Stefan Darmann, BSc

# Optimization of an Air Classifier

**MASTERARBEIT**

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Masterstudium Verfahrenstechnik

eingereicht an der

**Technischen Universität Graz**

Betreuer

Ass.Prof. Dipl.-Ing. Dr.techn. Stefan Radl

Institut für Prozess- und Partikeltechnik

Graz, Juni 2014

## EIDESSTATTLICHE ERKLÄRUNG

## *AFFIDAVIT*

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

*I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.*

| | |
|---|---|
| Datum / Date | Unterschrift / Signature |

OpenFOAM® and OpenCFD are registered trademarks of OpenCFD Ltd. This offering is not affiliated, approved or endorsed by ESI Group, the producer of the OpenFOAM® software and owner of the OpenFOAM® trade mark.

CFDEMcoupling stands for Computational Fluid Dynamics (CFD)-Discrete Element Method (DEM) coupling. CFDEMcoupling is an open-source code, distributed freely under the terms of the GNU Public License (GPL).

# Kurzfassung

Im letzten Produktionsschritt der industriellen Herstellung von feinkörnigen Pulvern ist das Abtrennen grober Partikel mittels Stabkorbsichtern ein wichtiger Vorgang. Die Produktqualität und die Trennleistung werden dabei wesentlich vom Strömungsfeld im Sichter beeinflusst. Aufgrund der komplexen Geometrie sind analytische Betrachtungen, oder experimentelle Untersuchungen des Trennprozesses beschränkt. Numerische Strömungs- und Partikelsimulationen können einen besseren Einblick in den Trennprozess verschaffen, um eine Optimierung der Sichtergeometrie zu ermöglichen. Der Schwerpunkt dieser Arbeit liegt in derartigen Simulationen, um Verbesserungsmöglichkeiten bezüglich der erzielbaren Produktqualität und einer Reduktion des Energiebedarfes zu identifizieren.

Simulationen der Partikeleindüsung in eine turbulente Querströmung wurden durchgeführt, und der Einfluss verschiedener Betriebsparameter auf die Trennung untersucht. Weiters wurde der Einfluss von Änderungen der Sichtergeometrie, sowie unterschiedlicher Arten der Partikeleinbringung in den Sichter untersucht. Simulationen der Partikelbewegung im Sichterkorb wurden ebenfalls durchgeführt.

Die Ergebnisse der Partikeleindüsung in eine Querströmung zeigen, dass die Injektions-geschwindigkeit wesentlich die Partikeldispergierung im Kanal, sowie die Auftrennung einzelner Partikelfraktionen beeinflusst. Die Injektionsgeschwindigkeit sollte deshalb kontrolliert werden, um eine möglichst gute Trennung der Partikel in einem Sichter zu erreichen. Darüber hinaus kann die Partikeldispergierung durch den Injektionswinkel beeinflusst werden. Die Simulationen des Stabkorbsichters machen deutlich, dass die Partikel nur geringfügig radial nach außen transportiert werden, und auch die axiale Dispergierung suboptimal ist. Dies wird durch Beobachtungen im Betrieb bestätigt. Die Simulationen zeigen, dass eine Verbesserung durch den Einsatz eines Partikelleitblechs und Luftdüsen erreicht werden könnte. Die Luftströmung durch den Stabkorb, und somit die Abscheidung am Rotor, wird maßgeblich von einer Wirbelstruktur zwischen benachbarten Rotorblättern beeinflusst. Diese Struktur ist wiederum von der Geometrie der Rotorblätter abhängig. Die Wirbelstruktur verursacht eine Verblockung des Rotors, und in weiterer Folge eine erhöhte (mittlere) radiale Einströmgeschwindigkeit. Der Wirbel führt jedoch auch zu einer lokal erhöhten Tangentialgeschwindigkeiten zwischen den Rotorblättern, wodurch die Abscheidung in einer komplexen Art beeinflusst wird.

# Abstract

Air classifiers are the key equipment in the final production process of fine powders. The product quality, separation efficiency, and power draw strongly depend on the flow field in the classifier. Unfortunately, it is known that analytical or experimental investigations of the flow field are rather limited. Often, numerical simulations can provide better insight and process understanding, enabling a rational optimization of classifier design. The focus of this thesis is on such a simulation, specifically to identify measures to improve product quality (i.e., reduce coarse particle slip) and minimize power draw.

Simulations of particle injection in a turbulent cross flow have been performed to understand the effect of key process parameters on the separation process. Furthermore, the influence of geometrical modifications of the classifier and particle feeding strategies was investigated. Simulations of particle motion in the rotor were performed.

The results of the cross-flow simulations reveal that the particle injection velocity strongly influences whether separation or mixing of particles occurs in the channel. Hence, the particle injection velocity should be controlled in order to exploit the separation right after particles are injected into the classifier. Furthermore, the dispersion of particles can be improved when injecting the particles against the cross-flow direction. The studies of the flow in the annular gap region indicate that the radial and axial dispersion of the particle cloud is sub-optimal. This is in agreement with experimental results provided by the industry partner. Simulations suggest that a particle deflector in combination with air nozzles can improve the situation. Investigations of the flow between the rotor blades reveal vortex formation between two neighboring rotor blades, which is influenced by blade geometry. The vortex causes a blockage of the rotor, which results in a larger (mean) radial inflow velocity. However, the vortex structure also leads to a locally larger tangential velocity, consequently influencing the separation in a complex manner.

# Acknowledgement

I owe my sincerest gratitude to all who supported me in so many ways.

First and foremost, I would like to thank my supervisor Ass.Prof. Dipl.-Ing. Dr.techn. Stefan Radl for his invaluable guidance and advice throughout this project. He introduced me to CFD-DEM and imparted comprehensive knowledge about numerous aspects of scientific work. I remain indebted to him for all his efforts.

Next, I gratefully acknowledge support from NAWI Graz by providing access to dcluster.tugraz.at.

In addition, I would like to thank all of my friends who helped me in any respects during the completion of this thesis.

Finally, I am deeply grateful to my parents, my sister and my brother, who have been supporting me for years. This thesis would not have been possible without you.

# Table of Contents

# List of Figures

# List of Tables

# 1. Introduction

Air classifiers are of major importance in the final production process of fine powders. For many subsequent powder processing industrial processes, e.g. manufacturing of polymer films, the particle size distribution (PSD) of fines is crucial. In some applications already a small number of coarse particles can lead to manufacturing problems. Hence, it is essential to understand which geometrical and process parameters influence the PSD of the product.

Numerous experimental studies on the pilot and full scale have been performed to identify key influencing parameters [1–3]. However, an experimental strategy is limited to visual observation of the particle cloud, measurement of the overall separation efficiency (e.g., from the feed and product size distribution), as well as the measurement of the overall pressure drop and power draw. Unfortunately, it is extremely difficult to measure the local particle and air velocities in these devices in order to gain a better understanding of the separation process.

Alternative to experiments, a rigorous simulation of an air classifier can be performed. This approach can provide local information (e.g., on the air velocity), and can be used to understand the effect of geometrical modifications on the classifier's performance (e.g., its separation efficiency). Unfortunately, a direct simulation of the full classifier is impossible due to (i) the turbulent flow, (ii) the presence of moving boundaries (i.e., the rotor), and (iii) the large spectrum of length and time scales. Hence, it is necessary to de-couple the flow field, and study the flow in different regions of the classifier separately. In a previous work, Radl and Mohan [4] studied the coarse particle slip by means of such a (spatially) de-coupled simulation approach. These studies have been continued in the present thesis in order to investigate the effect of new design details on the product quality and the power draw. That is, a large set of simulations concerning the gas particle-flow in a classification device have been performed.

## *1.1. Goals*

The overall goal of this work is the optimization of an air classifier by modification of some geometrical details. Specifically, this geometrical optimization refers to (i) a reduction of the coarse particle slip and (ii) the reduction of power draw at a constant product quality. Therefore, the following tasks have been defined:

- **Screening of parameters that influence particle dispersion**

  Since the dispersion of particles is supposed to influence the classifier efficiency, a screening of process parameters affecting the particle dispersion should be performed. This screening should include simulations of gas-particle flow in simplified geometries (i.e., particle injection in a turbulent cross-flow to mimic particle feeding into the primary classification air, and simulations of the classifier without detailed modelling of the rotor). These simplifications allow a time-efficient screening of the effect of simulation and process parameters (i.e., the feeding arrangement) on the dispersion of particles.

- **Validation based on visual observation of the particle cloud**

  Visual observations of the particle cloud in the air classifier should be used for a qualitative comparison with simulation data. That is, the motion of the particle cloud has to be extracted from selected simulations to enable a (qualitative) validation of the simulation results.

- **Study of gas-particle flow between rotor blades**

  The effect of different rotor configurations on the gas-particle flow between the blades should be studied. This should include simulations, in which the rotor is considered separately, i.e., only the flow in the rotor region is subject of these investigations.

## *1.2. Outline*

In the first part of the thesis, the state of the art including different approaches for modelling air classifiers is presented. In the subsequent Chapter 3, an overview of the underlying principles, key parameters and basic models used for the simulations is given. This is essential for understanding the complex features of such a classification process. The simulation setup, the effect of key numerical parameters and the results for the particle injection in a turbulent cross-flow are subject of Chapter 4. Here, a simplified 2D-setup is chosen for a fast screening of different operation conditions (i.e., particle injection angle and velocity), and the influence of numerical parameters. Simulations concerning the annular gap region without detailed modelling of the rotor region are presented in Chapter 5. Afterwards, details on air flow between the rotor blades are provided. That is, studies of the flow field and pressure drop, as well as the size distributions of particles entering the classifier's fines exit are presented in Chapter 6 and 7. Finally, conclusions are summarized in Chapter 8.

## 1.3. References

[1] K. Leschonski, K. Legenhausen, Investigation of the flow field in deflector wheel classifiers, Chem. Eng. Process. 31 (1992) 131–136.

[2] J. Galk, W. Peukert, J. Krahnen, Industrial classification in a new impeller wheel classifier, Powder Technol. 105 (1999) 186–189.

[3] A. Bauder, F. Müller, R. Polke, Investigations concerning the separation mechanism in deflector wheel classifiers, Int. J. Miner. Process. 74 (2004) 147 – 154.

[4] S. Radl, B. Mohan, Simulation of Coarse Particle Slip in an Air Classifier, Graz, Austria, 2013.

# 2. State of the Art

The separation of coarse particles from a powder mixture is a key operation in a wide range of industrial applications. The requirements on such a separation process are varying, and depend on the feed material, the operation conditions, and the product specifications for a certain application. Hence, various types and designs of air classifiers have been developed to meet these requirements. A general overview of the most commonly used types of classification devices can be found in the article of Shapiro and Galperin [1]. Additionally, the underlying principles of the classifying process are summarized by Furchner and Zampini [2].

Obviously, the fluid (i.e., air) flow field in a classification device strongly influences the particle trajectories, and hence the classification efficiency (i.e., cut size, sharpness of cut, and power draw). Different approaches are commonly used to assess these affects, with the most rudimentary one being the calculation of theoretical particle trajectories based on an undisturbed flow field and infinite particle dilution. Another, more sophisticated way is to compute the grade efficiency by employing a particle population balance equation and accounting for particle dispersion due to the background flow. For example, this enables the calculation of the theoretical cut size for different product discharge methods (see Tomas [3] for details). However, closures for the terms in the population balance equation (e.g., to quantify the dispersion effect) are required, making this models to rely on experimental data or more sophisticated simulation methods, and hence not fully predictive.

Truly predictive models can be built based on models to describe fluid flow and particle trajectories that account for phase coupling effects in classifiers. Unfortunately, the theoretical calculation (by means of analytical expressions) of the flow field in the separation device under consideration is not possible, since the classifier has a complex geometry and rotating internals [4]. Clearly, numerical methods have to be used in such a case for analyzing the separation process.

## *2.1. Deflector Wheel Classifiers*

An early study of the flow field in a deflector wheel classifier by means of experiments was performed by Leschonski and Legenhausen [5]. They reported that the vortex between two neighboring rotor blades is negatively influencing the sharpness of cut and the cut-size. They conclude that to avoid this vortex and improve the performance of the classifier, a homogenous flow pattern is required. They speculated that this be achieved when the circumferential velocity of the fluid that enters the rotor is of the same magnitude as the rotor

tip speed. Additionally, the authors propose to avoid the free vortex flow within the deflector wheel by extending the blades toward the center of the rotor, or removing the gas-particle mixture immediately after passing the blades. For their experimental studies, the authors used a single-phase model of the deflector wheel classifier, in which water is fed tangentially to the classifier through nozzles. However, the used volumetric flow rate was not specified.

For the separation of very small particles (i.e. < 5 [μm]) Galk et al. [6] proposed an improved impeller wheel classifier design. Specifically, they found that the forced vortex (used for classifying) results in a high classification efficiency. They also suggested employing a second air inlet to improve the sharpness of cut. However, the reported product size distributions were achieved for comparably small mass loadings ($\mu \leq 0.19$) and for feed rates of less than 500 [kg/h].

Bauder et al. [7] experimentally compared the performance of two classifiers with modified deflector wheel geometry. They reported that a deflector wheel with comparably large distance between the blades (an angled orientation of straight blades was used) resulted in a finer product compared to a standard wheel with straight blades. It was concluded that this was due the lower mean radial velocity resulting from the larger distance between the blades. However, the inclined blades led to a worse sharpness of cut. Additionally, the formation of a potential vortex inside the deflector wheel, which depends on the deflector wheel as well as on the vortex finder geometry, was identified to influence the classifier's performance. The experiments were performed at small mass loadings ($\mu \leq 0.04$) and the obtained cut-size varied depending on the centrifugal acceleration (rotational speed of classifier wheel) in the classifier. At lower centrifugal acceleration (i.e. 1,800 [$m/s^2$]), the difference in cut-size between the two investigated deflector wheel geometries was large (difference of ca. 60 [μm]). On the other hand, at a large centrifugal acceleration (i.e. 28,000 [$m/s^2$]) the same cut-size (ca. 10 [μm]) was obtained for both deflector wheels.

Only recently, numerical studies focusing on the prediction of deflector wheel classifier performance have been reported. Toneva et al. [8,9] investigated a combination of a hammer mill and an impeller wheel classifier (i.e., a so-called "air classifier mill") by means of single phase and coupled gas-particle flow studies. This previous work observed that the formation of a forced vortex between to neighboring rotor blades results in a drop of the separation efficiency. This is due to the increased mean radial velocity which is caused by the reduced cross section available for the gas-particle flow. The authors found that when using forward curved blades, this vortex formation is suppressed, and hence the separation efficiency is

improved. The mass loading considered by Toneva et al. was small (i.e., less than 0.084) and the obtained cut-size was not reported.

In the available literature a significantly lower throughput and mass loading as for the simulations presented in the current thesis (see Chapter 5 for details) was used. Also, the flow field in the classifier was the main focus of previous research, and particle trajectories were not predicted. Furthermore, investigations of the particle feeding strategy for high particle mass loadings, i.e., in situations where the air flow is substantially modified by the presence of particles, were not reported in literature. The lack of literature for these situations was the motivation of the studies of the particle injection into a turbulent cross-flow presented in Chapter 4. Next, a short overview of the literature concerning this cross-flow setup is presented.

## 2.2. Gas-Particle Jets in Cross Flow (JICF)

Han and Chung [10] developed a method for predicting mean trajectories of a gas-particle jet in a cross-flow. They found, that the gas-particle jet penetrated deeper into the cross flow (compared to a single-phase gas jet), and that a deeper penetration can be obtained when (i) injecting larger particles, (ii) increasing the jet velocity, and (iii) using a higher particle mass loading. While the findings (i) and (ii) can be explained by the increased Stokes number of the injected particles (i.e., a higher relative particle inertia), finding (iii) is a result of the back-coupling effect of the particle on the fluid flow field. Thus, back coupling effects need to be taken into account when aiming on a physically correct prediction of the flow in a JICF configuration.

A detailed numerical study of the particle dispersion in a cross-flow was presented by Li and Lin [11]. Here, the authors described that a counter rotating vortex pair was influencing the particle motion. This is in agreement with an earlier experimental study of Yi and Plesniak [12]. The latter authors report that smaller particles were stronger affected by the presence of the vortex pair since they were able to follow the vortices easier. Furthermore, this work includes investigations concerning the dispersion of spherical and non-spherical particles (mean diameter of 200 [μm]) in a rectangular cross-flow. The authors found that the distribution of the particles in the channel was affected by the particle-shape, i.e. the distribution of spherical particles was more uniform than that of non-spherical particles. Additionally, with increasing distance downstream of the particle injection, the distribution in the cross-flow was more uniform for both investigated types of particles.

## 2.3. Numerical Approaches to Study Moderately Dense Gas-Particle Flows

The experimental observation of the flow details in classifiers is very challenging, and typically limited to (i) qualitative observations of the particle cloud, (ii) integral flow parameters like power draw and pressure differences, as well as (iii) a global separation efficiency. A numerical simulation approach, however, can deliver local (i.e., spatially-resolved) flow information. In the next paragraph, we discuss some recent developments of such an approach, in order to highlight its advantages and drawbacks.

Commonly two different strategies are used for the simulation of gas-particle-flows. In the first approach, both the gas and the particle phase are considered as a continuum. This model is referred to as the two-fluid (TFM) or Eulerian-Eulerian (EE) model, since both phases are tracked in an Eulerian reference frame. This model is suitable for large systems with many particles but it neglects some key effects (e.g., that of particle rotation [13]). Furthermore, for polydisperse systems a large set of transport equations need to be solved since every particle class needs to be described by a separate transport equation [14]. Hence, an EE model is most efficient for monodisperse systems.

The second widely used approach is known as the Eulerian-Lagrangian (EL) model. Here, the continuous gas phase is modelled by means of a fixed Eulerian grid, whereas the solid phase is traced in a Lagrangian manner. That is, the equation of motion needs to be solved for each individual particle. Consequently, this approach is more convenient for smaller systems with comparably low particle count. However, EL simulations are preferably used for polydisperse systems since the number of transport equations is not influenced by the number of particle classes.

Several variants of the Eulerian-Lagrangian approach are available in literature. These approaches are mainly differing in the coupling algorithm which has to be used to exchange information between the dispersed and the continuous phase. While algorithms to account for the fluid flow on the particle motion (i.e., the "forward-coupling") can be implemented in a straight forward manner, models that simulate the effect of particle motion on the flow field (i.e., the "backward-coupling") are more involved. Laín and Sommerfeld [15] use a time and ensemble averaged dispersed phase source term, which is back-coupled to the momentum equation for the fluid phase using an under-relaxation procedure. A different strategy based on an EE model is used by Schellander et al. [13]. These authors use an approach referred to as EUgran+ model in which information is transferred between an Eulerian-Lagrangian and

an Eulerian-Eulerian model. Here, tracer particles are used to calculate the EL-quantities on a fixed EE flow field. Subsequently, these quantities are back-coupled to the transient EE-simulations. The two approaches discussed above provide efficient computations. However, both strategies do not yield a simultaneous solution for the continuous and disperse phase in a transient manner [16].

The approach used in this work is based on the method proposed by Zhou et al. [17]. Here, the instantaneous particle-fluid interaction force is added to the Navier-Stokes equations (see Section 3.2 for details), allowing for a fully transient back-coupled simulation. Since large systems are computational expensive, the number or particles to be tracked has to be reduced. Hence, a parcel based approach is employed. That is, computational parcels that represent a number of physical particles are used (cp. Section 3.1.7). Of course, the modelling of the particle-fluid interaction forces in this approach is based on the size of the physical particles [16].

## *2.4. References*

[1] M. Shapiro, V. Galperin, Air classification of solid particles: a review, Chem. Eng. Process. 44 (2005) 279–285.

[2] B. Furchner, S. Zampini, Air Classifying, Ullmann's Encycl. Ind. Chem. 2 (2009) 215–234.

[3] J. Tomas, Mechanische Trennprozesse, in: lecture notes, Otto von Guericke Universität Magdeburg, 2011: pp. 79–84.

[4] L. Karunakumari, C. Eswaraiah, S. Jayanti, S.S. Narayanan, Experimental and numerical study of a rotating wheel air classifier, AIChE J. 51 (2005) 776–790.

[5] K. Leschonski, K. Legenhausen, Investigation of the flow field in deflector wheel classifiers, Chem. Eng. Process. 31 (1992) 131–136.

[6] J. Galk, W. Peukert, J. Krahnen, Industrial classification in a new impeller wheel classifier, Powder Technol. 105 (1999) 186–189.

[7] A. Bauder, F. Müller, R. Polke, Investigations concerning the separation mechanism in deflector wheel classifiers, Int. J. Miner. Process. 74 (2004) 147 – 154.

