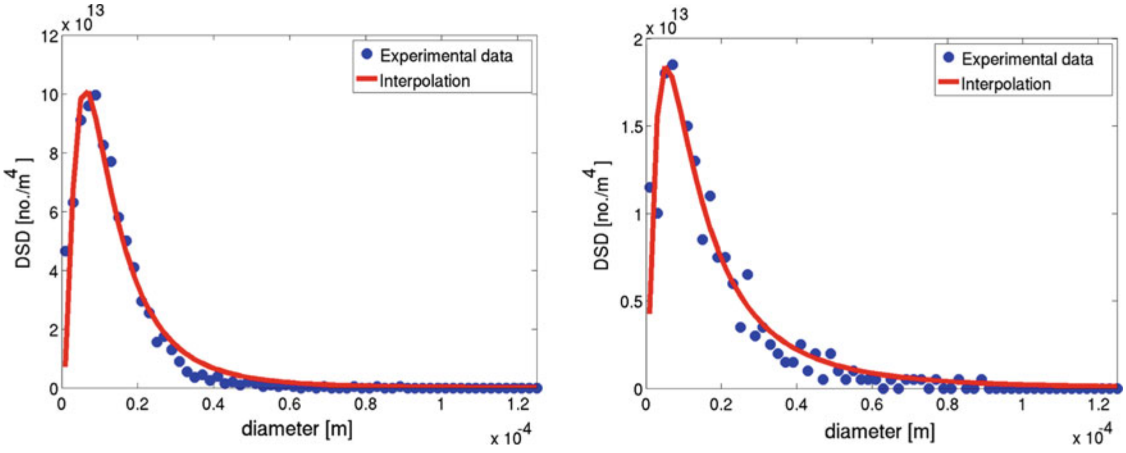


Fig. 3 Log-normal interpolation of the DSD (*left*) and of its standard deviation (*right*) at the inlet

A finite element error analysis for local projection methods of type (9) in bounded domains is available in cases where the projection spaces do not change in time, for example, in [35,36], but there is no analysis so far for the adaptive FEVMS.

4 Numerical methods for DSD equation (3)

4.1 Obtaining an inlet boundary condition

Measurements were taken in the nodes of a grid with the y-z coordinates $(-0.225, -0.215, \ldots, 0.225)$ m $\times$ $(-0.18, -0.17, \ldots, 0)$ m at the plane $x = 0$ m for 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 $1, 3, \ldots, 171$ µm. In [13], 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 interpolations 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 that obeys a certain function. The experimental data suggest a log-normal interpolation of the form

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

with the parameters σ, μ, and c_f. Exemplarily, Fig. 3 shows a representative result for the approximation in some 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 preprocessing 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 coalescence terms. The standard deviation at the inlet is also approximated with the ansatz (10).

4.2 Discretization of the differential operator

Several types of discretization techniques for the differential operator are studied. In the considered example, the tensor product form of the 4D domain allows an easy application of finite difference methods. One method of this type is studied, in combination with an explicit time-stepping scheme. However, 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 nonstandard 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.

A crucial requirement for all methods is that the computed solutions are (nearly) free of spurious oscillations, see [37] for a competitive study of such methods.

4.2.1 A total variation diminishing essentially non-oscillatory (TVD-ENO) finite difference method

In the numerical studies presented below, the combination of a third-order TVD Runge–Kutta scheme and a third-order ENO scheme is included. The appearance of spurious oscillations cannot be excluded with this method. But if such oscillations appear, they are generally small.

Write Eq. (3) in the generic form $\partial f/\partial t = F(t, f)$. Then, an optimal third-order TVD Runge–Kutta method has the form [38]

$$\begin{aligned} k_1 &= F(t_n, f_n), \\ k_2 &= F(t_n + \Delta t, f_n + \Delta t k_1), \\ k_3 &= F\left(t_n + \frac{\Delta t}{2}, f_n + \frac{\Delta t}{4} k_1 + \frac{\Delta t}{4} k_2\right), \\ f_{n+1} &= f_n + \Delta t \left(\frac{k_1}{6} + \frac{k_2}{6} + \frac{4k_3}{6}\right). \end{aligned} \tag{11}$$

The first-order derivatives on the right-hand side of (11) are approximated by finite differences. The basic idea of ENO schemes consists in constructing several interpolation polynomials of a certain order and using as approximation the smoothest polynomial.

Consider for simplicity of notation a one-dimensional situation. For a third-order ENO scheme, the seven nodes $x_{i-3} < x_{i-2} < \cdots < x_{i+3}$ 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 $(P_j)_x(x_i)$, $j = 1, \ldots, 4$. 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 nonnegative. 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 [37] for a detailed algorithm.