[8] P. Toneva, P. Epple, M. Breuer, W. Peukert, K.-E. Wirth, Grinding in an air classifier mill — Part I: Characterisation of the one-phase flow, Powder Technol. 211 (2011) 19–27.

[9] P. Toneva, K.-E. Wirth, W. Peukert, Grinding in an air classifier mill — Part II: Characterisation of the two-phase flow, Powder Technol. 211 (2011) 28–37.

[10] K.S. Han, M.K. Chung, Numerical Simulation of Two-Phase Gas-Particle Jet in a Crossflow, Aerosol Sci. Technol. 16 (1992) 126–139.

[11] G. Li, J. Lin, Characteristics of Particle Dispersion in a Jet in Cross-Flow Based on Computational Fluid Dynamics, in: Power Energy Eng. Conf. (APPEEC), 2010 Asia-Pacific, Ieee, Chengdu, 2010: pp. 1–4.

[12] J. Yi, M.. Plesniak, Dispersion of a particle-laden air jet in a confined rectangular crossflow, Powder Technol. 125 (2002) 168–178.

[13] D. Schellander, S. Schneiderbauer, S. Pirker, Numerical study of dilute and dense poly-dispersed gas-solid two-phase flows using an Eulerian and Lagrangian hybrid model, Chem. Eng. Sci. 95 (2013) 107–118.

[14] S. Pirker, D. Kahrimanovic, A combined method for simulating gas-particle flows in highly laden cyclones, in: Fifth Int. Conf. CFD Process Ind., Melbourne, Australia, 2006: pp. 1–6.

[15] S. Laín, M. Sommerfeld, Euler/Lagrange computations of pneumatic conveying in a horizontal channel with different wall roughness, Powder Technol. 184 (2008) 76–88.

[16] S. Radl, B.C. Gonzales, C. Goniva, S. Pirker, State of the Art in Mapping Schemes for Dilute and Dense Euler-Lagrange Simulations, in: 10th Int. Conf. CFD Oil Gas, Metall. Process Ind., Trondheim, Norway, 2014: pp. 1–8.

[17] Z.Y. Zhou, S.B. Kuang, K.W. Chu, A.B. Yu, Discrete particle simulation of particle–fluid flow: model formulations and their applicability, J. Fluid Mech. 661 (2010) 482–510.

# 3. Theory and Models

This section provides an overview of the most important parameters in gas-particle flows. Also, governing equations, as well as the most important models for forces acting on individual particles are introduced. Finally, a summary of the turbulence models used in this work is provided.

## *3.1. Key Parameters Characterizing Gas-Particle Flow*

### 3.1.1. Volume Fraction and Mass Loading

The dispersed phase volume fraction for particles species *i* (e.g., particles in a certain size class) is given by:

$$\phi_{p,i} = \frac{V_{p,i}}{V_f + \sum_i V_{p,i}}, \tag{3-1}$$

where $V_{p,i}$ is the volume of species $i$ of the dispersed phase and $V_f$ is the volume occupied by the fluid. Similarly, the volume fraction for the fluid phase is defined as $\phi_f = 1 - \sum_i \phi_{p,i}$.

In order to characterize the density of the multiphase flow, an (average) inlet mass loading is introduced:

$$\mu_{inlet} = \frac{\dot{m}_{p,inject}}{Q_f \ \rho_f}. \tag{3-2}$$

Here $Q_f$, $\dot{m}_{p,inject}$ and $\rho_f$ are the volumetric air flow rate, the mass rate of injected particles and the fluid density.

These two parameters are very important since it is known that (i) the particles influence the surrounding fluid already for moderate mass loadings ($\mu$ > ca. 0.01), and (ii) that particle-particle collisions become relevant for particle volume fractions of ca. $\phi_p > 0.001$ and above [1]. Consequently, the dispersion of the injected particles gets more difficult as the mass loading increases, because the surrounding fluid is decelerated. In these moderately dense flows, particle-particle and particle-fluid interactions might dominate the dispersed phase motion. As a result, turbulent motion, for example, is of secondary importance for the mean trajectory of the particles, or the clustering phenomena observed in some devices. Hence, turbulence is not modelled in some applications.

### 3.1.2. Particle Size Distribution

The (cumulative) particle size distribution for the feed and product can be approximated by using an RRSB distribution function:

$$Q_3 = 1 - \exp\left(-\left(\frac{x}{d'}\right)^n\right). \tag{3-3}$$

In the equation above, the cumulative size distribution is related to the particle volume. The parameters $d'$ and $n$ are the characteristic particle size (i.e., the particle diameter, at which $Q_3 \approx 63.2\%$) and the steepness of the distribution, respectively. Linearizing this equation gives:

$$\ln\left(\ln\left(\frac{1}{1-Q_3}\right)\right) = n\ \ln(x) - n\ \ln(d'). \tag{3-4}$$

This linearized form enables an easy fitting of a given size distribution to determine the parameters of the distribution function. We have used such a procedure to allow a compact quantitative comparison of simulation results, for example.

The particle size distributions and the characteristic parameters of the RRSB distribution function for the feed used in the cross-flow and the air classifier simulations are summarized in Section 4.1 and 5, respectively.

### 3.1.3. Reynolds Number

The Reynolds number is a dimensionless number that characterizes the flow regime and is defined as:

$$Re = \frac{|\mathbf{u}|\ L\ \rho_f}{\eta_f}. \tag{3-5}$$

Here $\mathbf{u}$, $L$ and $\eta_f$ are a characteristic velocity, a characteristic dimension (e.g., the diameter of a spherical particle), the density and the dynamic viscosity of the fluid. To characterize the flow at the classifier inlet, a Reynolds number can be defined as:

$$Re_{inlet} = \frac{Q_f\ d_{h,inlet}}{A_{inlet}} \frac{\rho_f}{\eta_f}, \tag{3-6}$$

where $d_{h,inlet}$ and $A_{inlet}$ are the hydraulic diameter and the cross-sectional area of the inlet channel, respectively.

A dispersed-phase Reynolds number can be introduced by utilizing the difference between the fluid and the particle velocity $\mathbf{u}_{rel} = \mathbf{u}_f - \mathbf{u}_p$:

$$Re_p = \frac{|\mathbf{u}_{rel}| \; d_p \; \rho_f}{\eta_f}. \tag{3-7}$$

The above definition requires the knowledge of the fluid-particle relative velocity, for which assumptions need to be made. Specifically, the particle Reynolds numbers summarized in Section 4.1 are the Reynolds numbers of freely-sedimenting particles, i.e., we have used the terminal settling velocity as the slip velocity. This velocity can be obtained from the drag law of Beetstra et al. (cp. Section 3.3) in the limit of infinite dilution.

### 3.1.4. Stokes Number

The Stokes number is an important dimensionless number, since it quantifies the ability of particles to follow the fluid motion. It is defined as the ratio of the characteristic relaxation time of a particle and a characteristic flow time:

$$Stk = \frac{\tau_{relax}}{\tau_f}. \tag{3-8}$$

Here $\tau_{relax}$ is the particle relaxation time, which characterizes the time a particle needs to adapt to changes in the surrounding flow field. For low Reynolds number flow around a particle, it can be calculated using:

$$\tau_{relax} = \frac{{d_p}^2 \rho_p}{18 \; \eta_f}. \tag{3-9}$$

The characteristic time scale for the flow field is related to a characteristic dimension and a fluid velocity, i.e.,

$$\tau_f = \frac{L}{u_f}. \tag{3-10}$$

The parameters used in the above relationship depend on the flow configuration under consideration. For example, for a *JICF* configuration, be the jet diameter width and the cross-flow velocity can be used. At low Stokes numbers ($Stk << 1$), the particles' velocity will be close to the velocity of the fluid. In such a case, the relaxation time is much smaller than the characteristic time of the fluid flow field (i.e., the particles respond quickly to changes in the flow field). In contrast, if the Stokes number is much greater than unity, the particles' velocity

will differ significantly from the fluid velocity, since particles are not able to adapt fast enough to changes in the flow field [2].

### 3.1.5. Terminal Settling Velocity

The terminal settling velocity and the Stokes drag are introduced below, since these quantities are often used as reference parameters. A particle moving relative to the surrounding fluid is subjected to different types of forces. The particle will accelerate until the individual forces balance each other, and the total force is zero. At this steady state, the particle moves with a constant velocity. Considering an isolated sedimenting spherical particle under the influence of gravity, the terminal settling velocity can be calculated:

$$u_t = \sqrt{\frac{4}{3}\frac{(\rho_p - \rho_f)\, d_p\, g}{\rho_f\, C_D}}\,. \tag{3-11}$$

Here $\rho_p$, $\mathbf{g}$, $C_D$ and $A_{pr}$ are the particle density, the gravitational acceleration, the drag coefficient and the projected area of the particle respectively. For low particle Reynolds numbers (i.e., $Re_p < 0.1$), Stokes' drag law is valid, leading to:

$$C_{D,Stokes} = \frac{24}{Re_p}\,. \tag{3-12}$$

$$u_{t,Stokes} = \sqrt{\frac{d_p{}^2\, g\, (\rho_p - \rho_f)}{18\, \eta_f}}\,. \tag{3-13}$$

Schiller and Naumann early reported a more accurate relationship for the drag coefficient at higher Reynolds numbers ($Re_p \leq 1000$) [2]:

$$C_D = \frac{24}{Re_p}\left(1 + 0.15\, Re_p{}^{0.687}\right). \tag{3-14}$$

Above equation accounts for the non-linear correlation between the drag coefficient and the Reynolds number in this transitional range.

### 3.1.6. Sedimentation Velocity in a Centrifugal Field

When a particle is moving in a rotating flow, the centrifugal force acting on the particle has to be considered. Assuming that the particle is moving with a tangential velocity $u_\phi$ on a circular orbit with radius *R* from the center of fluid's rotational motion, the centrifugal force is:

$$f_{cf} = \left(\rho_p - \rho_f\right) \frac{d_p{}^3 \pi}{6} \frac{u_\phi{}^2}{R}. \tag{3-15}$$

The force balance of the drag and the centrifugal force in case of insignificant Coriolis forces and negligible gravity yields:

$$u_{t,cf} = \sqrt{\frac{4}{3} \frac{(\rho_p - \rho_f)\, d_p}{\rho_f\, C_D} \frac{u_\phi{}^2}{R}}. \tag{3-16}$$

Here the only difference to the terminal settling velocity in a gravitational field without centrifugal force is the acceleration term due to circular motion (see equation (3-11)). For above relationship it is often assumed that the particles move at the same tangential velocity as the surrounding fluid (i.e., quasi-stationary conditions). This assumption is conflicted in case of high mass loadings, where a local deceleration of the fluid due to the presence of particles is observed.

### 3.1.7. Coarse Graining Ratio

In order to reduce the number of particles to be tracked, "parcels" (computational particles) that represent a cloud of many physical particles are often used in simulations. The diameter of the parcel (i.e., $d_p$) divided by the physical particle size (i.e., $d_{prim}$) is the coarse graining ratio (CG). Using this parameter offers the advantage of a reduced computational time. However, employing a too high coarse graining ratio may lead to inadequate results, as the physical flow pattern cannot be mimicked appropriately when tracking too less parcels. In the present work the effect of this numerical parameter has been determined by performing simulations with different coarse graining ratios. The obtained results are summarized in Section 4.3.1.

## 3.2. Governing Equations

In this study the modelling of the gas-particle flow is done by using an Euler-Lagrange approach. When using this approach, the behavior of the fluid is described in an Eulerian reference frame by employing the finite volume method [3]. The particle motion is tracked explicitly based on a Lagrangian treatment. This combination of computational fluid dynamics (CFD) and discrete element method (DEM) is referred to as the CFD-DEM [4], which is increasingly used to study gas-particle flows [5].

### 3.2.1. Fluid

Since the particles occupy a certain volume in the Eulerian reference frame (i.e., each computational cell), the volume fraction of the fluid $\phi_f$ must be accounted for in the Navier-Stokes equations. Also, the Navier-Stokes equations are expressed in terms of local averaged variables (i.e., an average over a computational cell), because fluid flow around individual particles is not resolved within the classical CFD-DEM approach [6]. In summary, the equations for the incompressible fluid phase are:

$$\frac{\partial\left(\phi_f \rho_f\right)}{\partial t} + \nabla \cdot \left(\phi_f \rho_f \mathbf{u}_f\right) = 0, \tag{3-17}$$

$$\frac{\partial\left(\phi_f \rho_f \mathbf{u}_f\right)}{\partial t} + \nabla \cdot \left(\phi_f \rho_f \mathbf{u}_f \mathbf{u}_f\right) = -\phi_f \nabla p_f - \phi_f \nabla \cdot \boldsymbol{\tau}_f + \phi_f \rho_f \mathbf{g} - \boldsymbol{\Phi}_{fp}. \tag{3-18}$$

In the equations above, $p_f$, $\boldsymbol{\tau}_f$ and $\boldsymbol{\Phi}_{fp}$ are the fluid pressure, the viscous stress tensor and the specific fluid-particle interaction force (i.e., the coupling force per unit volume of the mixture exerted by the particles on the fluid phase), respectively. Note that $\boldsymbol{\Phi}_{fp}$ comprises only the fluid-particle drag force, and not the pressure gradient (i.e., buoyancy) force.

### 3.2.2. Particles

The modelling of the particle phase is governed by the equation of translational and rotational motion:

$$m_i \frac{d\left(\mathbf{u}_{p,i}\right)}{dt} = \mathbf{f}_{fp,i} + \sum_j \mathbf{f}_{ij} + m_i \mathbf{g}, \tag{3-19}$$

$$I_i \frac{d(\boldsymbol{\omega}_i)}{dt} = \sum_j \mathbf{t}_{ij} \quad . \tag{3-20}$$

Here, the involved forces are the fluid-particle interaction force $\mathbf{f}_{fp,i}$ exerted on particle *i*, the sum of contact forces due to collisions with neighboring particles $\mathbf{f}_{ij}$, as well as gravity. *I*, $\boldsymbol{\omega}$, and $\mathbf{t}_{ij}$ are the moment of inertia, the angular velocity and the torque due to collisions, respectively. Collisional forces and torques are computed with a simple linear spring-dashpot model [4] for most of the simulations. This allows the modeling of dense particle layers that might form in the classifier. For the simulation of particle motion between blades, however, we used a simple hard-sphere collision model that assumes an instantaneous interaction of the particles and the walls. This was necessary due to the immense impact speed of the particles on the wall. In these blade flow simulations, the collisions between particles were not tracked.

## 3.3. Drag Model

To obtain the drag force beyond Stokes flow conditions (i.e., in the infinitely dilute limit and at zero particle Reynolds number), an appropriate drag law has to be used. For the simulations presented in this thesis, the drag law developed by Beetstra et al. [7] has been employed. Specifically, the drag force acting on each particle is modeled as [8]:

$$\mathbf{f}_{D,i} = V_{p,i}\ \beta_{p,i}\ \left(\mathbf{u}_{f,i} - \mathbf{u}_{p,i}\right). \tag{3-21}$$

The drag coefficient of particle $i$ is computed by:

$$\beta_{p,i} = \frac{18\ \eta_f}{d_{p,i}^2}\ \left(1-\phi_p\right)\ \left[F_0\left(\phi_p\right) + G_{0,i}\left(\phi_p, Re_{p,i}\right)\right]. \tag{3-22}$$

Here, the term in the square brackets is a dimensionless drag force (i.e. normalized with respect to the Stokes drag). The first term is defined as:

$$F_0\left(\phi_p\right) = \frac{10\ \phi_p}{\left(1-\phi_p\right)^2} + \left(1-\phi_p\right)^2\left(1+1.5\ \sqrt{\phi_p}\right), \tag{3-23}$$

and accounts for the normalized drag force in the limit of Stokes flow (i.e., $Re \to 0$) in a dense gas-particle flows (i.e., $\phi_p > 0$). Hence, the above term yields unity in case the particle volume fraction approaches zero. The second function depends on the particle Reynolds number:

$$G_{0,i}\left(\phi_p, Re_{p,i}\right) = \frac{0.413 Re_{p,i}}{24\ \left(1-\phi_p\right)^2}\left[\frac{\left(1-\phi_p\right)^{-1} + \ 3\phi_p\ \left(1-\phi_p\right) + 8.4 Re_{p,i}^{-0.343}}{1\ +\ 10^{3\phi_p}\ Re_{p,i}^{-(1+4\phi_p)/2}}\right], \tag{3-24}$$

with the particle Reynolds number defined as:

$$Re_{p,i} = \frac{\left(1-\phi_p\right)\ \left(\mathbf{u}_{f,i} - \mathbf{u}_{p,i}\right)\ d_{p,i}\ \rho_f}{\eta_f}. \tag{3-25}$$

## 3.4. Turbulence Model

Turbulent flows are characterized by fluctuating velocity fields and a large spectrum of length and time scales. Hence, the direct simulation of the flow field would require an extremely fine computational mesh and small time steps, which is infeasible for the industrial application under consideration. To be able to mimic the effect of velocity and pressure fluctuations at reasonable computational costs, an adequate model has to be employed.

### 3.4.1. Large Eddy Simulation

In so-called Large Eddy Simulations (LES) a low-pass filter is employed to separate large scale from small scale fluctuations. The filter function is designed to eliminate eddies that are smaller than a certain filter width Δ, while the larger scales are retained and directly resolved. The eliminated and hence unresolved subgrid-scale fluctuations have to be modeled. The filtering, implicit to the computational grid used in a LES, results in a decomposition of the flow field in a filtered term and a term that describes the residual fluctuations $u_i^{SGS}\left(x,t\right)$:

$$u_i\left(x,t\right) = \overline{u}_i\left(x,t\right)\ + u_i^{SGS}\left(x,t\right). \tag{3-26}$$

Several approaches are used for the modelling of the residual fluctuation term, one of which is an eddy viscosity concept. For the simulations of the cross-flow particle injection a model referred to as the "one equation eddy viscosity" model has been used. Here, a transport equation for the subgrid-scale kinetic energy $k$ [9] must be solved in order to obtain the subgrid-scale viscosity $\nu_{SGS}$ as:

$$\nu_{SGS} = C_k\ \sqrt{k}\ \Delta, \tag{3-27}$$

where $C_k$ is a model constant. This model is denoted as "oneEqEddy" in the remaining parts of this work. The filtered momentum balance for incompressible flows without body forces is [10]:

$$\frac{\partial \overline{u}_j}{\partial t} + \frac{\partial \overline{u}_i \overline{u}_j}{\partial x_i} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_j} + \nu \frac{\partial^2 \overline{u}_j}{\partial x_i \partial x_i} - \frac{\partial \tau_{ij}^{SGS}}{\partial x_i}. \tag{3-28}$$

Here, $\rho$ and $\nu$ are the density and the kinematic viscosity of the fluid, $\tau_{ij}^{SGS}$ is the residual stress tensor (subgrid stress tensor) and $\overline{p}$ is the filtered pressure field.

The difference to the Navier-Stokes equation results from the subgrid-scale stress tensor which is defined as:

$$\tau_{ij}^{SGS} = \overline{u_i u_j} - \overline{u}_i \overline{u}_j. \tag{3-29}$$

This tensor describes the momentum transfer by turbulence at scales smaller than the filter width, i.e., the computational cell [3]. In order to estimate the subgrid stress tensor, only the deviatoric part of the subgrid stress tensor is modelled by using the subgrid-scale viscosity:

$$\tau_{ij}^{SGS,dev} = \tau_{ij}^{SGS} - \frac{1}{3}\tau_{ii}^{SGS}\delta_{ij} = -2\,\nu_{SGS}\,\overline{S}_{ij}. \tag{3-30}$$

Here, $\delta_{ij}$ is the Kronecker delta tensor, and $\overline{S}_{ij}$ is the filtered strain rate tensor calculated by:

$$\overline{S}_{ij} = \frac{1}{2}\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right). \tag{3-31}$$

For the simulations of the air classifier, the Smagorinsky model is used to close the above subgrid-scale models. Specifically, the subgrid-scale viscosity is modelled as:

$$\nu_{SGS} = \left(C_s\,\Delta\right)^2 \left|\overline{S}\right|, \tag{3-32}$$

where $C_s$ is the Smagorinsky coefficient and $\Delta$ is the filter width [10] (the latter can be computed from the volume of a computational grid cell). The subgrid-scale viscosity is proportional to the filtered rate of strain calculated by:

$$\left|\overline{S}\right| = \sqrt{2\overline{S}_{ij}\overline{S}_{ij}}. \tag{3-33}$$

The above model is valid in the bulk of the flow, where the subgrid-scale eddies are isotropic. However, turbulent fluctuations (and consequently subgrid-scale eddies) are damped near solid walls [11], and above model loses its validity. This is due to the dominance of viscous effects in wall-bounded regions. In order to model the flow near boundaries adequately, the damping effect must be considered. One approach is to use a particular damping function, e.g.

the van Driest damping, which results in a reduction of the turbulent viscosity near boundaries. Other approaches are (i) local grid refinement near the wall to directly resolve the damping effect, or (ii) the use of wall functions that assume a certain velocity profile near the wall. For the simulations in this work the latter approach is used, which is known to deliver reliably results [9].

### 3.4.2. Reynolds-Averaged Simulation

Due to the extremely small size of the classifier's rotor blades in combination with high velocity gradients, the flow in the rotor region cannot be resolved directly as it is computationally demanding. Hence, for the simulations of the flow between the rotor blades a model based on the Reynolds-Averaged Navier-Stokes (RANS) equations was employed. As RANS models are generally less demanding in terms of computational resources than LES models, the *k*-*ω*-SST model [12] was used, which provides acceptable predictive capabilities at reasonable computational time. For RANS simulations the characteristic variables are decomposed in a time-averaged value $\langle u_i(x)\rangle$ and a fluctuating value:

$$u_i(x,t) = \langle u_i(x)\rangle + u_i'(x,t) \tag{3-34}$$

Applying this decomposition in the Navier-Stokes equations leads to the RANS equations for incompressible flows [13]:

$$\frac{\partial \rho\langle u_i\rangle}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho\langle u_i\rangle\langle u_j\rangle + \rho\langle u_i'u_j'\rangle\right) = -\frac{\partial\langle p\rangle}{\partial x_i} + \frac{\partial\langle \tau_{ij}\rangle}{\partial x_j}, \tag{3-35}$$

with the mean viscous stress tensor being:

$$\langle \tau_{ij}\rangle = \eta\left(\frac{\partial\langle u_i\rangle}{\partial x_j} + \frac{\partial\langle u_j\rangle}{\partial x_i}\right). \tag{3-36}$$

Here, $\rho\langle u_i'u_j'\rangle$ are the Reynolds stresses and $\eta$ is the dynamic viscosity.

In order to obtain the Reynolds stresses, the *k*-*ω*-SST model is used. In this model the equations for the turbulent kinetic energy *k* and the specific dissipation of turbulent kinetic energy *ω* are solved close to walls (i.e., in the boundary layers), whereas the *k*-*ε* model is used elsewhere. By combining these two models (i.e., shear stress transfer formulation) the sensitivity of the *k*-*ω* model to free stream turbulence is eliminated as it is only used in regions close to the walls [12].

As described in the preceding section, turbulence models without certain near wall treatment are not suitable in the near wall regions. Hence, in the RANS-simulations presented in this work, the model equations in the near boundary region are not solved and particular wall functions are employed instead. The wall functions are providing boundary conditions for the turbulence model at a grid point far from the wall (see Pope [10] for more details). This of course results in some loss in accuracy. Given that the above models have been derived for single phase turbulent flow, and are strictly speaking not applicable to comparably dense gas-particle flows for which still reliable models are missing, the use of wall function appears to be suitable.

## *3.5. References*

[1] C.T. Crowe, J.D. Schwarzkopf, M. Sommerfeld, Y. Tsuji, Properties of Dispersed Phase Flows, in: Multiph. Flows with Droplets Part. Second Ed., CRC Press, Taylor & Francis Group, 2011: pp. 17–38.

[2] C.T. Crowe, E.E. Michaelides, Basics Concepts and Definitions, in: Multiph. Flow Handb., CRC Press, Taylor & Francis Group, 2006: pp. 5–6.

[3] B. Andersson, R. Andersson, L. Håkansson, M. Mortensen, R. Sudiyo, B. van Wachem, Computational Fluid Dynamics for Engineers, in: Cambridge University Press, 2012: pp. 79–81.

[4] C. Goniva, C. Kloss, A. Hager, S. Pirker, An Open Source CFD-DEM Perspective, in: Proc. OpenFOAM Work., 2010: pp. 1–10.

[5] Z.Y. Zhou, S.B. Kuang, K.W. Chu, A.B. Yu, Discrete particle simulation of particle–fluid flow: model formulations and their applicability, J. Fluid Mech. 661 (2010) 482–510.

[6] B.H. Xu, A.B. Yu, Numerical simulation of the gas-solid flow in a fluidized bed by combining discrete particle method with computational fluid dynamics, Chem. Eng. Sci. 52 (1997) 2785–2809.

[7] R. Beetstra, M.A. Van Der Hoef, J.A.M. Kuipers, Drag Force of Intermediate Reynolds Number Flow Past Mono- and Bidisperse Arrays of Spheres, AIChE J. 53 (2007) 489–501.

[8] S. Radl, M. Girardi, S. Sundaresan, Effective Drag Law for Parcel-Based Approaches - What can we learn from CFD-DEM?, in: Eur. Congr. Comput. Methods Appl. Sci. Eng., Vienna, Austria, 2012.

[9] E. De Villiers, The Potential of Large Eddy Simulation for the Modeling of Wall Bounded Flows, 2006.

[10] S.B. Pope, Turbulent Flows, Cambridge University Press, 2000.

[11] J. Fröhlich, Statistische Turbulenzmodellierung, in: Large Eddy Simul. Turbul. Strömungen, 1. Auflage, B.G. Teubner Verlag, 2006: pp. 49–55.

[12] A. Hellsten, Some Improvements in Menter's k-ω SST Turbulence Model, in: 29th AIAA Fluid Dyn. Conf., American Institute of Aeronautics and Astronautics, Albuquerque, New Mexico, 1997: pp. 1–11.

[13] J.H. Ferziger, M. Perić, Computational Methods For Fluid Dynamics, in: 3rd ed., Springer-Verlag, 2002: pp. 292–304.

# 4. Cross-Flow Particle Injection

## 4.1. Setup

Simulations of particle injection in a pseudo-two-dimensional cross-flow have been performed to identify the effects of different particle feeding conditions on particle dispersion. The cross-flow configuration studied in this chapter mimics the key features of the particle injection device in the classifier: particles are fed in a comparably dense state and with comparably low velocity into a turbulent air flow void of particles.

A two-dimensional setup has been chosen, because it allows the rapid screening of key influence parameters. Also, this setup allows us to perform grid-refinement studies to identify the effect of grid and parcel size on the simulation results.

The physical parameters of the base case for this setup are summarized in Table 4-1.

**Table 4-1: Physical parameters of the cross-flow particle injection base case.**

| *Parameters* | *Value* |
|---|---|
| Particle density ($\rho_p$) | 2.500 [kg/m³] |
| Gas density ($\rho_f$) | 1.1 [kg/m³] |
| Gas dynamic viscosity ($\eta_f$) | $1.91 \cdot 10^{-5}$ [Pa·s] |
| Gravity (**g**) | (0 0 0) [m/s²] |
| Mass loading ($\mu_{\text{inlet}}$) | 1.0 |
| Domain length($x_{domain}$) | 1.8 [m] |
| Domain with ($y_{domain}$) | 0.7 [m] |
| Injection position ($x_{inject}$) | 0.3 [m] |
| Particle injection angle ($\alpha_p$) | 90 [°] |
| Particle injection velocity ($\mathbf{u}_{p,y,0}$) | 10.0 [m/s] |
| Cross-flow velocity ($\mathbf{u}_{inlet,x}$) | 25.4 [m/s] |
| $Re$ | $1.0 \cdot 10^{6}$ |
| $Re_p$ | 0.1 … 26.5 |

The simulation domain is characterized by its width $y_{domain}$ and length $x_{domain}$ as illustrated in Figure 4-1. The depth of the simulation domain in this pseudo-two-dimensional cross-flow is

one computational cell (i.e., $z_{domain}$ = 0.002 [m]). Air at ambient conditions enters the channel with the cross-flow velocity $\mathbf{u}_{inlet,x}$. At the bottom wall, perpendicular to the mean flow direction of the air, the polydisperse feedstock is injected with the velocity $\mathbf{u}_{p,y,0}$. The particle injection point is located at a distance $x_{inject}$ away from the air inlet.

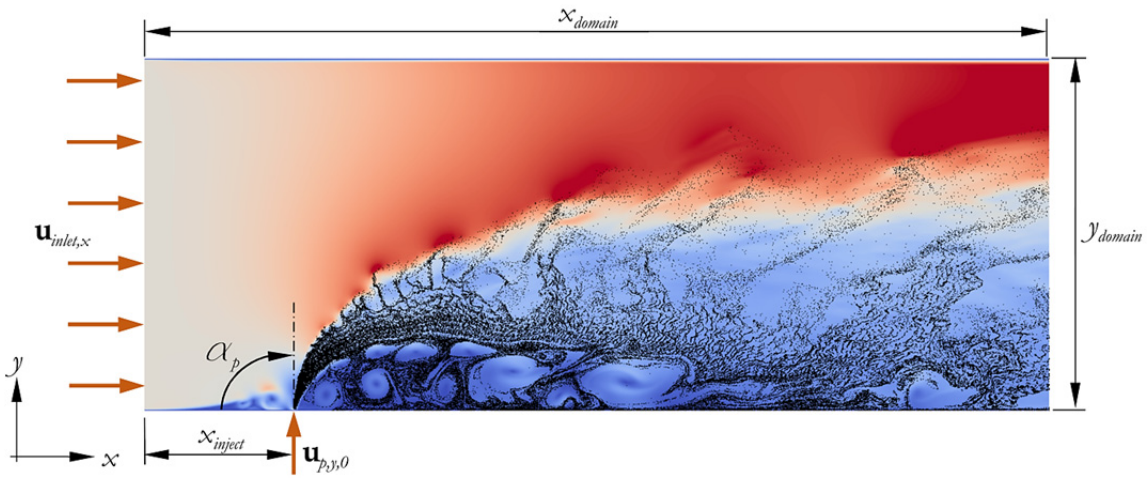


**Figure 4-1: Sketch of the simulation domain of the cross-flow particle injection case.**

The particle size distribution of the feedstock is provided in Table 4-2. Here, the volume fraction of each particle class is chosen in a manner which yields a uniformly distributed particle feedstock in terms of the number of particles (i.e. the number fraction of each particle class equals 10%).

**Table 4-2: Feed particle size distribution used for the cross-flow particle injection simulations**

| *No.* | $d_{prim}$ [µm] | $Q_3$ | $\Delta Q_3$ |
|---|---|---|---|
| 1 | 50 | 1.000 | 0.501 |
| 2 | 40 | 0.499 | 0.257 |
| 3 | 30 | 0.242 | 0.108 |
| 4 | 24 | 0.134 | 0.055 |
| 5 | 20 | 0.079 | 0.032 |
| 6 | 18 | 0.047 | 0.023 |
| 7 | 15 | 0.024 | 0.014 |
| 8 | 12 | 0.01 | 0.007 |
| 9 | 8 | 0.003 | 0.002 |
| 10 | 6 | 0.001 | 0.001 |

The corresponding parameters of the RRSB function for the particle size distribution of the feed are $d'$ = 43.08 [µm] and $n$ = 3.49.

For the simulations of the cross-flow, gravity is neglected as it does not affect the penetration depth $y_{pen}$ of the particles. The calculated penetration depth at the end of the channel for three different particle diameters is summarized in Table 4-3.

**Table 4-3: Analysis of gravity effects on theoretical penetration depth at the end of the channel assuming Stokes' drag law and an undisturbed background flow.**

| $d_{prim}$ [µm] | $y_{pen}$ [m] without gravity | $y_{pen}$ [m] with gravity |
|---|---|---|
| 50 | 0.2687 | 0.2582 |
| 20 | 0.0436 | 0.0419 |
| 6 | 0.0039 | 0.0038 |

The penetration depth is calculated by considering a simple force balance. As can be seen, gravitational effects are insignificant, as the comparison of the penetration depth for the two cases reported in Table 4-3 is revealing minor deviations.

The most important numerical parameters for the cross-flow simulations are given in Table 4-4. To determine the DEM time step $\Delta t_{DEM}$, a contact time $t_c$ as explained by Luding [1] is used. Further information regarding the setup of these simulations is presented in the Appendix (see Section 10).

**Table 4-4: Numerical parameters of the cross-flow particle injection base case.**

| ***Parameters*** | ***Value*** |
|---|---|
| $Co$ | < 0.55 |
| $\Delta x/d_p$ | 8 |
| Discretization schemes | 2nd order Euler / limitedLinear |
| Collision tracking | Yes (DEM) |
| Coarse graining ratio (CG) | 5 |
| $\Delta t_{DEM}/t_c$ | 1/50 |
| $\Delta t_{DEM}$ | $6 \cdot 10^{-8}$ [s] |
| LES turbulence model | oneEqEddy |

## 4.2. Theoretical Aspects

### 4.2.1. Mass Averaged Particle Trajectories and Standard Deviation

Mass averaged particle trajectories of the particles in the cross-flow configuration are calculated to illustrate the mean particle motion in the channel. The trajectories are obtained by recording the mass and the position of the particles. The simulation domain is divided into 90 bins along the *x*-axis, which enables a sufficiently high spatial resolution of mean particle properties. The spatially-averaged data is then used for computing the mean particle trajectories:

$$\overline{y_{i,j}} = \frac{\overline{m_{i,j}\, y_{i,j}}}{\overline{m_{i,j}}} \tag{4-1}$$

Here, the subscripts *i* and *j* refer to the particle class and the bin number, respectively. The numerator on the right hand side of above equation represents the average product of the particle mass *m* and the particle *y*-position.

For calculating the standard deviation of the mean particle position in a certain region of the simulation domain, it is required to record the mass and the position of the particles located in the region. The mean *y*-position of each particle class in this region can be calculated as shown in Equation (4-1). The standard deviation of the mean *y*-position $\sigma_y$ can be computed via:

$$\sigma_{y,i} = \sqrt{\frac{\sum_k m_{i,k}\left(y_{i,k} - \overline{y_i}\right)}{\sum_k m_{i,k}}}\,. \tag{4-2}$$

### 4.2.2. Theoretical Penetration Depth

In order to compare the obtained results with theory, we have computed a theoretical particle trajectory and penetration depth. The penetration depth for spherical particles injected in an undisturbed gas flow can be calculated by considering a force balance on a particle under the assumption of an undisturbed background flow:

$$m_p \cdot \frac{d\mathbf{u}_{p,y}}{dt} = -\,3\,\pi\,\eta\,d_p\,\mathbf{u}_{p,y}\,. \tag{4-3}$$

Here, the term on the left-hand side represents the particle's inertia, while the right-hand side represents Stokes' drag force. Introducing the terminal settling velocity for Stokes flow $u_{t,Stokes}$ (see Section 3.1.5 for details) and a constant defined as

$$\tilde{k} = \frac{\left|\rho_p - \rho_f\right| \mathbf{g}}{\rho_p}, \tag{4-4}$$

double integration, yields:

$$y = \frac{u_{t,\text{Stokes}}\, u_{p,y,0}}{\tilde{k}} \left(1 - \exp\left(-\frac{\tilde{k} \cdot t}{u_{t,\text{Stokes}}}\right)\right). \tag{4-5}$$

Above equation gives the penetration depth at time $t$, where $u_{p,y,0}$ is the initial particle velocity in $y$-direction (it was assumed that the particle is injected at $t = 0$ and $y = 0$).

A similar approach is employed for calculating the particle position in the $x$-direction. The force balance gives:

$$m_p \cdot \frac{d\mathbf{u}_{p,x}}{dt} = -3\,\pi\,\eta\,d_p\,\left|\mathbf{u}_{p,x} - \mathbf{u}_{f,x}\right|. \tag{4-6}$$

Double integration of this equation, and assuming appropriate initial conditions (i.e., injection at $x = 0$ and $\mathbf{u}_{p,x} = 0$), yields:

$$x = \frac{u_{t,\text{Stokes}}\, u_{cross}}{\tilde{k}} \left(\frac{\tilde{k} \cdot t}{u_{t,\text{Stokes}}} + \exp\left(-\frac{\tilde{k} \cdot t}{u_{t,\text{Stokes}}}\right) - 1\right). \tag{4-7}$$

A comparison of the calculated particle trajectories applying a drag force according to Stokes law and the simulation results is shown in Figure 4-2.

Apparently, the theoretical approach is only applicable (to a first approximation) for comparably large particles. For smaller particles, there are significant deviations compared to the simulation results (compare the results for $d_p^*$ and $d_{p,min}$; the theory strongly underpredicts the penetration into the flow). The reason for this is that the analytical method neglects the disturbance of the gas flow due to the injected particles, as well as particle-particle interactions. Also, the applied drag law is not appropriate for the given flow conditions, as the particle Reynolds number is *O(25)* for the largest particles (see Table 4-1). Furthermore, it is assumed that there is no fluid velocity in vertical direction when calculating the theoretical penetration depth, which is clearly not the case in the current situation as illustrated in Figure 4-3.

Additionally, the entrainment of small particles by the particle cloud accounts for the huge difference in the predicted particle trajectories for the small particles.

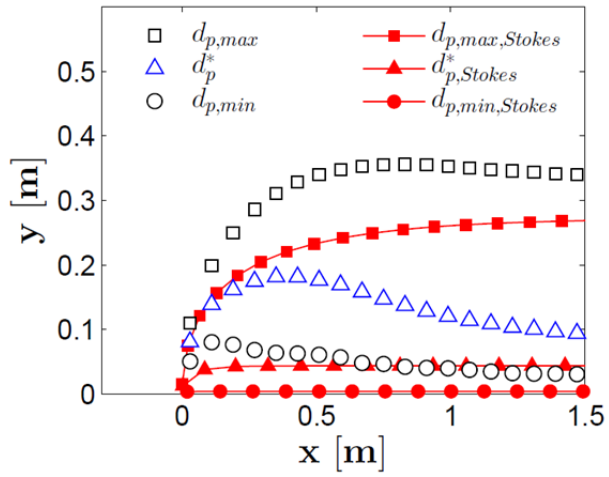


**Figure 4-2: Calculated particle trajectories applying Stokes' law vs. simulation results (CG = 10, particle injection velocity = 15 [m/s]; for all other parameters see the base case definition in Section 4.1).**

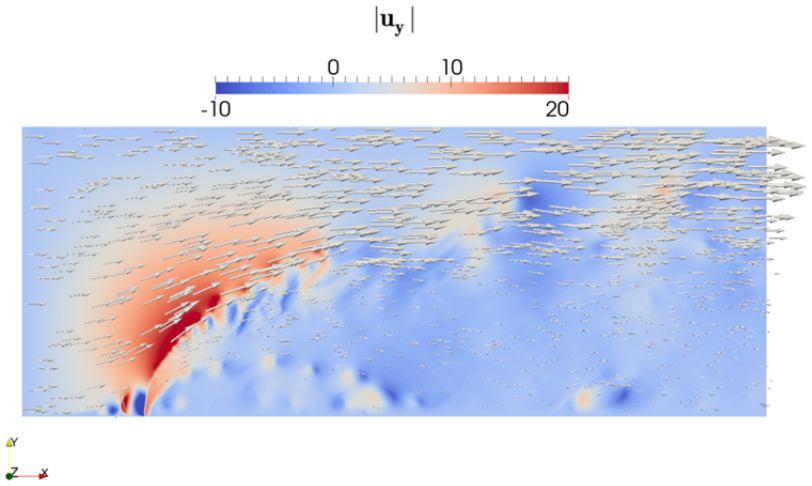


**Figure 4-3: Flow field normal to the main flow direction of the cross-flow channel injection (shown is the gas velocity in the *y*-direction and arrows indicating the local flow; particles are not shown; CG = 10, particle injection velocity = 15 [m/s]; for all other parameters see the base case definition in Section 4.1).**

Above illustration shows, that the inflowing air is substantially redirected by the injected particles. This leads to a vector component in vertical direction (i.e., the *y*-axis) of the air velocity. Also, it becomes obvious that the injection of particles is significantly decreasing the magnitude of the fluid velocity in a large region behind the injection point, while the air at the top of the channel is accelerated to velocities much higher than the fluid's inlet velocity.

### 4.2.3. Approach to Steady State

Another important parameter is the behavior of the particle as a function of time. This parameter has been analyzed by considering the time evolution of the mean *y*-position of the particles in a region between 0.7 [m] < $x$ < 0.8 [m] downstream of the injection point (see Figure 4-4).

After about 0.3 [s] the particle positions in this region can be considered as steady, since there are limited fluctuations about a mean value. A typical time for the particles to reach this position is ca. 0.03 [s], indicating that the flow is developed after about 10 dimensionless time units. The standard deviation of the particles' *y*-position for all except the smallest particle class is on average ca. 0.12 [m], which is about 17% of the channel width. Hence, large particles particles are well dispersed over the channel width. On the contrary, very small particles are moving close to the bottom wall where the air velocity is low. This results in a narrow distribution of the particles' position, and consequently a low standard deviation of their position (see Figure 4-4b).