4.2.2 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 to deliver the best ratio of accuracy and efficiency in numerical studies of stabilized finite element methods for transport-dominated problems in [39,40]. 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

$$\left(M_C + \frac{\Delta t}{2} A\right) \mathbf{f}_{n+1} = \left(M_C - \frac{\Delta t}{2} A\right) \mathbf{f}_n + \frac{\Delta t}{2} (\mathbf{c}_n + \mathbf{c}_{n+1}). \tag{12}$$

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, that is, $f(t_n, \mathbf{x}, d) = \sum_{j=1}^N f_{j,n}\varphi_j(\mathbf{x}, d)$. The values for the coalescence integrals are computed in the vertices of the mesh cells. For this reason, the term of the coalescence integrals can be interpreted as a Q_1 finite element function. That means, there is a representation of the form $C_{+,n} + C_{-,n} = \sum_{j=1}^N \tilde{c}_{j,n}\varphi_j(\mathbf{x}, d)$. With this representation, the finite element right-hand side becomes

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

Discretization (12) is instable for transport-dominated problems. The FCT methodology changes in the first step the matrix to a stable matrix. This step introduces a large amount of diffusion. In the second step, this diffusion is removed where it is not needed by modifying the right-hand side of (12). For the definition of the stable matrix, let

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

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

$$\left(M_L + \frac{\Delta t}{2} L\right)\mathbf{f}_{n+1} = \left(M_L - \frac{\Delta t}{2} L\right)\mathbf{f}_n + \frac{\Delta t}{2}(\mathbf{c}_n + \mathbf{c}_{n+1}). \tag{13}$$

The system matrix possesses properties of an M-matrix.

The linear FCT scheme adds to the right-hand side of (13) an anti-diffusive contribution $\mathbf{c}^*(\mathbf{f}_n)$. This term is defined with the help of the residual vector, which is the difference of (13) and (12). A straightforward calculation 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 (13) is given by

$$\mathbf{c}^*(\mathbf{f}_n) = \left(\sum_{j=1}^N \alpha_{ij} r_{ij}\right)_{i=1}^N, \tag{14}$$

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

$$\begin{aligned} r_{ij} &= \Big[m_{ij}\big[(f_{n+1,i} - f_{n+1,j}) - (f_{n,i} - f_{n,j})\big] \\ &\quad - \frac{\Delta t}{2} d_{ij}\big[(f_{n+1,i} - f_{n+1,j}) + (f_{n,i} - f_{n,j})\big]\Big] \\ &= 2m_{ij}\big[(f_{n+1/2,i} - f_{n,i}) - (f_{n+1/2,j} - f_{n,j})\big] - \Delta t d_{ij}\big[(f_{n+1/2,i} - f_{n+1/2,j})\big], \end{aligned} \tag{15}$$

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 [41]. 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 (13). The result obtained with this scheme is inserted into (15). Note that this predictor step is performed with an explicit method such that a CFL condition applies, see [41].

The computation of the weights uses Zalesak's algorithm [42]. This algorithm is described and discussed also in [43]. In the simulations presented below, the version of Zalesak's algorithm is used, which is given in [37].

The linear system (13) with the modification (14) of the right-hand side is solved with a BiCGSTAB method with an SSOR preconditioner [44]. 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

$$\sum_{k=1}^{4}\sum_{j=1}^{N}\left(\mathbf{v}_k\partial_k\varphi_j, \varphi_i\right) \tag{16}$$

for the matrix entry (i, j). Here, $\mathbf{v}_k$ stands for the convection, that is, for $\mathbf{u}_{\text{drop}}$ or for the growth. For the form (16), the divergence of $\mathbf{u}_{\text{drop}}$ is assumed to be small. In fact, 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. Because the interpolations of the experimental data are constant in Ω, they are divergence-free. The computed flow field is discretely divergence-free, but not pointwise. Hence, the divergence-free constraint will be violated somewhat but it can be assumed that this violation is small.

In our implementation, a two-point Gaussian quadrature rule in each dimension is used to keep the quadrature errors for (16) 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 that were performed in the simulation of the DSD.

4.2.3 A linear Crank–Nicolson group finite element FCT (CN-GFCT) method

The most serious drawback of CN-FCT is the computational cost that 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, for example, in [45]. The group finite element method was applied in combination with FCT schemes already, for example, in [41,43].

Let

$$\nabla\cdot(\mathbf{u}_{\text{drop}} f) + \frac{\partial}{\partial d}\left(\frac{a}{d} f\right) = \nabla\cdot(\mathbf{v} f), \tag{17}$$

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, see above for a discussion of this point.