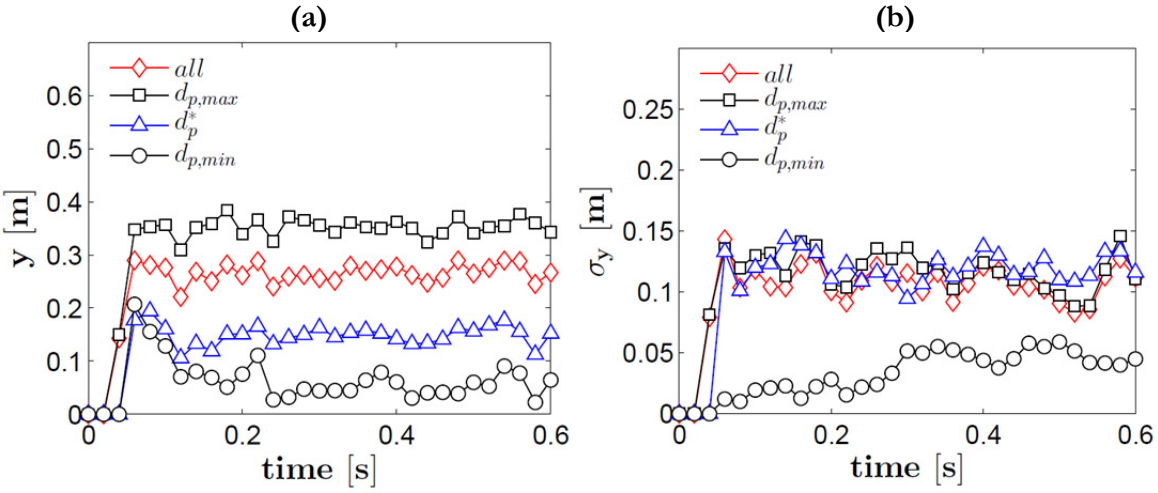


**Figure 4-4: (a) Mean *y*-position of the particles between 0.7 [m] < $x$ < 0.8 [m] downstream of the injection point, (b) standard deviation of the particle position (CG = 10, particle injection velocity = 15 [m/s]; for all other parameters see the base case definition in Section 4.1).**

## 4.3. *Effect of Numerical Parameters*

The simulations of the cross-flow particle injection are used to identify the effect of various key numerical parameters such as the coarse graining ratio (CG) and the mesh size.

### 4.3.1. Coarse Graining Ratio

Simulations with three different coarse graining ratios have been performed to assess the influence of this parameter on the simulation results. A qualitative comparison of velocity flow fields obtained from simulations with different coarse graining ratios is illustrated in Figure 4-5.

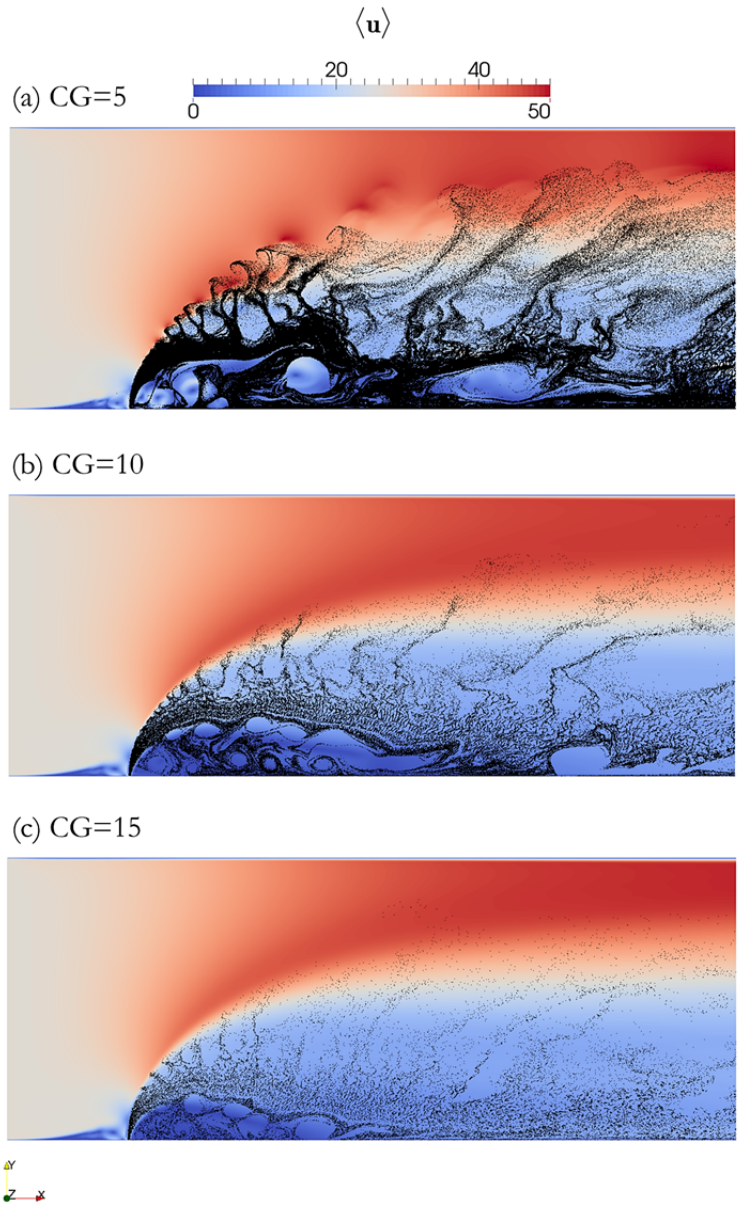


**Figure 4-5: Time averaged air velocity fields for different coarse graining ratios and a particle injection velocity of 15 [m/s]; (a) CG = 5, (b) CG = 10, (c) CG = 15 (for all other parameters see the base case definition in Section 4.1).**

The coarse graining ratio is an important numerical parameter that must be used for simulations with small particles which cannot be tracked individually. Clearly, by employing a higher coarse graining ratio, fewer particles are tracked (cp. Section 3.1.7). As can be seen in Figure 4-5, using a smaller coarse graining ratio (top panel in Figure 4-5) yields a qualitatively similar flow field compared to the simulations that are tracking less parcels (panel (b) and (c) in Figure 4-5).

The illustration of the mean (i.e., mass averaged) particle trajectories for different particle classes and coarse graining ratios gives a more quantitative comparison (see Figure 4-6).

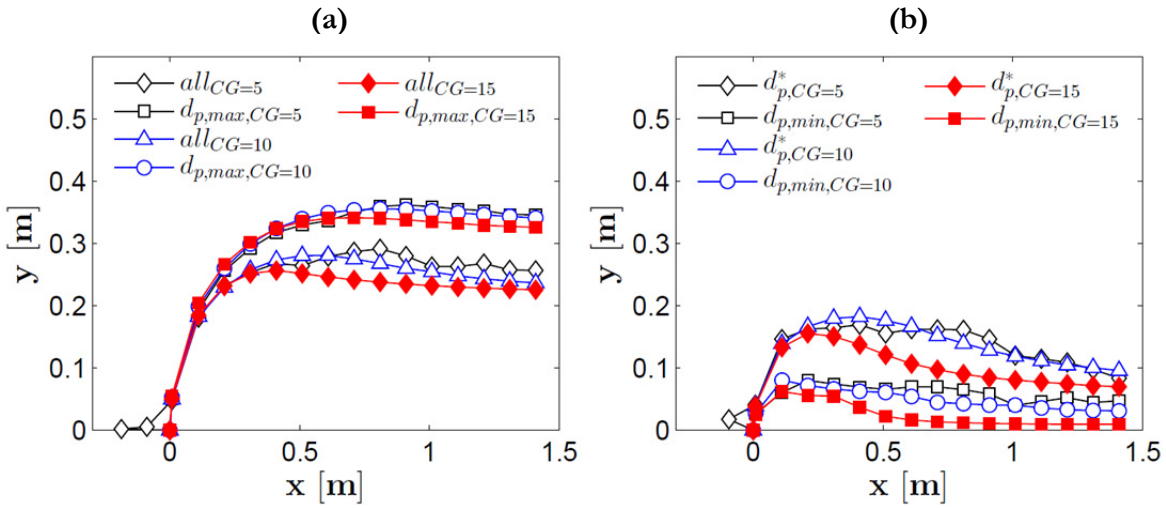


**Figure 4-6: Mass averaged particle trajectories for different particle classes and coarse graining ratios (particle injection velocity = 15 [m/s], (a) all particles and particles with $d_{p,max}$ = 50 [µm], (b) particles with $d_p^*$ = 20 [µm] and $d_{p,min}$ = 6 [µm]; (for all other parameters see the base case definition in Section 4.1).**

The simulations show that the particle distribution is qualitatively similar for the different coarse graining ratios. However, by employing a too large coarse graining ratio (i.e., CG = 15) the obtained particle trajectories are noticeably differing from the other simulations (CG = 5 and CG = 10). Specifically, an underprediction of the penetration depth (pronounced for comparably small particles) can be observed (cp. panel (b) in Figure 4-6).

The relative mean deviation of the particle trajectories with respect to the channel with is summarized in Table 4-5. Here, the relative mean deviation of the simulations employing a small (CG = 5) and a large (CG = 15) coarse graining ratio from the case using a CG of 10 is shown. As can be seen, the deviation is always less than 5% of the channel width, which seems acceptable given other uncertainties (e.g., fluctuations in the air inflow).

**Table 4-5: Relative mean deviation (in per cent of the channel width) to the particle trajectories of the case with CG = 10.**

| $d_{prim}$ [µm] | RMD [%] CG = 5 | RMD [%] CG = 15 |
|---|---|---|
| 50 | 0.88 | 1.54 |
| 20 | 1.42 | 4.91 |
| 6 | 1.75 | 3.57 |

As has already been indicated in Figure 4-6, there is a larger deviation for small particles. On the other hand, by using a smaller coarse graining ratio (i.e., CG = 5 instead of CG = 15), smaller deviations are obtained for all particle classes (cp. Table 4-5). Though, using a coarse graining ratio of 15 yields faster simulations, since a much smaller amount of particles is tracked. Particle trajectories from simulations with coarse graining ratios of five and ten are in excellent agreement (i.e., less than 1.8% deviation of the channel width).

### 4.3.2. Grid Resolution

Another important numerical parameter is the grid resolution. Four simulations with different mesh sizes have been performed to analyze the influence of this parameter on the simulation results. The particle trajectories obtained from these simulations are illustrated in Figure 4-7.

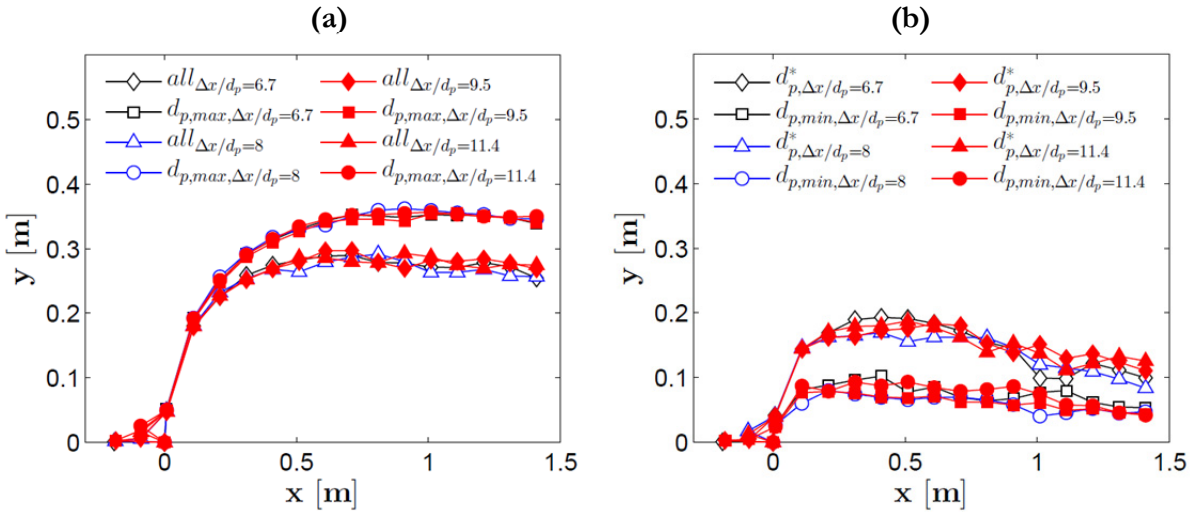


**Figure 4-7: Mass averaged particle trajectories for different particle classes and grid resolutions (particle injection velocity = 15 [m/s]; (a) all particles and particles with $d_{p,max}$ = 50 [µm], (b) particles with $d_p^*$ = 20 [µm] and $d_{p,min}$ = 6 [µm]; (for all other parameters see the base case definition in Section 4.1).**

As can be seen, the effect of the grid resolution is very small. This is especially true for the largest particles, where the particle trajectories are almost identical. The particle trajectories for other particle classes differ from the reference results (i.e., $\Delta x/d_p = 8$) only insignificantly (see Table 4-6 for a summary).

**Table 4-6: Relative mean deviation (in per cent of the channel width) to the particle trajectories for the case with a grid resolution of $\Delta x/d_p = 8$.**

| $d_{prim}$ [µm] | RMD [%] $\Delta x/d_p = 6.7$ | RMD [%] $\Delta x/d_p = 9.5$ | RMD [%] $\Delta x/d_p = 11.4$ |
|---|---|---|---|
| 50 | 0.64 | 0.80 | 0.53 |
| 20 | 2.07 | 2.40 | 2.36 |
| 6 | 2.24 | 0.79 | 2.19 |

Table 4-6 above reveals that there is hardly any deviation of the particle trajectories for different grid resolutions. Hence, changing the grid resolution between $\Delta x/d_p = 11.4$ (no. of cells = $310 \cdot 10^3$) and $\Delta x/d_p = 6.7$ (no. of cells = $890 \cdot 10^3$) yields similar results. Most significantly, the trajectory of comparably large particles (i.e. $d_{prim}$ = 50 [µm]) is not influenced at all by altering the mesh size. This is due to larger Stokes number of these particles, i.e., fluid velocity fluctuations (resolved on finer grids) do not affect the trajectories of the particles appreciably.

Additionally, the flow fields for the different grid resolutions are exhibiting the expected behavior since no distinct influence of the grid size can be noticed (see Figure 4-8). Thus, the mesh size of the base case (i.e., $\Delta x/d_p = 8$) is suitable for the cross-flow simulations. Consequently, it is used for further studies of the cross-flow (which are described in the following section) as a coarser computational mesh did not indicate a noticeable benefit in terms of computing time.

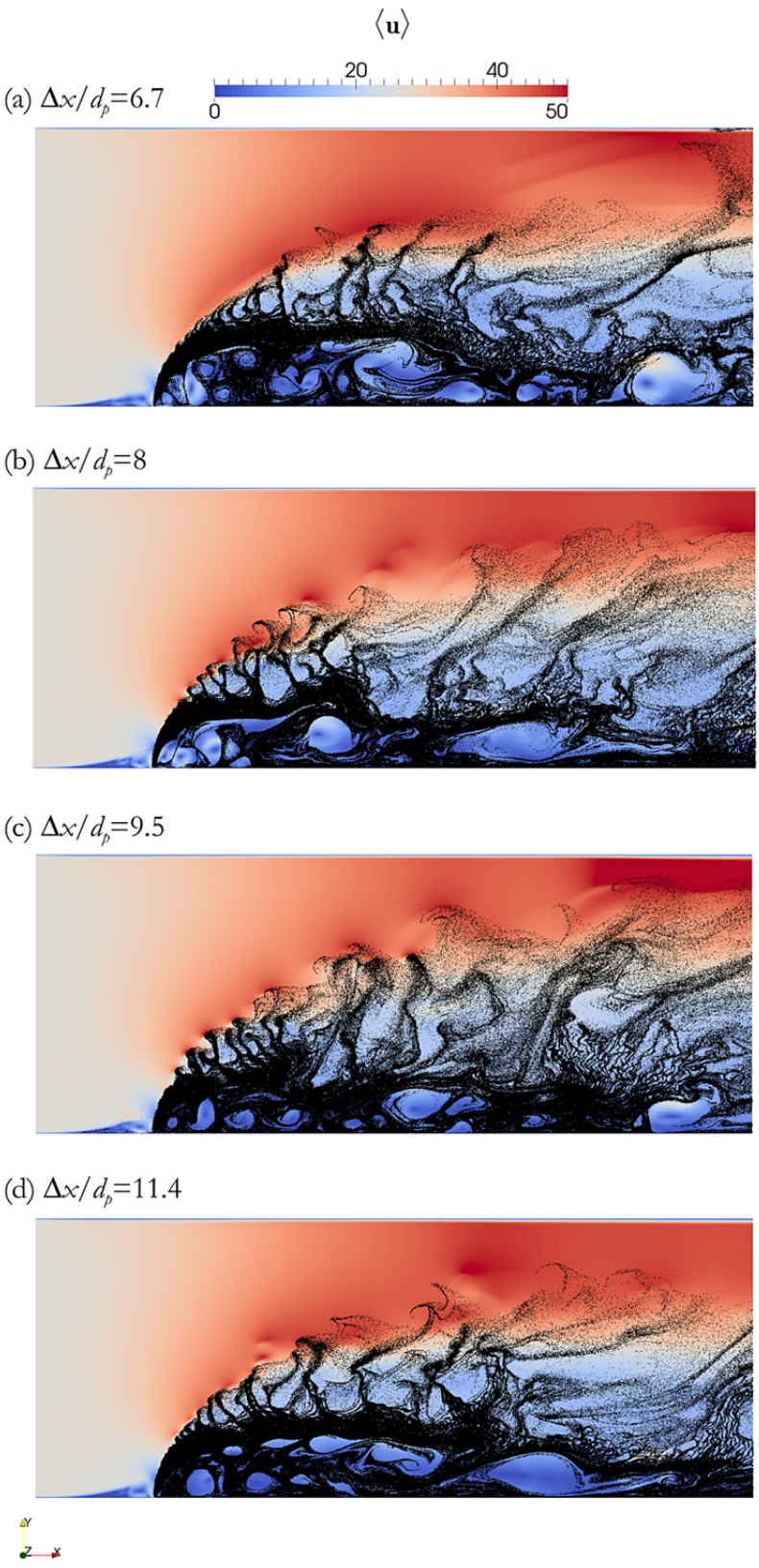


**Figure 4-8: Time averaged air velocity fields for different grid resolutions and a particle injection velocity of 15 [m/s]; (a) $\Delta x/d_p = 6.7$, (b) $\Delta x/d_p = 8$, (c) $\Delta x/d_p = 9.5$, (d) $\Delta x/d_p = 11.4$ (for all other parameters see the base case definition in Section 4.1).**

### 4.3.3. Void Fraction Model

The varying number and the overlapping of particles with more than one computational cell require an adequate model for the calculation of the volume fraction occupied by the fluid (i.e. void fraction). Two different models for calculating the void fraction in a computational cell have been compared in this work. The "divided" void fraction model divides the particle volume among the cells according to the volume the particles are occupying in each of those cells. To quantify the fraction of the particle volume in the affected cells, satellite points distributed over the volume of a particle are used. This procedure is similar to the mapping algorithm reported by Radl et al.[2]. A schematic illustration of the divided void fraction model is given in Figure 4-9, where 17 satellite points are displayed in a two-dimensional setup. In the (three-dimensional) simulations, a total number of 29 satellite points was used for each particle.

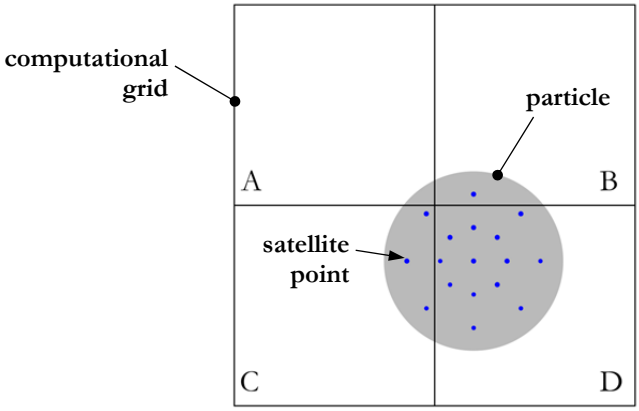


**Figure 4-9: Schematic illustration of the divided void fraction model (blue dots are representing satellite points).**

The sum of satellite points (of a single particle) in a cell divided by the total number of satellite points yields a weighting coefficient that is used for dividing the particle volume among surrounding cells. Specifically, the particle shown in Figure 4-9 would not contribute volume to cell A, as this cell does not contain any satellite point. In contrast, one satellite point can be found in cell B which means that $1/29^{th}$ of the particle volume is assigned to this cell. The same procedure yields the particle volume assigned to the other cells.

The second void fraction model used in the simulations is the "weighted neighbor" model. Here, additionally to the divided model a further weighting coefficient is included in the calculation of the occupied volume. This coefficient is the reciprocal of the distance from a particular satellite point to the cell center. In other words, the closer a satellite point is to the cell center, the more particle volume (i.e. less void fraction) is assigned to this cell.

Snapshots of the obtained void fraction fields for the two different void fraction models are illustrated in Figure 4-10.

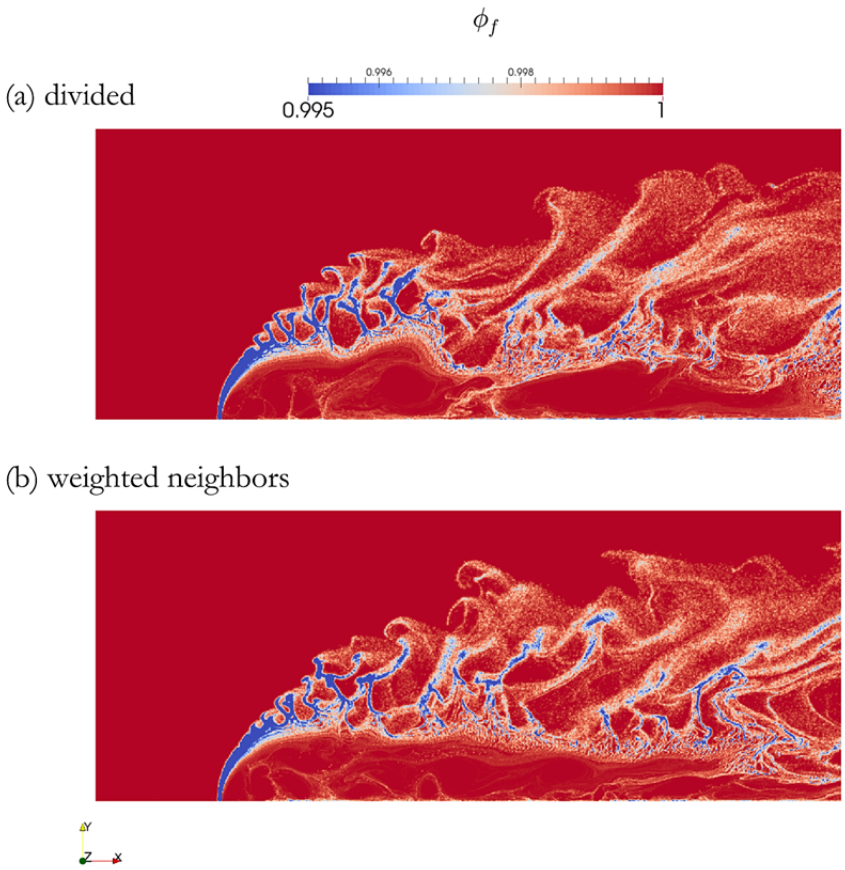


**Figure 4-10: Fluid volume fraction $\phi_f$ for different void fraction models; (a) divided, (b) weighted neighbors (CG = 5, particle injection velocity = 15 [m/s]; for all other parameters see the base case definition in Section 4.1).**

Evidently, the deviation of the void fraction fields for the tested models is limited, and both models yield almost identical results. Also, the influence on the particle trajectories is insignificant (see Figure 4-11). In summary, the comparison of the two void fraction models does not reveal significant differences, and hence we have used the "divided" void fraction model for the remaining part of this study.

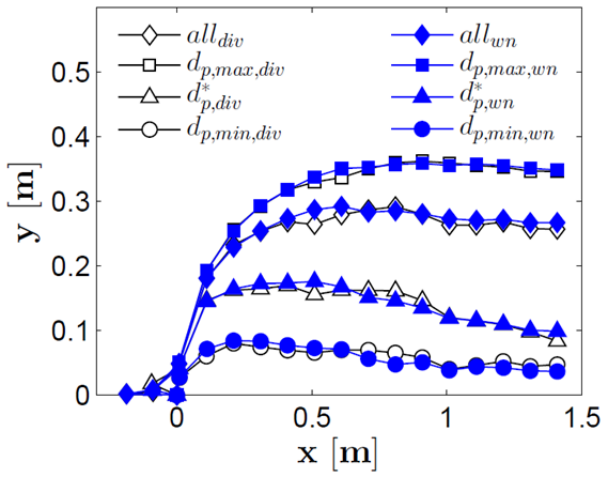


**Figure 4-11: Mass averaged particle trajectories of different void fraction models (CG = 5, particle injection velocity = 15 [m/s]; open symbols refer to the divided void fraction model, and closed symbols to the weighted neighbor model; for all other parameters see the base case definition in Section 4.1).**

## 4.4. Dispersion Pattern

As already briefly illustrated in the last chapter, large particles will have a higher penetration depth compared to small ones when injecting a polydisperse particle ensemble in a cross-flow. In Figure 4-12 and Figure 4-13 we illustrate this behavior in terms of the spatial particle distribution, as well as with the mean trajectory of certain particle classes.

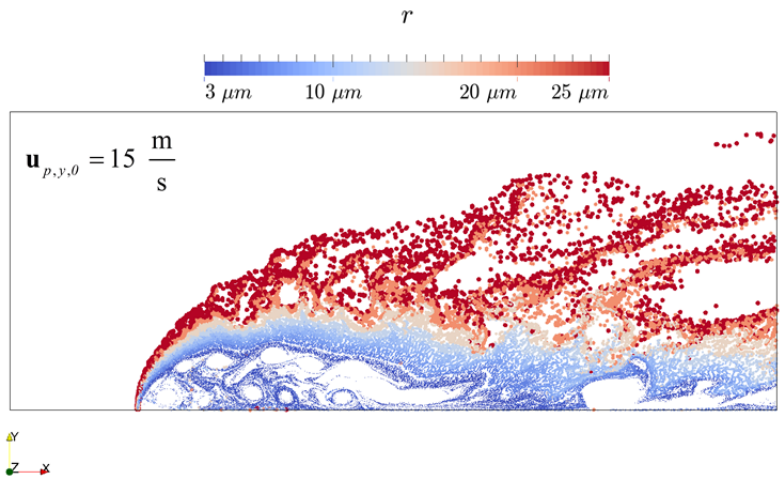


**Figure 4-12: Particle distribution in the channel (CG = 10, particle injection velocity = 15 [m/s]; particles are colored according their size; for all other parameters see the base case definition in Section 4.1).**

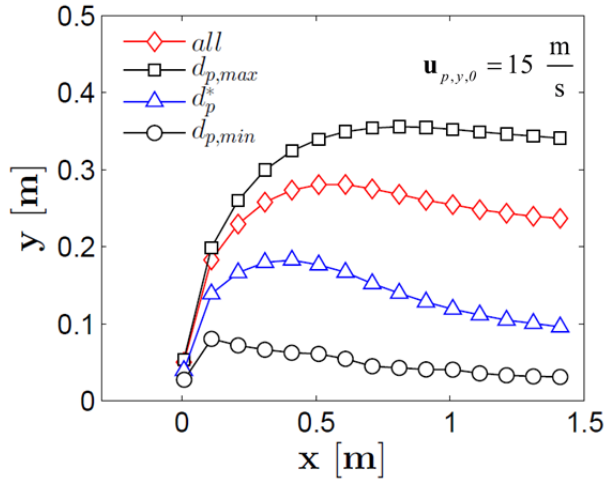


**Figure 4-13: Mass averaged particle trajectories of different particle classes (CG = 10, particle injection velocity = 15 [m/s]; for all other parameters see the base case definition in Section 4.1).**

The different particle masses are the reason for different average penetration depth of each particle class. Clearly, as large particles have a higher mass, they are able to penetrate deeper because of their larger Stokes number (see Table 4-7). In case the Stokes number is small, the particles adapt quickly to changes in the surrounding fluid flow field. Hence, they are immediately redirected by the cross-flow and are moving at a velocity similar to the fluids'

velocity. Obviously, these particles are not able to penetrate as deep as particles with larger Stokes numbers. Also, these small particles exhibit a more complex trajectory, as they initially penetrate deep into the flow, but subsequently move toward the bottom wall. This effect is a result of the recirculation bubble that forms downstream of the injection point.

**Table 4-7: Particle diameters and corresponding Stokes numbers.**

| ***Particle diameter* [µm]** | ***Stk*** |
|---|---|
| $d_{p,max} = 50$ | 92.4 |
| $d_p^* = 20$ | 14.8 |
| $d_{p,min} = 6$ | 1.3 |

## *4.5. Sensitivity to the Particle Injection Velocity*

The importance of the particle injection velocity on the particle dispersion becomes obvious when analyzing the particle distribution in the channel at a lower particle injection velocity (see Figure 4-14).

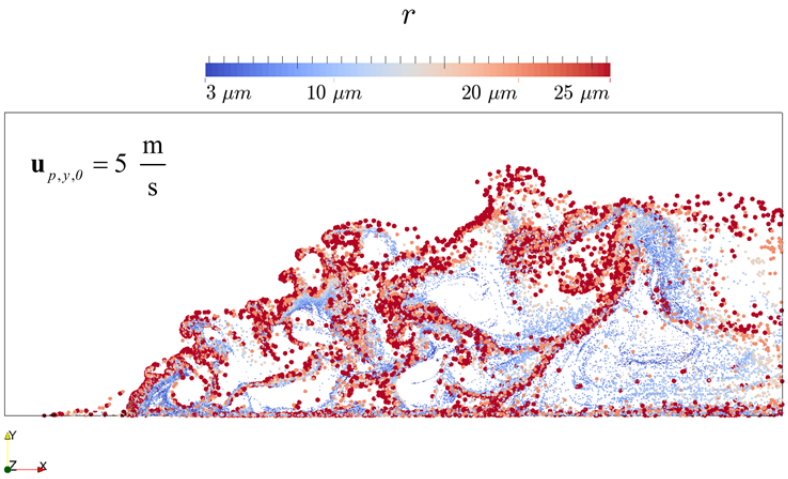


**Figure 4-14: Particle distribution in the channel (CG = 5, particle injection velocity = 5 [m/s]; particles are colored according their size; for all other parameters see the base case definition in Section 4.1).**

Here, no distinct particle trajectory can be recognized as the particles are well mixed over large parts of the simulation domain. Also the separation of the particles is worse compared to the cases in which the particles are injected with a higher velocity (cp. Figure 4-12 and Figure 4-14). Hence, this setup does not yield a satisfying separation of particles in the channel, and might negatively influence the separation in the classifier.

The sensitivity of the particle dispersion and penetration depth to the injection velocity has been studied by performing simulations with different particle injection velocities. Clearly, by injecting particles at a higher velocity they are supposed to penetrate deeper because of the larger Stokes number (see Equation (4-5)). The particle trajectories obtained from the simulations are illustrated in Figure 4-15.

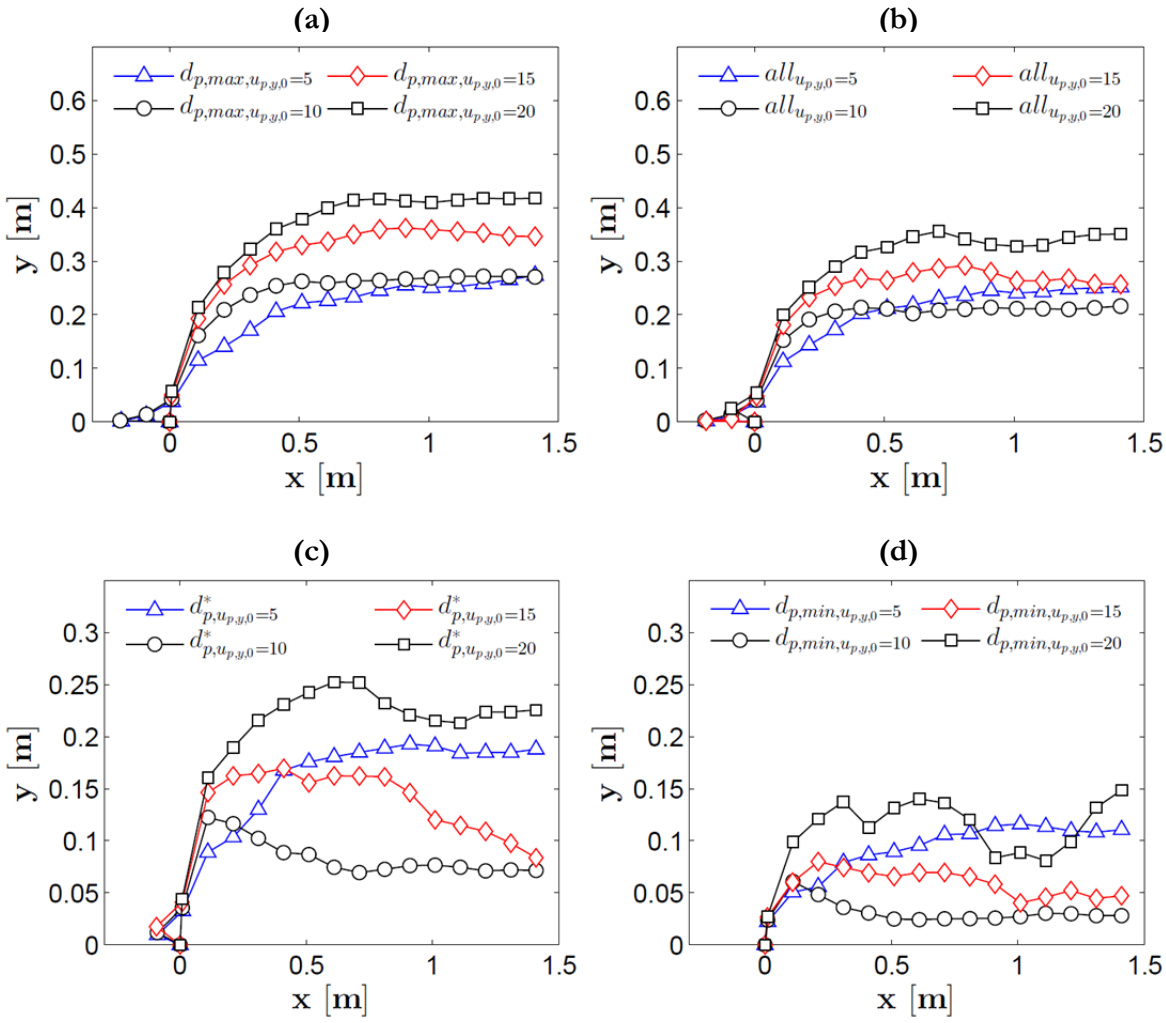


**Figure 4-15: Mass averaged particle trajectories for different particle injection velocities, (a) particles with $d_{p,max}$ = 50 [µm], (b) all particles, (c) particles with $d_p^*$ = 20 [µm] and (d) $d_{p,min}$ = 6 [µm]; (for all other parameters see the base case definition in Section 4.1).**

As can be seen, comparably large particles (i.e., $d_{p,max}$ = 50 [µm]) are penetrating deeper at higher injection velocities (see Figure 4-15a). In contrast, due to the insufficient separation of the particles when injecting them at a lower velocity (i.e., 5 [m/s]), the small particles are entrained and, surprisingly, are penetrating deeper for low injection velocities compared to larger ones (see Figure 4-15c and d, as well as Table 4-8). This leads to an increased penetration depth of the particle assembly for the lowest injection velocity when compared to much larger injection velocities (see the data for $x > 0.5$ [m] in Figure 4-15b).

In addition to the influence of the particle injection velocity on the penetration depth, the dispersion of the particles across the channel height is also affected. The standard deviation of the mean particle position characterizes the spreading and dispersion of the particles (see Table 4-8).

**Table 4-8: Time-averaged penetration depth and corresponding standard deviation for different particle classes and particle injection velocities in a region between 0.7 [m] < x < 0.8 [m] downstream of the injection point; (for all other parameters see the base case definition in Section 4.1).**

| | $d_{p,min}$ = 6 [µm] | | $d_p^*$ = 20 [µm] | | $d_{p,max}$ = 50 [µm] | |
|---|---|---|---|---|---|---|
| $u_{p,y,0}$ [m/s] | $\langle y \rangle$ [m] | $\sigma_y$ [m] | $\langle y \rangle$ [m] | $\sigma_y$ [m] | $\langle y \rangle$ [m] | $\sigma_y$ [m] |
| **5** | 0.111 | 0.024 | 0.191 | 0.137 | 0.239 | 0.147 |
| **10** | 0.028 | 0.035 | 0.072 | 0.149 | 0.265 | 0.146 |
| **15** | 0.072 | 0.070 | 0.170 | 0.126 | 0.357 | 0.120 |
| **20** | 0.118 | 0.071 | 0.247 | 0.110 | 0.416 | 0.082 |

Clearly, the particle dispersion at all investigated injection velocities is the lowest for the smallest particle class considered. These particles remain close to the bottom wall and are insufficiently dispersed over the channel width. By reducing the particle injection velocity (i.e., 5 [m/s]), the spreading of the smallest particle class is lowest. In contrast, for sufficiently large particles (i.e., $d_{p,max}$ = 50 [µm]) the spreading of particles in the channel becomes smaller when increasing the injection velocity. At the same time, large particles penetrate deeper when injecting the feed at a higher velocity.

## 4.6. Sensitivity to the Particle Injection Angle

Another way to affect the penetration depth is to change the angle of injection $\alpha_p$. Therefore, six different injection configurations have been investigated, in which an injection velocity of 15 [m/s] has been used. By decreasing the injection angle (i.e., injecting the particles against the cross-flow direction), a particle distribution as shown in Figure 4-16 is obtained.

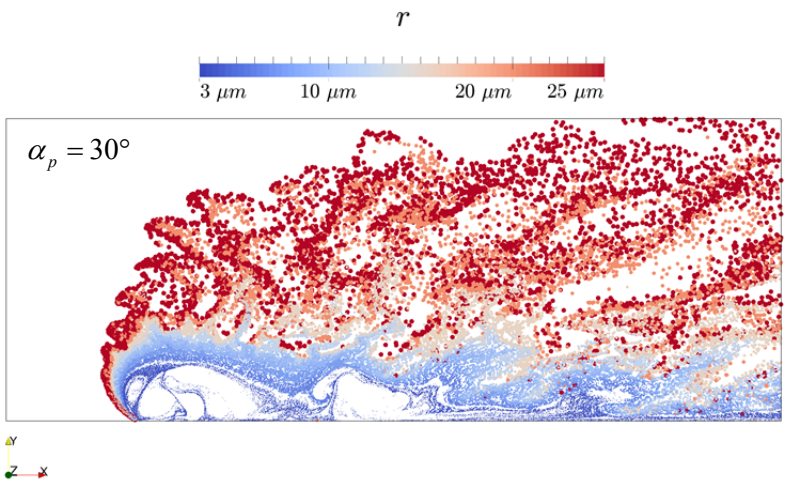


**Figure 4-16: Particle distribution in the channel ($\alpha_p$ = 30°, CG = 10, particle injection velocity = 15 [m/s]; particles are colored according their size; for all other parameters see the base case definition in Section 4.1 ).**

Apparently, using a smaller injection angle leads to a distribution of particles over the whole channel width, whereas in the case of perpendicular injection (cp. Figure 4-12) the particles are not able to advance to the upper regions of the cannel. Also, the particles respond quickly to the gas velocity as they are only slightly penetrating further upstream after being injected. This is due to the particles' low Stokes numbers (cp. Table 4-7).

The angle of injection has not only a strong influence on the penetration depth, but also the particle dispersion across the channel height shows a significant dependency on this parameter (cp. Table 4-9).

**Table 4-9: Time-averaged penetration depth and corresponding standard deviation for different particle classes and particle injection angles in a region between 0.7 [m] < $x$ < 0.8 [m] downstream of the injection point; (CG = 10, particle injection velocity = 15 [m/s]; for all other parameters see the base case definition in Section 4.1).**

| | $d_{p,min}$ = 6 [µm] | | $d_p^*$ = 20 [µm] | | $d_{p,max}$ = 50 [µm] | |
|---|---|---|---|---|---|---|
| $\alpha_p$ [°] | $\langle y \rangle$ [m] | $\sigma_y$ [m] | $\langle y \rangle$ [m] | $\sigma_y$ [m] | $\langle y \rangle$ [m] | $\sigma_y$ [m] |
| **30** | 0.020 | 0.045 | 0.112 | 0.197 | 0.430 | 0.175 |
| **45** | 0.018 | 0.043 | 0.111 | 0.182 | 0.414 | 0.174 |
| **60** | 0.017 | 0.039 | 0.110 | 0.162 | 0.387 | 0.163 |
| **75** | 0.028 | 0.034 | 0.107 | 0.151 | 0.362 | 0.150 |
| **90** | 0.048 | 0.054 | 0.149 | 0.122 | 0.355 | 0.114 |
| **105** | 0.029 | 0.032 | 0.106 | 0.116 | 0.296 | 0.104 |

An increased dispersion is observed for large particles (i.e., $d_p \geq 20$ µm) when decreasing the injection angle. That is, the particles are spread over the channel cross section more completely. However, this effect is not observed for the smallest particles, as the standard deviation of the particle position is approximately constant when changing the injection angle.