The basic idea of the group finite element method consists in not only using f as variable in (3) but to use for (17) 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 (17) gives

$$(\mathbf{v} f) \approx (\mathbf{v} f)^h = \sum_{k=1}^{4}\left(\sum_{j=1}^{N}(\partial_k\varphi_j, \varphi_i)(\mathbf{v}_j)_k f_j\right). \tag{18}$$

The matrices $B_k = (\partial_k\varphi_j, \varphi_i)_{i,j=1}^{N}$, $k = 1, \ldots, 4$ have to be assembled only once. To obtain the approximation of the transport matrices from (18), B_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 (16) with (18), 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 coalescence term

Let $0 = d_0 < d_1 = d_{\min} < \cdots < d_M = d_{\max}$ 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 to the internal coordinate and $f(d)$ vanishes in (d_0, d_1). The DSD is assumed to be continuous within each interval.

4.3.1 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.

4.3.2 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 preprocessing step.

This approach will be illustrated exemplarily 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

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

For the integral of the kernel holds

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

$i = 1, \ldots, M - 1$. Since the grid with respect to the internal coordinate is given, the integrals on the right-hand side can be computed in a preprocessing 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.

4.3.3 A mass-conserving method for computing convolution integrals

The method, which was developed in [46–48], will be applied to the volume formulations of the coalescence 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 [49].

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, which is refined toward 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: $|V_{i+1} - V_i|/|V_i - V_{i-1}| = l$, $l \in \{1, 2\}$, $i = 1, \ldots, M-1$, 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, that is, the collision kernel can be written in the form $\kappa_{\text{col}}(V, V') = \sum_{i=1}^{k} a_i(V)b_i(V')$. Then, the coalescence term becomes

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

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 [46–48].

The exact convolution $\omega_{\text{exact}} = \sum_{i=1}^{M} \varphi_i * \psi_i$ 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 $\omega_{\text{exact}} - \omega_{\text{comp}}$ is L^2 orthogonal to all piecewise linear functions, one gets for all intervals $[V_i, V_{i+1}]$

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

that is, 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 coalescence, only nonnegative contributions will be neglected by not considering the complete support of the convolution such that the mass will always decrease. To avoid the decrease in mass, the following correction to the coalescence term is applied

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

Then, although local mass conservation is violated, the total mass of the computed coalescence 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 [49].

5 Numerical studies

The numerical studies presented below were performed with the code MooNMD [50]. They are based on non-dimensional equations using the following reference values

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

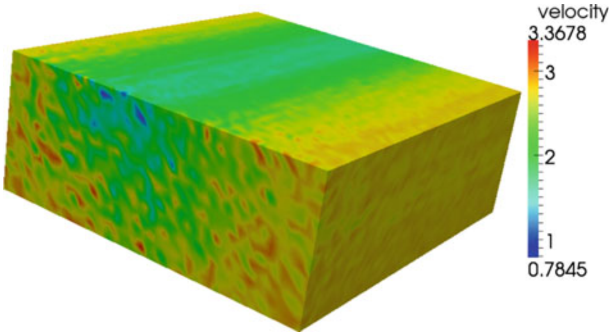


Fig. 4 Snapshot of the stream-wise component of the velocity field. The mean profile of the velocity field is due to the experimental setup, see [13]

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, is used for the triangulation of Ω. With this grid, the positions of the measurement points are located at nodes. Since the computational domain is away from the walls of the wind tunnel, the boundary layers at these walls are outside the computational domain. For this reason, an equi-distant grid can be used. 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 $d_\infty = d_{\max} = 175 \times 10^{-6}$ m is used. The second grid is refined toward 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 $d_\infty = d_{\max} = 249 \times 10^{-6}$ m. With the used grid, $[124.5 \times 10^{-6}, 249 \times 10^{-6}]$ 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 snapshot of the flow field is presented in Fig. 4. 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 stochastic perturbation at the inlet Γ_{in}, the temporal variation of the minimal and maximal value of the stream-wise velocity is quite large. From the mean value of the stream-wise velocity, it can be deduced that the average residence time of a droplet in the measurement section is less than 0.25 s. Hence, a fully developed process is obtained in the time interval for computing the averaged data. It can be seen that the flow is basically unidirectional in x-direction. It follows that the droplets are transported essentially from the inlet to the outlet. In addition, it was already observed in [13] that different turbulence models possess only a small impact on the computed quantities of interest. We think that the unidirectionality of the flow is an essential reason for this behavior.