A comparison of the influence of the injection angle on the penetration depth is shown in Figure 4-17.

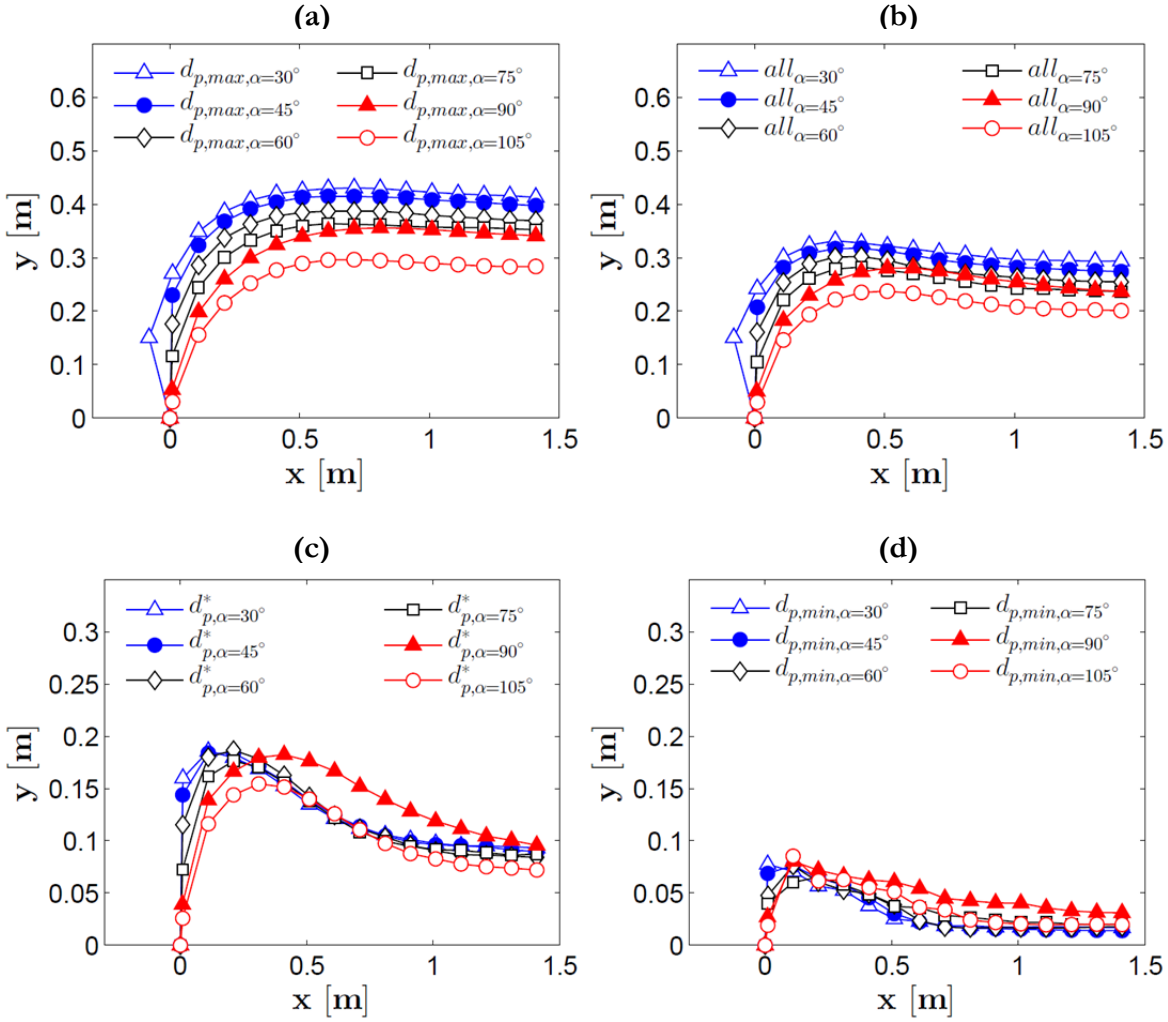


**Figure 4-17: Mass averaged particle trajectories for different particle injection angles, particle injection velocity = 15 [m/s], CG = 10, (a) particles with $d_{p,max}$ = 50 [µm], (b) all particles, (c) particles with $d_p^*$ = 20 [µm] and (d) $d_{p,min}$ = 6 [µm]; (for all other parameters see the base case definition in Section 4.1).**

The mass averaged particle trajectories reveal that the largest particles used in the simulations attain a higher penetration depth when decreasing the injection angle (cp. panel (a) in Figure 4-17). In contrast, particles with a diameter of $d_p^*$ = 20 [µm] show a different behavior (cp. panel (c) in Figure 4-17). After a certain distance (i.e., 0.5 [m]) downstream of the injection point, particles that are injected perpendicular to the cross-flow are penetrating

deepest. However, the particle penetration distance in a region ca. 0.4 [m] to 0.8 [m] downstream of the injection point is very similar for all other injection angles. At the end of the channel, the trajectory for the perpendicular injection is again getting closer to the trajectories of the other cases. Thus, the largest effect of the injection angle on the penetration depth for this particle class is only visible in the region described before. For the smallest particles, the injection angle has limited influence (see Figure 4-17, panel (d)). Nevertheless, perpendicular injection seems to yield the largest penetration depth for this particle class.

## 4.7. Cross-Flow Particle Injection with Additional Air Nozzles

The cross-flow simulations have already indicated that the particle injection velocity has a strong influence on the penetration depth and the particle dispersion over the channel height. However, establishing the required particle injection velocity in order to obtain a satisfactory particle penetration may be connected to practical difficulties. One alternative to the direction acceleration of particles is the utilization of air nozzles to accelerate the particles indirectly. Consequently, we have studied the effect of different nozzle configurations on the particle distribution in the channel.

For calculating the required air velocity at the nozzles and the required nozzle diameter, a simple momentum balance can be used:

$$\dot{m}_{nozzle} \cdot \mathbf{u}_{nozzle} + \dot{m}_p \cdot \mathbf{u}_{p,0} = \left( \dot{m}_{nozzle} + \dot{m}_p \right) \cdot \mathbf{u}_{mix}, \tag{4-8}$$

where $\dot{m}_{nozzle}$, $\dot{m}_p$, $\mathbf{u}_{nozzle}$ and $\mathbf{u}_{mix}$ are the mass rate of air injected through the nozzle(s), the mass rate of particles, the air injection velocity at each nozzle and the velocity of the air-particle mixture, respectively. Assuming that the particle injection velocity is negligible ( $\mathbf{u}_{p,0} \approx 0$ ) and that $\dot{m}_{nozzle}$ equals 10% of the main air inflow rate, the nozzle diameter as well as the nozzle velocity for a certain number of nozzles $n$ can be calculated. In the case where the mass loading approaches unity ($\mu_{\text{inlet}}$ = 1.0) we can estimate the nozzle velocity to be $\mathbf{u}_{nozzle} \approx 11 \cdot \mathbf{u}_{mix}$. With the volumetric nozzle air flow rate $Q_{nozzle}$, the nozzle diameter $d_{nozzle}$ is given by:

$$d_{nozzle} = \sqrt{\frac{4}{\pi} \frac{Q_{nozzle}}{\left|\mathbf{u}_{nozzle}\right| \cdot n}}. \tag{4-9}$$

For the simulation of the different nozzle configurations, the volumetric air flow rate through the nozzles was kept constant. Thus, the particular nozzle cross-section (diameter) had to be

adapted for different numbers of nozzles in order to retain the same nozzle velocity. The nozzles are located at the bottom wall ($y = 0$), starting at the point of particle injection ($x = 0$). Nozzles are separated by a horizontal distance of 0.1 [m] (e.g., when using eight nozzles, the last nozzle is at $x = 0.7$ [m]).

Some results for the particle distribution in the channel obtained from simulations with different nozzle configurations are shown in Figure 4-18. These results reveal that the particle distribution is only influenced in regions close to the nozzles (cp. Figure 4-18 with Figure 4-14 in which no nozzle is used). Clearly, the particle cloud can be lifted off from the bottom wall due to the presence of air nozzles. However, further downstream of the nozzles a layer of particles is covering the bottom wall again. When using only one nozzle the total amount of air is injected close to the particle feeding position. This yields obviously the best momentum transport (from the nozzle) to the particle phase. Consequently, it leads to the dispersion of small particles over the channel height (see panel (c) in Figure 4-18). Additionally, due to the air nozzle the size of the recirculation bubble behind the particle injection can be observed, which leads to partial redispersion of small particles. The particle distribution in the channel for the case with a higher nozzle velocity is similar to the result for a lower injection angle $\alpha_n$ (cp. Figure 4-18, panel (d) and (e)). In both cases, excellent particle dispersion over the channel width can be obtained. In order to quantify the effect of different nozzle configurations, the mean penetration depth for different particle classes and the corresponding standard deviation from the particle position is summarized in Table 4-10.

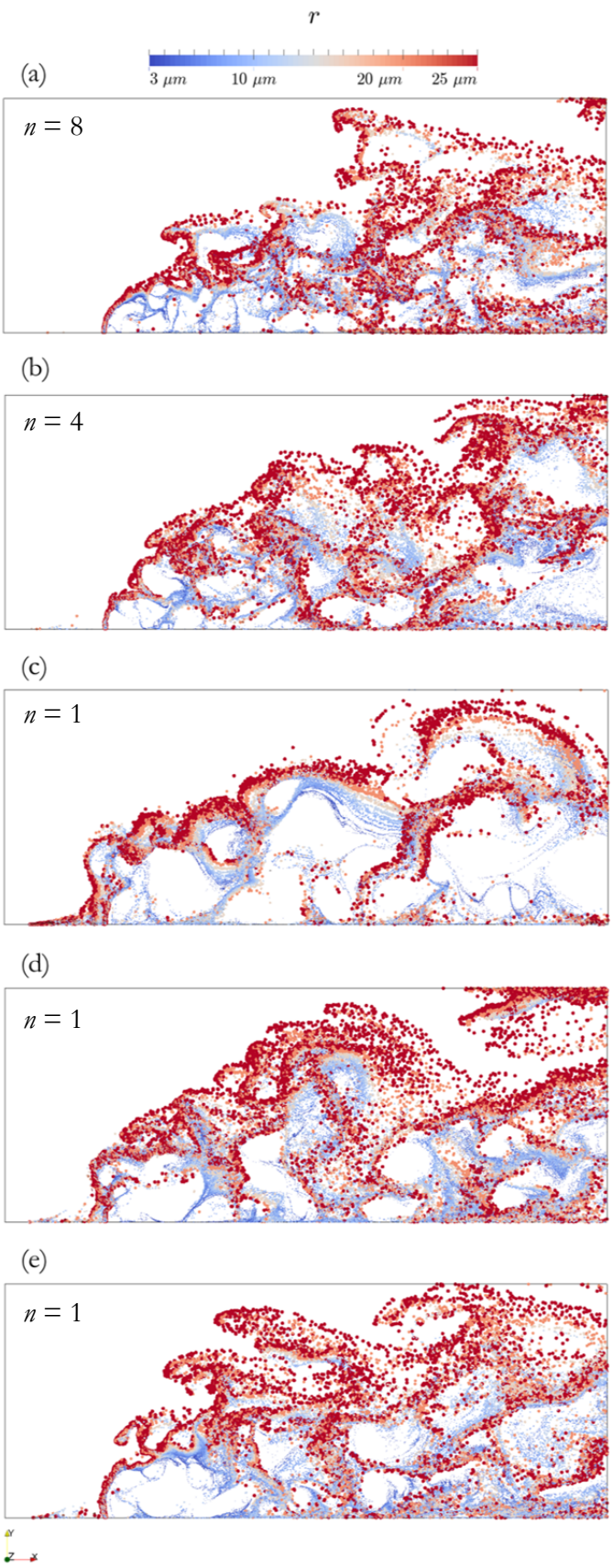


**Figure 4-18: Particle distribution in the channel (CG = 10, particle injection velocity = 5 [m/s], $u_{mix}$ = 10 [m/s]; (a) eight nozzles, (b) four nozzles, (c) one nozzle, (d) one nozzle: $u_{mix}$ = 15 [m/s], (e) one nozzle: $\alpha_n$ = 30°; particles are colored according their size; for all other parameters see the base case definition in Section 4.1).**

**Table 4-10: Time-averaged penetration depth and corresponding standard deviation for different particle classes and nozzle configurations in a region between 0.7 [m] < $x$ < 0.8 [m] downstream of the injection point; (CG = 10, particle injection velocity = 5 [m/s]; for all other parameters see the base case definition in Section 4.1).**

| *Case* | | | $d_{p,min}$ = 6 [µm] | | $d_p^*$ = 20 [µm] | | $d_{p,max}$ = 50 [µm] | |
|---|---|---|---|---|---|---|---|---|
| $n$ [-] | $u_{mix}$ [m/s] | $\alpha_n$ [°] | $\langle y \rangle$ [m] | $\sigma_y$ [m] | $\langle y \rangle$ [m] | $\sigma_y$ [m] | $\langle y \rangle$ [m] | $\sigma_y$ [m] |
| **8** | **10** | **90** | 0.177 | 0.021 | 0.218 | 0.129 | 0.281 | 0.142 |
| **4** | **10** | **90** | 0.145 | 0.023 | 0.181 | 0.147 | 0.267 | 0.159 |
| **2** | **10** | **90** | 0.134 | 0.030 | 0.177 | 0.140 | 0.264 | 0.140 |
| **1** | **10** | **90** | 0.173 | 0.033 | 0.201 | 0.174 | 0.345 | 0.136 |
| **1** | **15** | **90** | 0.144 | 0.031 | 0.185 | 0.167 | 0.348 | 0.174 |
| **1** | **10** | **30** | 0.101 | 0.037 | 0.165 | 0.174 | 0.317 | 0.174 |
| **1** | **15** | **30** | 0.110 | 0.033 | 0.166 | 0.165 | 0.311 | 0.191 |
| **2** | **10** | **30** | 0.098 | 0.035 | 0.175 | 0.163 | 0.306 | 0.164 |

Obviously, the deepest penetration of the largest particle class (particles with $d_{p,max}$) can be obtained when injecting the entire secondary air close to the particle feeding by using only a single nozzle. Changing the nozzle angle in a manner that the air is injected against the cross-flow direction yields an increased dispersion of large particles across the channel height. However, the mean penetration depth is not affected. Distributing the secondary air to more nozzles gives similar results for the smaller particle classes, but at the same time it leads to a decrease of the large particles' penetration. This is due to the lack of momentum transfer to large particles directly at the feed point. That is, nozzles installed further downstream of the injection point are not able to force particles closer to the channel center.

The summary of particle trajectories shown in Figure 4-19 for different nozzle configurations also supports this finding. As can be seen, all investigated nozzle configurations improve the penetration of particles into the channel compared to the case without nozzles. For comparably large particles (i.e. $d_{p,max}$ = 50 [µm]), the most significant gain in penetration depth can be observed when using only one nozzle. On the contrary, the trajectories of the smaller particles ($d_{prim} \leq 20$ [µm]) are not showing such a clear tendency (Figure 4-19, panel (c) and (d)), since using eight nozzles also yields a relatively deep particle penetration. The trajectories of these particles do not vary much when employing a different number of nozzles, though. Only in the case without nozzle utilization and a particle injection velocity of 10 [m/s], a

significantly worse penetration depth is observed. This is due to the momentum transport perpendicular to the cross-flow when employing nozzles.

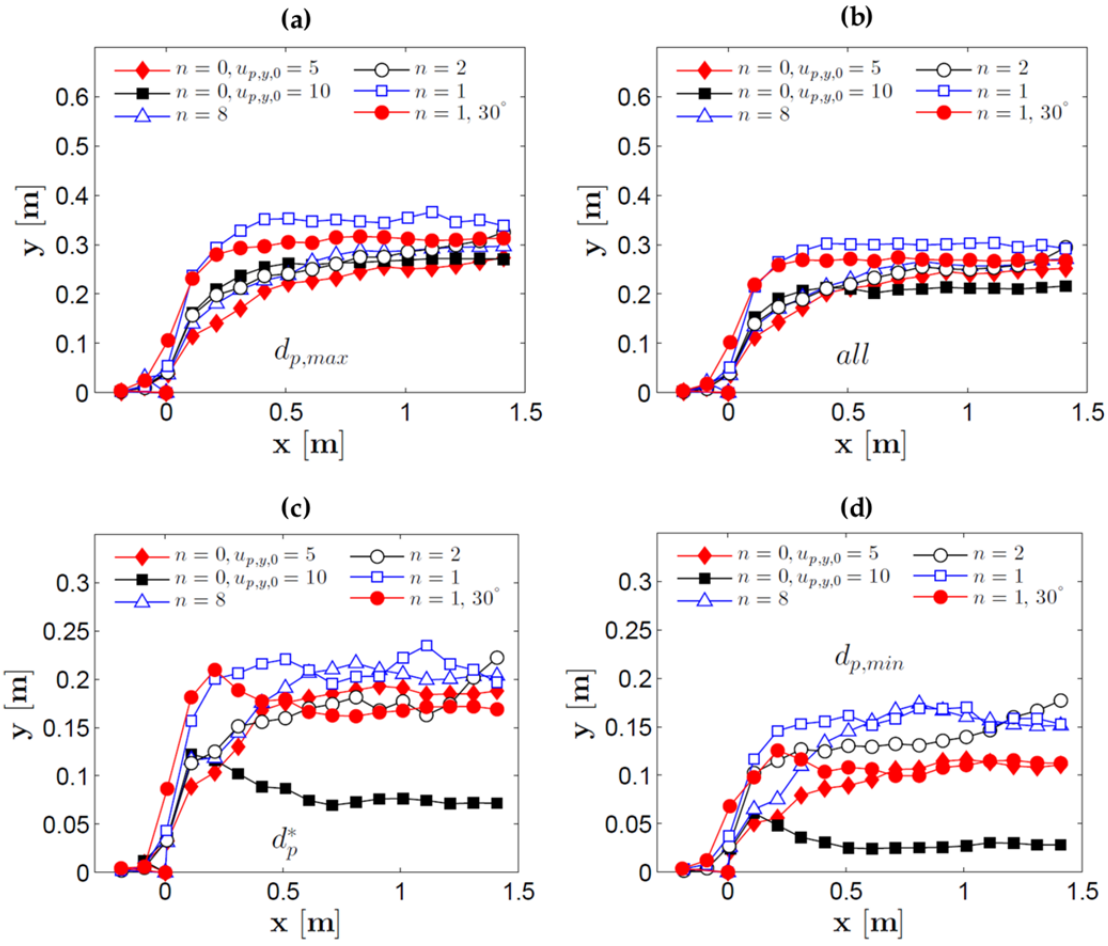


**Figure 4-19: Mass averaged particle trajectories for different nozzle configurations, particle injection velocity = 5 [m/s], $u_{mix}$ = 10 [m/s] and CG = 10 for all simulations except for the cases without nozzles (CG = 5), (a) particles with $d_{p,max}$ = 50 [µm], (b) all particles, (c) particles with $d_p^*$ = 20 [µm] and (d) $d_{p,min}$ = 6 [µm]; (for all other parameters see the base case definition in Section 4.1).**

Additional information about the obtained flow fields, particle size distributions in the channel, and void fraction fields for the different setups is provided in the Appendix (see Section 11).

## 4.8. Conclusions

Based on the simulation results and the theoretical description of particle motion in a cross flow, the following conclusions can be drawn:

- The effect of the parcel size and the grid resolution is insignificant for comparably large particles (i.e., $d_{prim}$ = 50 [µm]) at all investigated coarse graining ratios. A minimal

underprediction of the trajectory of smaller particles is observed for CG = 15 compared to finely resolved simulations using CG = 5.

- In case of a particle injection velocity of 15 [m/s], particles are classified in a satisfactory manner right after the injection point (i.e., for $x$ < ca. 0.5 [m] downstream of the injection point). Particles are then re-mixed when moving further downstream of the channel (i.e., $x$ > ca. 0.5 [m]). This knowledge could be exploited when designing the feed channel for optimal separation at the injection point.

- Simulations with smaller particle injection velocity indicate that the separation of the particles in the channel is worse compared to the case of a particle injection velocity of 15 [m/s]. Thus, the particle injection velocity should be controlled, in order to fully exploit the separation of particles during particle dispersion. Alternatively, the cross flow velocity should be reduced to realize larger jet to cross flow velocity ratios (e.g., by designing a wider channel). Also, we speculate that particle injection at the center of the channel (e.g., with a feeding pipe) could improve particle dispersion and separation, since the negative effect of the recirculation bubble would be alleviated.

- Reducing the injection angle (i.e., injecting the particles against the cross-flow direction) leads to a deeper penetration of sufficiently large particles (i.e., $d_{prim}$ > 20 [µm]), however, at the cost of an increased particle spreading.

- The penetration of comparably large particles (i.e., $d_{prim}$ = 50 [µm]) can be estimated to a first approximation based on particle trajectories in an undisturbed flow. This information is useful to estimate appropriate particle injection velocities.

- By employing air nozzles for establishing the required particle injection velocity, the penetration depth of the particles can be increased. The deepest penetration is obtained when using only one nozzle directly at the particle feed point which yields the best momentum transport to the particle phase.

- When injecting the particles with a velocity ≥ 10 [m/s] and without nozzles, the penetration of small particles (i.e., $d_{prim}$ = 6 [µm]) is significantly lower compared to the case where an air nozzle is used to accelerate the particles. This is due to an ongoing momentum transport in y-direction to the particle phase, as the secondary air injected by the nozzles influences the flow field close to the bottom wall.

- An increased penetration depth of particles (ca. 11.4% of the channel height for large particles with. $d_{prim}$ = 50 [µm]) is observed when using a single nozzle instead of injecting the particles with a velocity of 10 [m/s]. Similar results for the penetration depth of particles larger than 20 [µm] for cases with the same mixture injection velocity (i.e., without nozzle, $\mathbf{u}_{p,y,0}$ = 15 [m/s] and with one nozzle, $\mathbf{u}_{mix}$ = 15 [m/s]) can be obtained.

## 4.9. References

[1] S. Luding, Introduction to Discrete Element Methods Basics of Contact Force Models and how to perform the Micro-Macro Transition to Continuum Theory, EJECE. (2008) 785–826.

[2] S. Radl, B.C. Gonzales, C. Goniva, S. Pirker, State of the Art in Mapping Schemes for Dilute and Dense Euler-Lagrange Simulations, in: 10th Int. Conf. CFD Oil Gas, Metall. Process Ind., Trondheim, Norway, 2014: pp. 1–8.

# 5. Gas-Particle Flow in the Classifier - Annular Gap Region

## *5.1. Setup*

Due to the extremely small size of the rotor blades in the classifier, the flow between the blades cannot be modelled in these simulations. Thus, simulations of the annular gap region without a detailed model of the rotor region have been performed. Specifically, the rotor has been modeled with constant flow boundary conditions. That is, boundary conditions such as the radial velocity and the rotational speed due to the impeller wheel have been specified at the outer rotor radius. The physical parameters for the simulations of the classifier are summarized in Table 5-1, and numerical parameters are summarized in Table 5-2.

**Table 5-1: Physical parameters for the simulations of the gas-particle flow in the annular gap region.**

| ***Parameters*** | ***Value*** |
|---|---|
| Particle diameter ($d_{prim}$) | 5 … 100 [µm] |
| Particle injection velocity $\mathbf{u}_{p,y,0}$ | -5.0 [m/s] |
| Mass loading ($m_p/m_f$) | 2.0 |
| Particle density ($\rho_p$) | 2.500 [kg/m³] |
| Coefficient of restitution | 0.90 |
| Friction coefficient | 0.10 |
| Gas density ($\rho_f$) | 1.13 [kg/m³] |
| Gas dynamic viscosity ($\eta_f$) | $1.921 \cdot 10^{-5}$ [Pa·s] |
| Gravity (**g**) | (0 -9.81 0) [m/s²] |
| $Re_{inlet}$ | $9.66 \cdot 10^5$ |

In the simulations of the annular gap region, the particles are removed from the simulation domain just before they collide with the rotor, and the particle's position, velocity and size is recorded. Based on these parameters, simulations of the rotor region have been performed in which the flow between the blades is investigated separately.

**Table 5-2: Numerical parameters of the annular gap region simulation base case.**

| *Parameters* | *Value* |
|---|---|
| *Co* | < 0.6 |
| Grid size (min/max edge length) | $1.8 \cdot 10^{-5}$ [m] / 0.082 [m] |
| Number of cells | $856 \cdot 10^{3}$ |
| Discretization schemes | Euler / Gauss upwind |
| Collision tracking | Yes (DEM) |
| Coarse graining ratio (CG) | 200 |
| $\Delta t_{DEM}/t_c$ | 1/50 |
| $\Delta t_{DEM}$ | $2 \cdot 10^{-6}$ [s] |
| LES turbulence model | Smagorinsky |

The particle size distribution of the feed is summarized in Table 5-3 and is chosen to mimic a typical feed material. There are much more (in terms of the number) small particles represented in the simulation than large particles, since as the mass fraction of small particles is comparably high and the fact that the particle volume is proportional to the particle diameter cubed. Due to numerical reasons, very large and very small particles are approximated (i.e. the largest particles of a typical feed material are modeled as particles with $d_{prim}$ = 100 [µm], and for the smallest particles a diameter of $d_{prim}$ = 5 [µm] is used).

**Table 5-3: Feed particle size distribution used for the simulations of the classifier.**

| *No.* | $d_{prim}$ [µm] | $Q_3$ | $\Delta Q_3$ |
|---|---|---|---|
| 1 | 100 | 1.00 | 0.01 |
| 2 | 100 | 0.99 | 0.03 |
| 3 | 100 | 0.96 | 0.03 |
| 4 | 85.5 | 0.93 | 0.07 |
| 5 | 58 | 0.86 | 0.10 |
| 6 | 35 | 0.76 | 0.16 |
| 7 | 17.5 | 0.60 | 0.24 |
| 8 | 5 | 0.36 | 0.16 |
| 9 | 5 | 0.20 | 0.07 |
| 10 | 5 | 0.13 | 0.13 |

An illustration of a generic air classifier, similar to the investigated device in this thesis, is given in Figure 5-1. Here, the particles are inserted at the top of the classification device. Sufficiently small particles are able to approach the inner part of the rotor. Subsequently, these particles can be discharged with the fine fraction. On the other hand, comparably large particles are moving against the air flow towards the bottom of the classifier. That is, these particles can be removed with the coarse fraction.

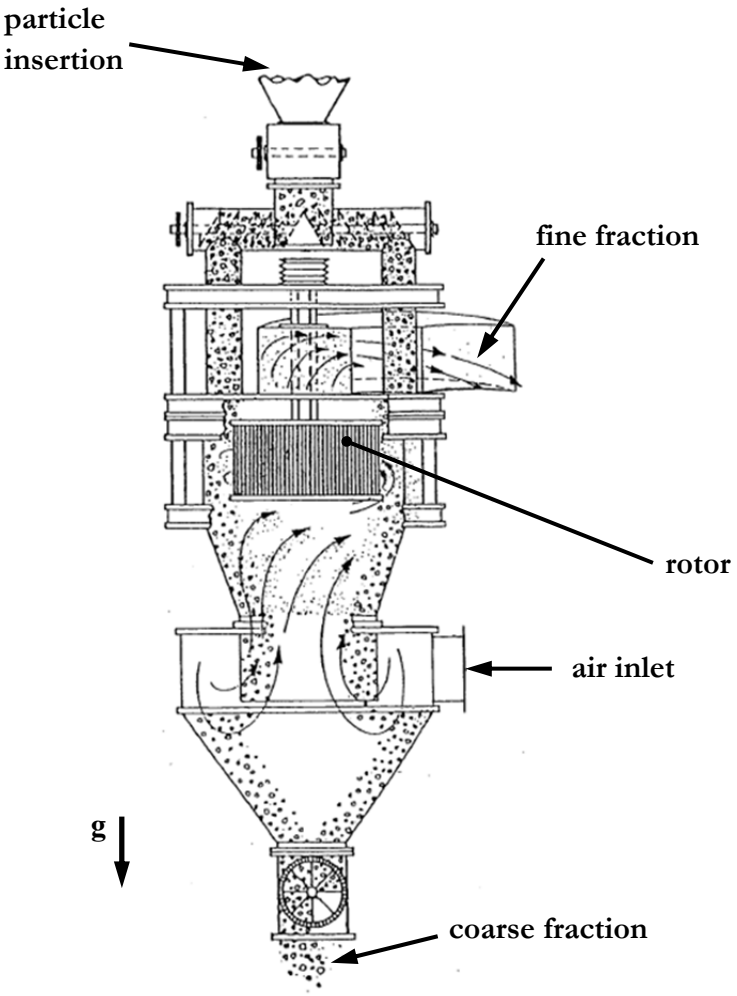


**Figure 5-1: Schematic illustration of a generic air classifier.**

## 5.2. Results

The simulation results indicate that a large fraction of particles is entering the rotor region right below the particle deflector (see Figure 5-2). This is due to the lower pressure right behind the particle deflector caused by the high rotational speed of the rotor. In the region next to the particle insertion and the particle deflector, the inflowing air is significantly decelerated by the injected particles. Due to the radial inflow, most particles are entering the rotor region merely in the top section of the rotor. Moreover, the deceleration of the flow near the particle injection leads to higher air velocities at the lower part of the classifier.

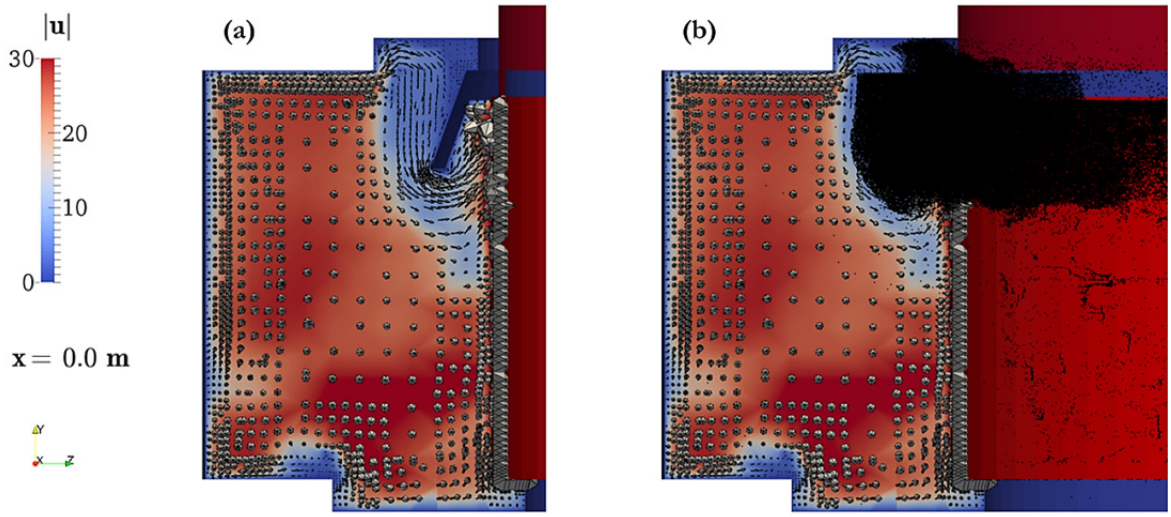


**Figure 5-2: Flow field in the cross-section perpendicular to the x-axis, rotor modeled with constant flow boundary conditions, (a) fluid flow field without particles, (b) flow field with particles (flow field is colored according to the magnitude of the fluid velocity).**

As a result of this behavior, there is little axial (i.e., *y*-) transport of the particles. Hence, the particles cannot be accelerated properly as they stay in a region with relatively slow fluid velocity close to the particle deflector and the feed plane.

Investigations of the fluid flow field in the cross-section perpendicular to the y-axis reveal that the fluid velocity in the region at the end of the helix (i.e., the spiral housing) is much lower compared to the inlet region (cp. Figure 5-3). Reasons for this are the deceleration of the gas by the particles and other losses, e.g., due to friction. However, the simulations indicate that the inflowing air, which is approaching the rotor of the classifier next to the air inlet, is partially redirected against the natural flow direction (i.e., anticlockwise in figure below) in such a device. This effect can be seen best in panel (d) of Figure 5-3 and is also accounting for the deceleration.

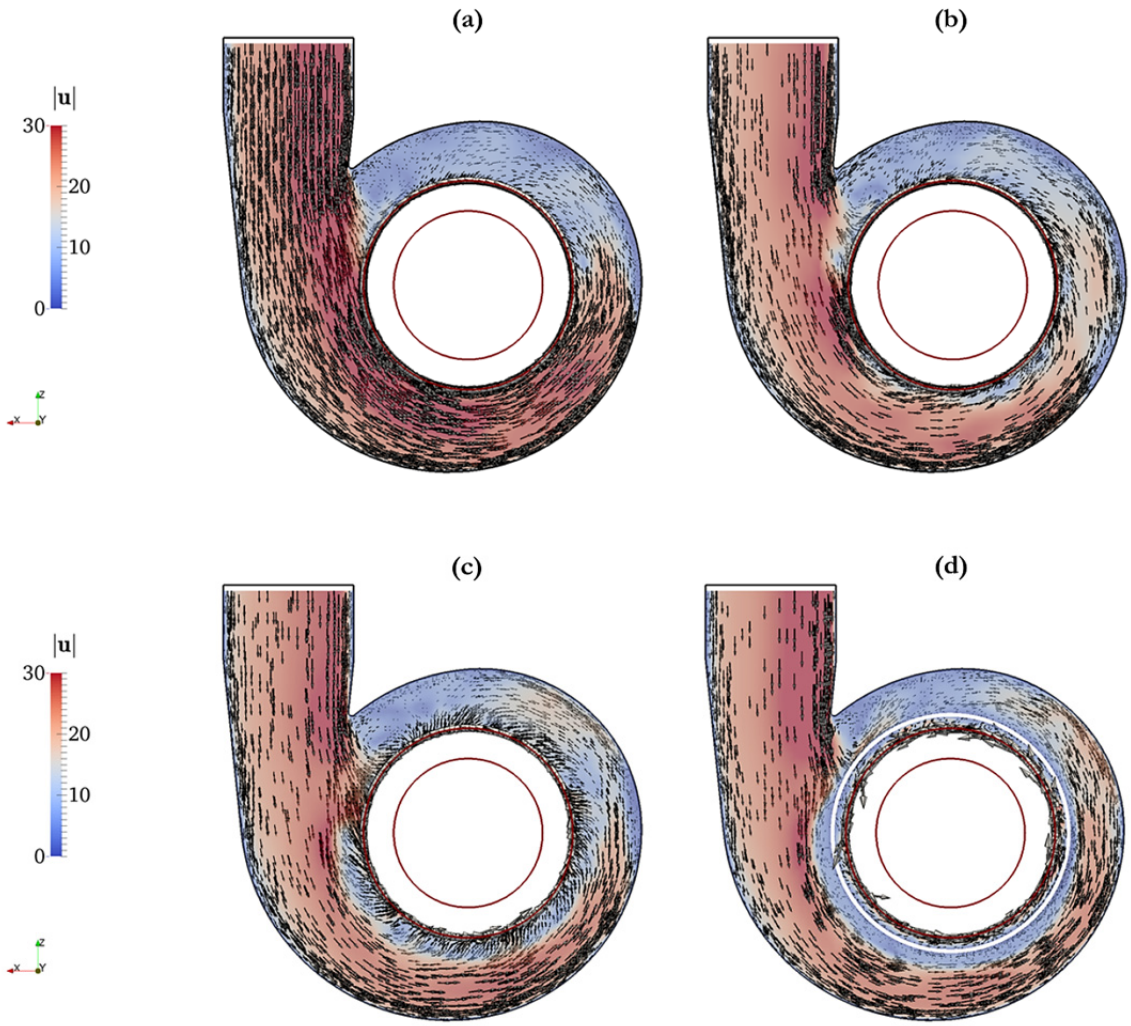


**Figure 5-3: Flow field in cross-sections perpendicular to the y-axis, (a) y = 0.10 m, (b) y = 0.30 m, (c) y = 0.40 m, (d) y = 0.50 m (flow field is colored according to the magnitude of the fluid velocity).**

Additional data of the annular gap region simulations is provided in the appendix (see Section 12), where the position and velocity statistics of particles entering the rotor region are summarized. Generally, the tangential velocity of large particles that are able to enter the rotor is varying less than the velocity of small particles. That is, the difference between the maximal and the minimal tangential velocity is smaller for larger particles. Also, large particles approach the rotor in lower sections of the classifier than small particles do. Clearly, this is due to the ability of large particles to penetrate deeper. Additionally, the deviation about the mean axial position (at which particles approach the rotor) is less for large particles compared to that of small particles. This is because the former are usually not able to approach the rotor in the top section of the classifier at the standard feeding strategy (i.e., particle insertion at the top of the classifier). However, the mean axial position of particles entering the rotor is varying substantially for different simulation setups.

# 6. Sensitivity of Particle Separation to the Rotor Design

Single-phase flow and one-way coupled gas-particle flow studies have been performed to identify the effect of various rotor blade designs on the particle separation. For these simulations a pseudo two-dimensional setup has been used which enables the resolution of the flow details between the blades. Since the annular gap region is not included in the simulations, appropriate boundary conditions for the radial velocity and the rotational speed had to be specified at the outer radius of the simulation domain. The numerical parameters used for these simulations are summarized in Table 6-1.

**Table 6-1: Numerical parameters for the simulations of the rotor region.**

| ***Parameters*** | ***Value*** |
|---|---|
| Grid size (min/max edge length) | $9 \cdot 10^{-5}$ [m] / 0.024 [m] |
| Number of cells | $> 1.5 \cdot 10^{5}$ |
| Discretization schemes | steady state / Gauss limited linear |
| RAS turbulence model | k-ω SST |

In order to avoid extremely expensive (in terms of computation time) simulations with a moving mesh, the rotor is modeled by using a rotating reference frame that is rotating with the speed of the rotor. That is, the surrounding air flow rotates relative to this reference frame, while the rotor is not moving in the simulations. Hence, the blade-relative velocity of the air is computed in these studies (cp. Figure 6-3a and b).

The simulations of the investigated rotor geometries reveal that a vortex is formed between the rotor blades. This finding is in accordance with available literature data [1,2]. Moreover, the tangential blade-relative velocity fields indicate that a free vortex is formed inside the rotor wheel. That is, the tangential velocity increases with decreasing radial distance from the rotor axis.

The vortex between two rotor blades clearly reduces the available area for radial air inflow. To assess this blockage, an open area ratio ($a_{\text{in}} = h_{in}/h_{gap}$, cp. Figure 6-1a) can be defined as the area available for the radial inflow divided by the area of the gap ($A_{gap}$) between two neighboring blades [3]. The influence of the vortex on the radial velocity profile is illustrated in Figure 6-1b. Here, a strongly non-uniform velocity profile can be observed over the (dimensionless) blade distance ($z_{gap} = z/h_{gap}$). However, the illustration of the profile is limited

to the grid resolution between the cells. Clearly, the higher the blockage of the rotor, the less area is available for the radial inflow and the larger is the mean radial inflow velocity which yields a worse separation. The value for this open area ratio has been derived from the simulated flow field and is ca. 0.51. Though, the available area for the radial inflow is varying over the rotor radius. That is, the open area ratio and the mean radial inflow velocity are not uniform, which makes the use of a simple force balance approximate.

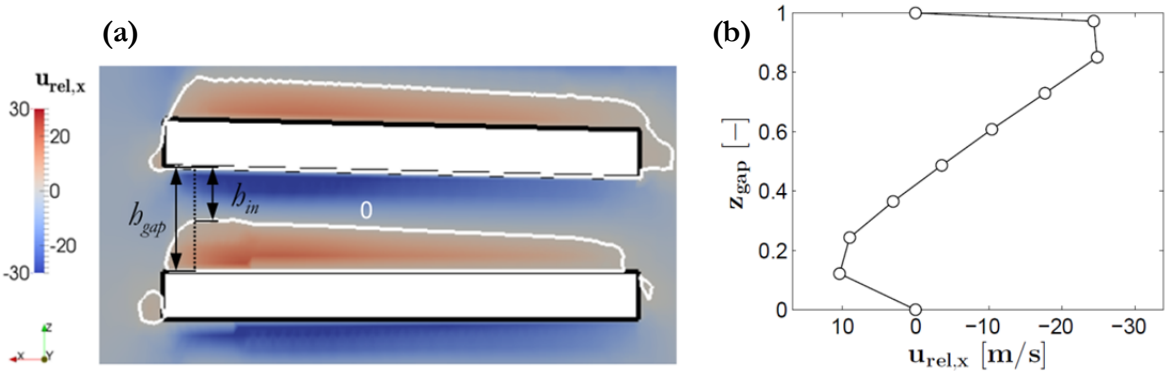


**Figure 6-1: Result of the case with straight blades; (a) radial blade-relative velocity field (white line indicates contour with no radial blade-relative velocity), (b) radial velocity profile between the blades at the mean blade radius.**

The pressure fields obtained from the simulations reveal that the lowest pressure difference over the blades ($\Delta p$) is obtained in the case with straight blades (cp. Figure 6-2a). This is due to the more uniform and generally low blade-relative tangential velocity between two neighboring blades (see Figure 6-2b). However, the results are obtained by assuming an incompressible air flow field. That is, simulations based on a compressible model of the fluid flow would be reasonable for comparing the results.