Results for different discretizations for the temporal derivative and the spatial derivative of the transport terms in Eq. (3) for the DSD are presented in Fig. 5 and Table 1. The curve for the finite difference scheme TVD-ENO fits 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 TVD-ENO in [13]. The results of CN-FCT and CN-GFCT are similar, but too few small droplets are predicted. This effect can be explained by the smearing, which is introduced by CN-FCT and CN-GFCT if the convection is aligned with the grid, see Appendix A for more details. If the fitting of the model parameters is done with CN-FCT or CN-GFCT, we could observe that then $C_{\mathrm{brown}} = 1 \times 10^6$, instead of $C_{\mathrm{brown}} = 1.5 \times 10^6$, would lead to a better agreement of experimental data and numerical results. This observation demonstrates the influence of numerical methods on the calibration of model parameters. The parameters were fitted with a trial-and-error strategy. An automatic parameter design requires the embedding of the population balance system in an optimization problem. To the best of our knowledge, this approach is not yet studied in the literature so far.

Figure 6 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 toward

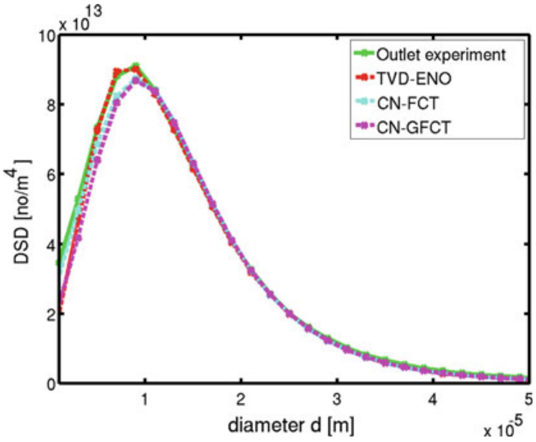


Fig. 5 Comparison of discretizations for temporal and transport-dominated spatial derivatives, uniform grid for DSD, pre-computed integrals for coalescence

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 coalescence

Method	Mode (m)	Maximum (no./m^4)	Comp. time (s)
Experimental data outlet	9×10^{-6}	9.0987×10^{13}	
TVD-ENO	9×10^{-6}	9.0006×10^{13}	140
CN-FCT	9×10^{-6}	8.7379×10^{13}	995
CN-GFCT	9×10^{-6}	8.6794×10^{13}	322

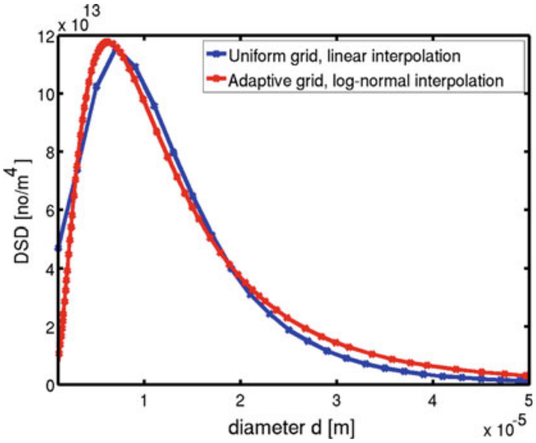


Fig. 6 Time-space-averaged inlet condition for the DSD

smaller droplets can be observed. This shift is still present in the computed results at the outlet, see Fig. 7. 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}.

The effect of applying different methods for computing the coalescence integrals is studied in Figs. 8, 9, 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.

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 method with explicit time-stepping scheme is faster than the finite element methods with implicit temporal discretization. CN-GFCT takes more than twice

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	7×10^{-6}	1.1560×10^{14}
Log-normal interpolation, adaptive grid	6.0688×10^{-6}	1.1767×10^{14}

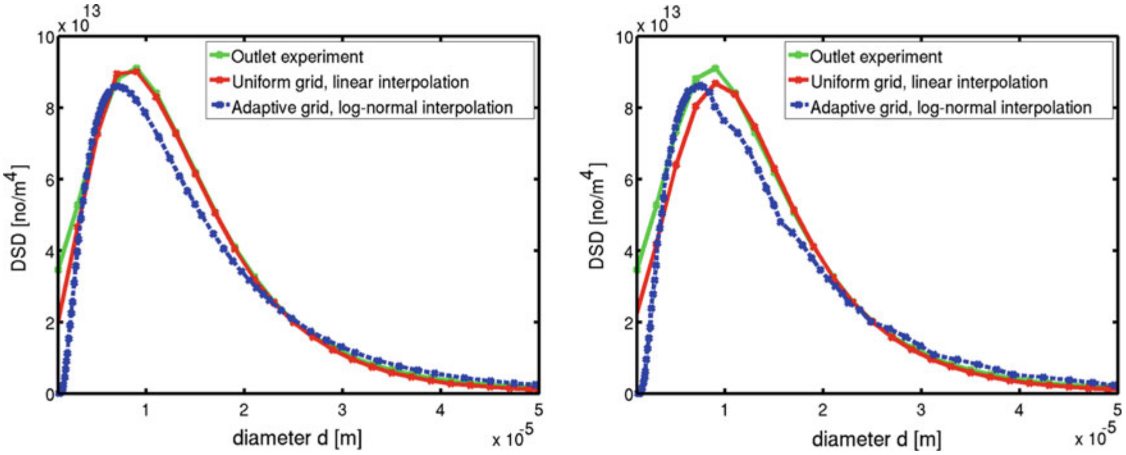


Fig. 7 Comparison of different approximations of the inlet boundary condition for the DSD with TVD-ENO (*left*) and CN-GFCT (*right*), pre-computed integrals

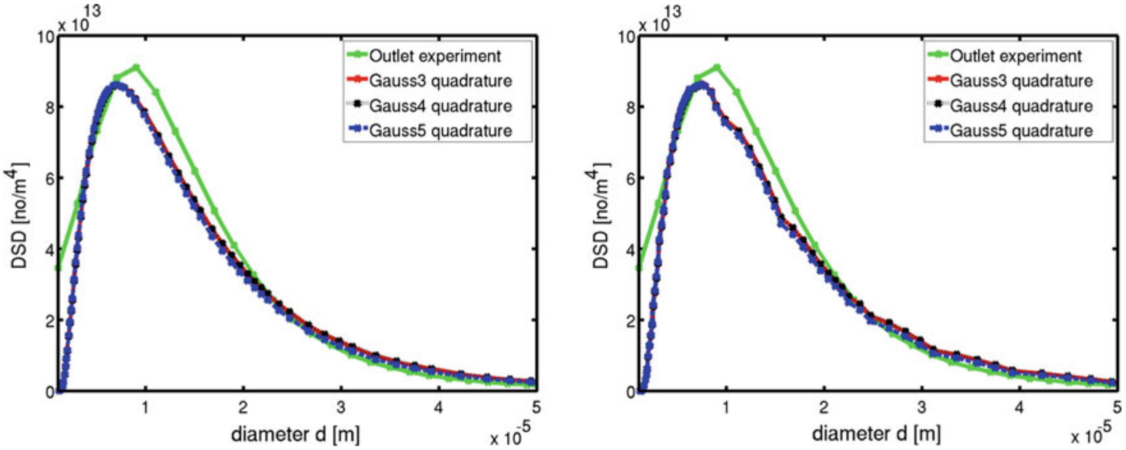


Fig. 8 Comparison of Gaussian quadrature methods for computing the coalescence of droplets, with TVD-ENO (*left*) and CN-GFCT (*right*), adaptive grid with log-normal interpolation for DSD

as long as TVD-ENO. The by far slowest scheme is CN-FCT, whose simulation time is three times longer than CN-GFCT. With respect to the method for computing the coalescence, 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 Sect. 5 reveal that the use of different numerical methods for solving Eq. (3) for the DSD might possess a non-negligible impact on the computed time-space-averaged DSD at the outlet. Unlike [10,11], 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 coalescence integrals lead to very similar

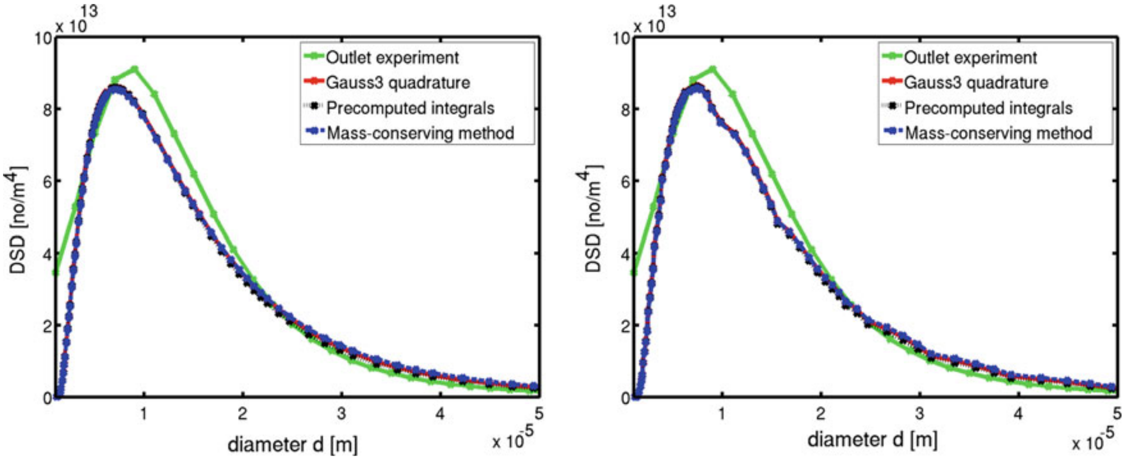


Fig. 9 Comparison of different methods for computing the coalescence of droplets, with TVD-ENO (*left*) and CN-GFCT (*right*), adaptive grid with log-normal interpolation for DSD

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

Method	Mode (m)	Maximum (no./m^4)	Comp. time (s)
Gauss3	7.0697×10^{-6}	8.6010×10^{13}	271
Gauss4	7.0697×10^{-6}	8.5887×10^{13}	309
Gauss5	7.0697×10^{-6}	8.6096×10^{13}	338
Pre-computed integrals	7.0697×10^{-6}	8.5981×10^{13}	174
Mass-conserving method	7.0697×10^{-6}	8.5434×10^{13}	215

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

Method	Mode (m)	Maximum (no./m^4)	Comp. time (s)
Gauss3	7.4425×10^{-6}	8.6318×10^{13}	376
Gauss4	7.4425×10^{-6}	8.6165×10^{13}	414
Gauss5	7.4425×10^{-6}	8.6285×10^{13}	442
Pre-computed integrals	7.4425×10^{-6}	8.6196×10^{13}	276
Mass-conserving method	7.4425×10^{-6}	8.5768×10^{13}	303

results. However, this situation might change if long-term simulations are performed where, for example, the loss of mass in some methods might become crucial.

This paper studied a situation where the DSD is defined in a 4D tensor product domain and where the grid could be easily aligned with the main flow direction. For this situation, the use of TVD-ENO together with the mass-conserving quadrature method for the coalescence integrals can be recommended as numerical method for solving the equation for the DSD.

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 [10,11], since it will be hard to obtain experimental data for such outputs. Another important topic is the inclusion of moment-based methods and operator-splitting schemes into the studies to quantify the differences in accuracy. Finally, the consideration of more complicated flow domains is of interest. Such domains lead to non-tensor product domains for the DSD such that the application of finite difference schemes becomes harder. These topics will be pursued next in our future work.

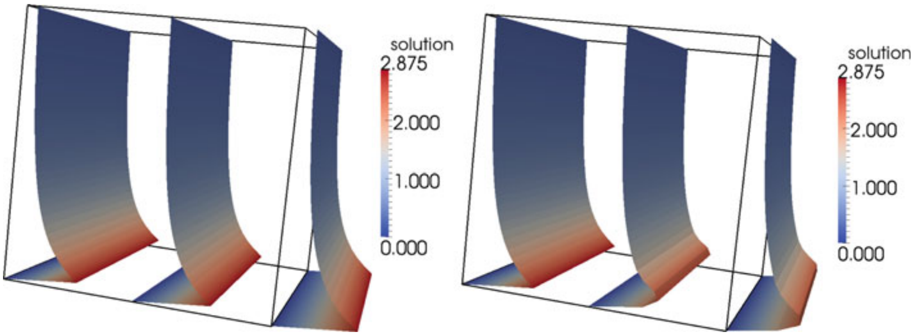


Fig. 10 Three-dimensional convection–diffusion problem with log-normal profile at the inlet; *left* picture: TVD-ENO, *right* picture: CN-GFCT; solution after a complete transport of the inlet condition to the outlet; *left* plane: inlet ($x = 0$), center plane: $x = 0.5$, *right* plane: outlet $x = 1$

Appendix A: On the smoothing introduced by the FCT schemes on aligned grids

In the comparative study of discretizations for evolutionary convection–diffusion equations [37], a rather pathological two-dimensional example was presented were the FCT schemes computed extremely smeared solutions. In this example, an impulse was transported along a grid line. The explanation for this behavior is that the example is essentially one-dimensional but the FCT approach is multi-dimensional. A dimensional splitting of the FCT approach would lead to better solutions.

This appendix shows that a strong smearing of solutions computed with the FCT schemes occurs also in situations that are similar to the setup of the DSD Eq. (3). To this end, consider an evolutionary convection–diffusion equation with the diffusion 10^{-6}, the convection $(1, 0, 0)^T$, and the right-hand side equal to zero in $(0, 1)^3$. On the inflow boundary $x = 0$, a Dirichlet boundary condition with a log-normal profile

$$f(z) = \begin{cases} \dfrac{2}{\sqrt{2\pi}\sigma z} \exp\left(-\dfrac{\ln^2(z)}{2\sigma^2}\right) & \text{with } \sigma = 2,\ z \in (10^{-6}, 1] \\ 0 & \text{else} \end{cases}$$

is prescribed for all times. For $z = 0$, homogeneous Dirichlet boundary conditions and on all other boundaries, homogeneous Neumann conditions are used. At the initial time, the solution is prescribed to fulfill the Dirichlet boundary condition and it is set to zero elsewhere in the domain.

With this setup, the inlet boundary condition should be transported from the inlet to the outlet within one time unit and with very little smearing. The grid is chosen to consist of 32 × 32 × 32 cubes. Thus, it is aligned to the convection. This situation is very desirable in flow simulations.

The inlet condition and the results at the outlet for the different methods are presented in Fig. 10. It can be clearly seen that the TVD-ENO scheme works well but the CN-FCT scheme gives a considerably smeared solution. The construction of this example is similar to the setup of the DSD equation (without coalescence) and one can see a similar smearing of the solution as in Fig. 5.

Altogether, the FEM-FCT schemes do not work well if convection and grid are aligned. Probably, one can improve the situation in the way described above. However, this raises the question how to detect the subregions where the dimensional splitting should be used and where not.

References

1. Ramkrishna, D.: Population Balances: Theory and Applications to Particulate Systems in Engineering. Academic Press, San Diego (2000)
2. Rogers, R.R., Yau, M.K.: Cloud Physics. 3rd edn. Butterworth–Heinemann, Oxford (1989)
3. Shaw, R.A.: Particle-turbulence interactions in atmospheric clouds. Annu. Rev. Mech. **35**, 183–227 (2003)
4. Mukhopadhyay, A., Jasor, G., Polifke, W.: Simulation of pure sedimentation of raindrops using quadrature method of moments. Atmos. Res. **106**, 61–70 (2012)
5. Hulburt, H.M., Katz, S.: Some problems in particle technology—a statistical mechanical formulation. Chem. Eng. Sci. **19**, 555–574 (1964)
6. McGraw, R.: Description of aerosol dynamics by the quadrature method of moments. Aerosol Sci. Technol. **27**, 255–265 (1997)

7. Marchisio, D.L., Fox, R.O.: Solution of population balance equations using the direct quadrature method of moments. J. Aerosol Sci. **36**, 43–73 (2005)
8. John, V., Angelov, I., Öncül, A.A., Thévenin, D.: Techniques for the reconstruction of a distribution from a finite number of its moments. Chem. Eng. Sci. **62**, 2890–2904 (2007)
9. de Souza, L.G.M., Janiga, G., John, V., Thévenin, D.: Reconstruction of a distribution from a finite number of moments with an adaptive spline-based algorithm. Chem. Eng. Sci. **65**, 2741–2750 (2010)
10. John, V., Roland, M., Mitkova, T., Sundmacher, K., Tobiska, L., Voigt, A.: Simulations of population balance systems with one internal coordinate using finite element methods. Chem. Eng. Sci. **64**, 733–741 (2009)
11. John, V., Roland, M.: On the impact of the scheme for solving the higher dimensional equation in coupled population balance systems. Int. J. Numer. Meth. Eng. **82**, 1450–1474 (2010)
12. Bordás, R., Hagemeier, T., Wunderlich, B., Thévenin, D.: Droplet collisions and interaction with the turbulent flow within a two-phase wind tunnel. Phys. Fluids **23**, 085105 (2011)
13. Bordás, R., John, V., Schmeyer, E., Thévenin, D.: Measurement and simulation of a droplet population in a turbulent flow field. Comput. Fluids **66**, 52–62 (2012)
14. Mersmann, A.: Crystallization Technology Handbook. 2nd edn. Marcel Dekker Inc., New York (2001)
15. Pruppacher, H.R., Klett, J.D.: Microphysics of Clouds and Precipitation. 2nd edn. Kluwer Academic Publishers, Dordrech (1997)
16. Hall, W.D.: A detailed microphysical model within a two-dimensional dynamic frameword: model description and preliminary results. J. Atmos. Sci. **37**, 2486–2507 (1980)
17. Ayala, O., Wang, L.-P., Grabowski, W.W.: A hybrid approach for simulating turbulent collisions of hydrodynamically-interacting particles. J. Comput. Phys. **225**, 51–73 (2007)
18. Xue, Y., Wang, L.-P., Grabowski, W.W.: Growth of cloud droplets by turbulent collision-coalescence. J. Atmos. Sci. **65**, 331–356 (2008)
19. Pinsky, M., Khain, A., Krugliak, H.: Collisions of cloud droplets in a turbulent flow. part v: Application of detailed tables of turbulent collision rate enhancement to simulation of droplet spectra evolution. J. Atmos. Sci. **65**(2), 357–374 (2008)
20. Wang, L.P., Grabowski, W.W.: The role of air turbulence in warm rain initiation. Atmos. Sci. Lett. **10**(1), 1–8 (2009)
21. Berglund, K.A.: Analysis and measurement of crystallization utilizing the population balance. In: Myerson, A. (ed.) Handbook of Industrial Crystallization, Butterworth-Heinemann, Oxford (2002)
22. Lindenberg, C., Schöll, J., Vicum, M., Brozio, J.: L-glutamic acid precipitation: agglomeration effects. Crystal Growth Des. **8**, 224–234 (2008)
23. Smoluchowski, M.V.: Versuch einer mathematischen Theorie der Koagulationskinetik kolloider Lösungen. Z. Phys. Chem. **92**, 129–168 (1917)
24. Smagorinsky, J.S.: General circulation experiments with the primitive equations. Mon. Weather Rev. **91**, 99–164 (1963)
25. John, V., Kaya, S., Layton, W.: A two-level variational multiscale method for convection-dominated convection-diffusion equations. Comput. Meth. Appl. Math. Eng. **195**, 4594–4603 (2006)
26. John, V., Kaya, S.: A finite element variational multiscale method for the Navier-Stokes equations. SIAM J. Sci. Comput. **26**, 1485–1503 (2005)
27. John, V., Kindl, A.: Variants of projection-based finite element variational multiscale methods for the simulation of turbulent flows. Int. J. Numer. Meth. Fluids **56**, 1321–1328 (2008)
28. John, V., Kindl, A.: Numerical studies of finite element variational multiscale methods for turbulent flow simulations. Comput. Methods Appl. Mech. Eng. **199**, 841–852 (2010)
29. John, V., Roland, M.: 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)
30. John, V., Kindl, A.: A variational multiscale method for turbulent flow simulation with adaptive large scale space. J. Comput. Phys. **229**, 301–312 (2010)
31. John, V., Kindl, A., Suciu, C.: Finite element LES and VMS methods on tetrahedral meshes. J. Comput. Appl. Math. **233**, 3095–3102 (2010)
32. Gresho, P.M., Sani, R.L.: Incompressible Flow and the Finite Element Method. Wiley, Chichester (2000)
33. John, V., Matthies, G.: Higher order finite element discretizations in a benchmark problem for incompressible flows. Int. J. Num. Meth. Fluids **37**, 885–903 (2001)
34. John, V.: 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)
35. John, V., Kaya, S.: Finite element error analysis of a variational multiscale method for the Navier–Stokes equations. Adv. Comput. Math. **28**, 43–61 (2008)
36. Röhe, L., Lube, G.: Analysis of a variational multiscale method for large-eddy simulation and its application to homogeneous isotropic turbulence. Comput. Methods Appl. Mech. Eng. **199**, 2331–2342 (2010)
37. John, V., Novo, J.: On (essentially) non-oscillatory discretizations of evolutionary convection-diffusion equations. J. Comput. Phys. **231**, 1570–1586 (2012)
38. Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock capturing schemes. J. Comput. Phys. **77**, 439–471 (1988)
39. John, V., Schmeyer, E.: Finite element methods for time-dependent convection-diffusion-reaction equations with small diffusion. Comput. Methods Appl. Mech. Eng. **198**, 475–494 (2008)
40. John, V., Schmeyer, E.: On finite element methods for 3d time-dependent convection-diffusion-reaction equations with small diffusion. In: Hegarty, A. (ed.), BAIL 2008—Boundary and Interior Layers, Number 69 in Lecture Notes in Computational Science and Engineering, pp. 173–181. Springer, Berlin (2009)
41. Kuzmin, D.: Explicit and implicit FEM-FCT algorithms with flux linearization. J. Comput. Phys. **228**, 2517–2534 (2009)
42. Zalesak, S.T.: Fully multi-dimensional flux corrected transport algorithms for fluid flow. J. Comput. Phys. **31**, 335–362 (1979)
43. Kuzmin, D., Möller, M. : Algebraic flux correction I. Scalar conservation laws. In: Kuzmin, D., Löhner, R., Turek, S. (eds.) Flux-Corrected Transport: Principles, Algorithms and Applications, pp. 155–206. Springer, Berlin (2005)

44. Saad, Y.: Iterative Methods for Sparse Linear Systems. 2nd edn. SIAM Philadelphia, Philadelphia (2003)
45. Fletcher, C.A.J.: The group finite element formulation. Int. J. Numer. Methods Fluids **37**, 225–243 (1983)
46. Hackbusch, W.: On the efficient evaluation of coalescence integrals in population balance models. Computing **78**, 145–159 (2006)
47. Hackbusch, W.: Approximation of coalescence integrals in population balance models with local mass conservation. Numer. Math. **106**, 627–657 (2007)
48. Hackbusch, W.: Convolution of hp-functions on locally refined grids. IMA J. Numer. Anal. **29**, 960–985 (2009)
49. Hackbusch, W., John, V., Khachatryan, A., Suciu, C.: A numerical method for the simulation of an aggregation-driven population balance system. Int. J. Numer. Methods Fluids **69**, 1646–1660 (2012)
50. John, V., Matthies, G.: MooNMD—a program package based on mapped finite element methods. Comput. Visual. Sci. **6**, 163–170 (2004)