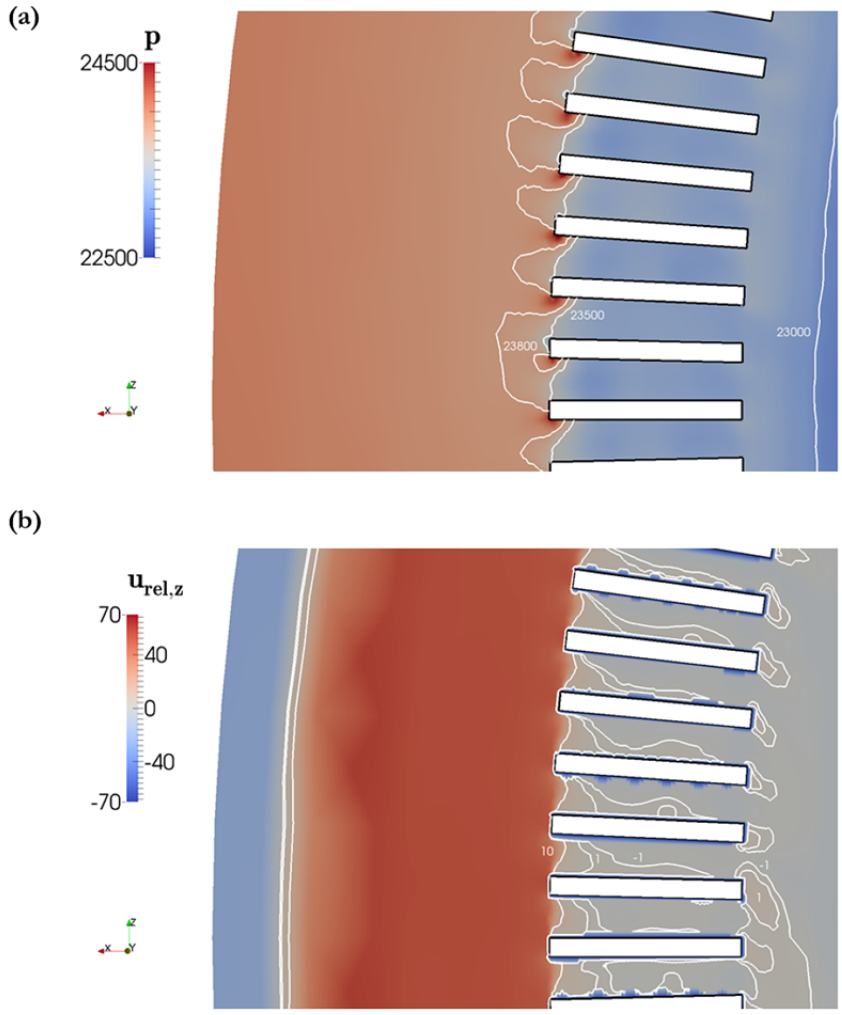


**Figure 6-2: Pressure field and flow field between the straight blades, (a) pressure field, (b) tangential blade-relative air velocity field.**

## 6.1. Particle Motion in the Rotor Region

Particle laden air flow through the rotor has been investigated by means of one-way coupled Euler-Lagrange simulations. That is, a fixed air flow field is assumed, which is not influenced by the presence of particles (i.e. no back-coupling of the particle-fluid interaction force to the air flow field has been performed). The application of one-way coupling is accurate for comparably dilute flows [4] (cp. Section 3.1.1). For these simulations, the computational particles have been of the same size as the physical particles, i.e. CG = 1. The particle size distribution of the feed was similar to that used in the cross-flow simulations (see Section 4.1).

Since the rotor region is modelled in a rotating reference frame, the Coriolis and centrifugal forces ($\mathbf{f}_C$ and $\mathbf{f}_{cf}$ respectively) exerted on the air-particle flow have to be taken into account. These additional forces exerted on particle *i* are:

$$\mathbf{f}_{C,i} = -\,2\, m_{p,i}\, \boldsymbol{\omega} \times \mathbf{u}_{p,i}, \tag{6-1}$$

$$\mathbf{f}_{cf,i} = -\, m_{p,i}\, \boldsymbol{\omega} \times \left(\boldsymbol{\omega} \times \mathbf{r}_i\right). \tag{6-2}$$

Here, $m_p$, $\boldsymbol{\omega}$, $\mathbf{u}_p$ and $\mathbf{r}$ are the particle mass, the rotation rate of the reference frame (in [rad/s]), the particle velocity (relative to the rotating reference frame), and the radial distance of the particle to the axis of rotation, respectively. Of course, these additional forces are also considered when computing the fluid flow field.

The computed particle flow pattern is illustrated in Figure 6-3. The results reveal that large particles are rejected by the rotor as they hit the blades. Smaller particles are able to follow the streamlines and hence can enter the region between the rotor blades. The vortex structure between the blades leads to a situation where small particles remain on the orbit of the formed vortex. Due to the blockage of the rotor, and the consequently increased mean radial inflow velocity, comparably large particles are able to enter the inner region of the rotor (see Figure 6-3d).

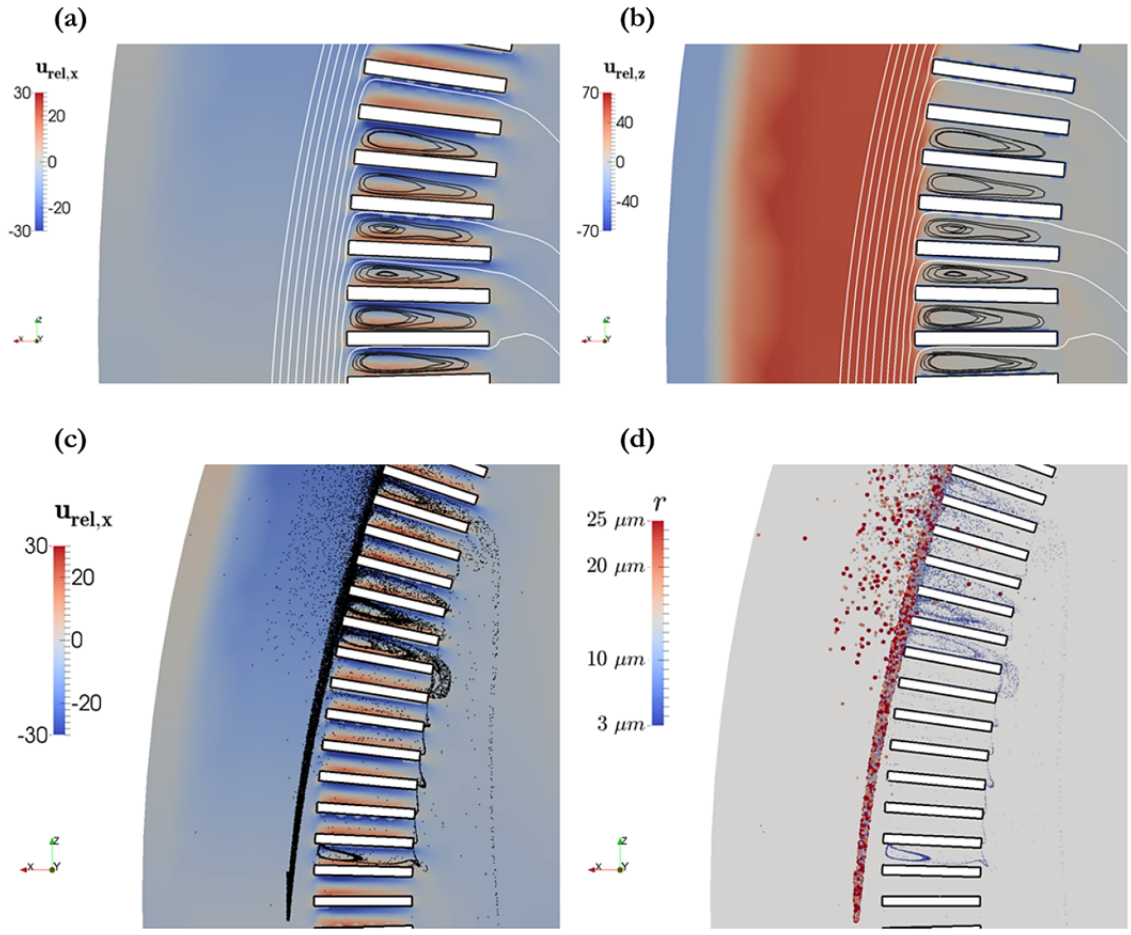


**Figure 6-3: Blade-relative air velocity fields and particle motion between the straight blades; (a) radial velocity, (b) tangential velocity, (c) radial velocity field with particles and (d) particle distribution in the rotor region.**

Moreover, the simulations indicate that at the inner side of the blades (i.e., close to the rotation axis), particles remain on a specific orbit. This is due to the formation of a free vortex in the core of the classifier's rotor, i.e., increasing tangential velocity with decreasing distance to the rotor axis. Hence, these particles do not (theoretically) enter the rotor outlet in the simulations. In industrial practice, however, particles inside the rotor will be transported axially towards the outlet, and hence will be considered as fine product. Unfortunately, it is unclear, at which radial position this axial transport becomes dominant, and removes particles from the blade region. Consequently, we had to decide, when we should remove particles from the simulation, record their properties, and count them as fine product. Specifically, we have decided to extract particles in the middle of the blades. This data set is used for determining the RRSB parameters of particles that are able to enter the inner region of the rotor ($n$ = 2.36 and $d'$ = 28.1 [µm]).

## 6.2. References

[1] P. Toneva, P. Epple, M. Breuer, W. Peukert, K.-E. Wirth, Grinding in an air classifier mill — Part I: Characterisation of the one-phase flow, Powder Technol. 211 (2011) 19–27.

[2] K. Leschonski, K. Legenhausen, Investigation of the flow field in deflector wheel classifiers, Chem. Eng. Process. 31 (1992) 131–136.

[3] S. Radl, Review and Implementation of Theoretical Models for Design and Operation of Counter Current Air Classifiers with Rotor, Graz, Austria, 2012.

[4] C.E. Brennen, Single Particle Motion, in: Fundam. Multiph. Flows, Cambridge University Press, Pasadena, California, 2005: pp. 52 – 85.

# 7. Flow through Anisotropic Porous Media

In order to mimic the flow through the rotor blades, the rotor action can be modelled by means of a rotating, anisotropic porous region (i.e., porous medium with a low resistance in radial direction and a large resistance in tangential direction).

### 7.1.1. Darcy-Forchheimer Relationship

For the modelling of the flow through this porous medium, an additional term $\mathbf{S}_i$ according to the Darcy-Forchheimer law is added to the Navier-Stokes equations. Employing Einstein summation convention, this term can be computed via:

$$\mathbf{S}_i = -\left[ \eta_f \ \mathbf{D}_{ij} + \frac{1}{2} \ \rho_f \ \left|\mathbf{u}_f\right| \ \mathbf{F}_{ij} \right] \mathbf{u}_{f,j} , \tag{7-1}$$

where $\mathbf{D}$ and $\mathbf{F}$ are symmetric tensors that represent the Darcy and Forchheimer coefficient, respectively. $\mathbf{u}_f$ is the fluid's relative velocity to the rotor blades, i.e., the fluid velocity specified in the rotating coordinate system. The tensorial notation allows one to model an anisotropic porous medium, i.e., the resistance to flow changes with the flow direction. Above equation accounts for pressure drop caused by viscous, as well as inertial effects.

### 7.1.2. Coordinate Transformation

When considering the rotor blades as an anisotropic porous media, it is useful to consider a cylindrical coordinate system to specify the Darcy and Forchheimer coefficients. By doing so, it is possible to define a different porosity in the radial, axial and tangential direction. Then, a transformation of coordinates can be used to map the coefficients to a Cartesian coordinate system. Finally, the mapped coefficients can be used when solving the Navier-Stokes equations in a Cartesian coordinate system.

In the current case, the unit vector $\mathbf{e}_{ax}$ pointing into the axis of the cylindrical porosity source is equal to the *y*-axis. The missing unity vectors in radial and tangential direction ($\mathbf{e}_r$ and $\mathbf{e}_t$, respectively) are:

$$\mathbf{e}_r = \frac{\mathbf{c} - \mathbf{s}_{ax}}{\left|\mathbf{c} - \mathbf{s}_{ax}\right|} = \frac{\mathbf{s}_r}{\left|\mathbf{s}_r\right|} , \tag{7-2}$$

$$\mathbf{e}_t = \mathbf{e}_{ax} \times \mathbf{e}_r . \tag{7-3}$$

Here, **c** is the vector pointing to the center of a computational cell in the porous region, and the vectors $\mathbf{s}_{ax}$ (calculated by: $\mathbf{s}_{ax} = (\mathbf{c} \cdot \mathbf{e}_{ax})\ \mathbf{e}_{ax}$) and $\mathbf{s}_r$ are its corresponding axial and radial components. In case we intend to model the flow through an angled porosity source (i.e., the flow field between angled blades of a rotor), a further coordinate transformation has to be undertaken:

$$\mathbf{e}_1 = -\sin(\beta_b)\ \mathbf{e}_r + \cos(\beta_b)\ \mathbf{e}_t\,, \tag{7-4}$$

$$\mathbf{e}_2 = \cos(\beta_b)\ \mathbf{e}_r + \sin(\beta_b)\ \mathbf{e}_t\,, \tag{7-5}$$

where $\beta_b$ is the angle of the rotor blades with respect to the radial direction. A schematic illustration of the different coordinate systems employed in the modeling of the rotor blades is shown in Figure 7-1.

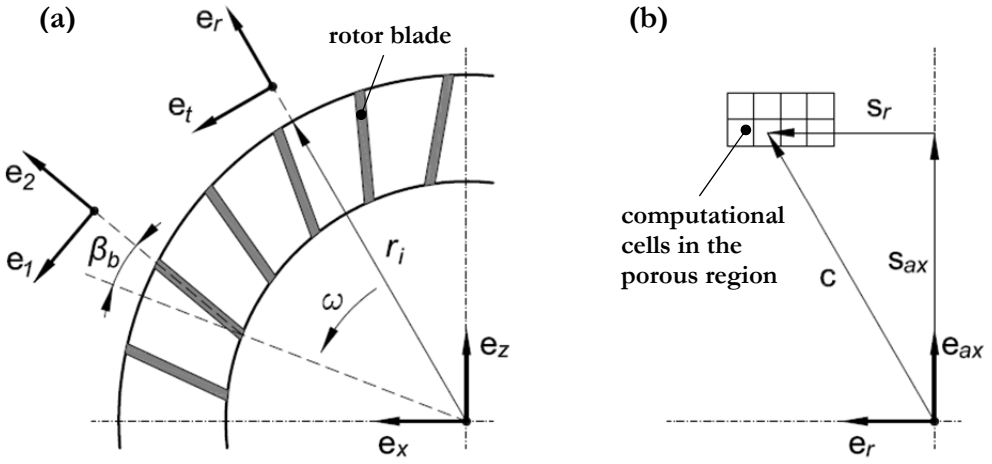


**Figure 7-1: Schematic 2D-illustration of coordinate systems for modelling flows through porous media; (a) different coordinate systems for a rotor configuration, (b) correlation between cell center vector in Cartesian coordinates and cylindrical coordinate system.**

In order to solve the Navier-Stokes equations with the included loss term in Equation (7-1), the Darcy and Forchheimer coefficients specified in the local (cylindrical) coordinate system ($\mathbf{D}_{cyl}$ and $\mathbf{F}_{cyl}$) have to be transformed back into the global (Cartesian) coordinate system (**D** and **F**). This is done by using the tensor transformation E, which can be computed from $\mathbf{e}_1$ and $\mathbf{e}_2$. For example, the Darcy coefficient can be transformed using:

$$\mathbf{D}_{ij} = \mathbf{E}_{ip}\mathbf{D}_{cyl,pq}\mathbf{E}_{jq}\,. \tag{7-6}$$

### 7.1.3. Force, Torque and Power Draw Calculation

For the calculation of the force exerted on the fluid in the porous region (more specifically, the force on each computational cell), a drag coefficient tensor $\mathbf{C}_D$ is used. This tensor equals the term in square brackets in Equation (7-1). This tensor can be used to split of an isotropic force, and compute the force on the fluid via:

$$\mathbf{f}_{fluid,i} = -C_{D,iso}\ V_{cell}\mathbf{u}_{f,i} - \left(\mathbf{C}_{D,ij} - \mathbf{I}_{ij}\ C_{D,iso}\right)V_{cell}\mathbf{u}_{f,j}\,. \tag{7-7}$$

Here, $C_{D,iso}$ is the trace (i.e., the sum of the main diagonal elements) of the drag coefficient tensor and $\mathbf{I}$ is the identity matrix. The above relationship has the advantage, that the first term on the right hand side (i.e., the isotropic contribution) can be treated in an implicit fashion when solving the Navier-Stokes equations. This improves the stability of the numerical integration algorithm.

The required force for calculating the power draw is: $\mathbf{f}_{porous} = -\mathbf{f}_{fluid}$. With this expression the torque in the porous region can be calculated:

$$\mathbf{t}_{porous} = \mathbf{f}_{porous} \times \mathbf{s}_r\,. \tag{7-8}$$

Eventually, the power draw in the porous region is obtained by:

$$P = \boldsymbol{\omega} \cdot \mathbf{t}_{porous}\,. \tag{7-9}$$

Unfortunately, due to the fact that the resolution near the porous zone is limited, the torque calculation is not accurate and strongly underpredicted. We hence have included an estimation of the power draw of the porous zone below. However, it is supposed that the torque can be calculated by means of the blade flow simulations since this simplified setup allows a higher grid resolution.

The power draw of the classifier due to the flow through the rotor can be estimated by considering the energy to accelerate the air and the particles to the circumferential speed of the rotor:

$$P_{rotor} = \dot{m}_{air}\left(1+\mu_{rotor}\right)\ \frac{u_\varphi^{\,2}}{2}\,. \tag{7-10}$$

Here, $\dot{m}_{air}$, $\mu_{rotor}$, and $u_\varphi$ are the mass rate of air, the mass loading at the rotor and the tangential velocity, respectively.

# 8. Conclusions

CFD-DEM based simulations of the particle injection in a turbulent cross-flow, as well as of various classifier setups have been performed. These numerical studies are used to identify the influence of different particle feeding strategies and modified classifier geometries on coarse particle slip and the power draw of such a device. Moreover, the effect of key numerical parameters on the simulation results has been investigated by considering a simplified cross-flow setup, which facilitates a fast screening of these parameters. The most important findings of this thesis are summarized as follows:

- The cross-flow simulations reveal that the influence of the coarse graining ratio on the penetration depth of comparably large particles (i.e., $d_{prim}$ = 50 [µm]) is insignificant. Contrary, the penetration depth of small particles is underpredicted by ca. 3.57% of the channel width when using a too large coarse graining ratio (i.e., CG = 15 instead of CG = 10). The studies of the grid resolution and the void fraction model do not reveal a significant dependency of the simulation results on these parameters. Recent results in literature [1] indicate that larger coarse graining ratios (up to CG = 80) yield predictions for the dispersion pattern within an error corridor of +/- 20%. Hence, the chosen simulation approach based on parcels can be expected to yield reliable results.
- The particle injection velocity determines whether separation or mixing occurs right after injection. That is, the feeding strategy is affecting the particle dispersion. The key dimensionless parameter is the jet to cross flow velocity ratio, which should be at least 0.4 or larger. Hence, the particle injection velocity, or the cross flow velocity should be controlled in order to exploit the separation of particles right after injection.
- The mean penetration depth of comparably large particles (i.e., particles having a large Stokes number) can be increased when reducing the injection angle (i.e. injecting the particles against the cross-flow direction). However, an increased dispersion of particles across the channel can be observed when reducing the injection angle, which might be unwanted in some applications.
- An increased penetration depth of all particles was obtained when using a single air injection nozzle in order to accelerate the air-particle mixture (considered mixture velocity: 10 [m/s]). Hence, the usage of nozzles is expected to improve the particle dispersion in the classifier, and the separation efficiency.
- The simulations of the gas-particle flow in the annular gap region reveal that an increased sharpness of cut can be obtained when injecting the particles at the classifier

air inlet. However, the characteristic diameter of the RRSB-distribution for the rotor product could not be improved, since the particles were not able to penetrate as deep as supposed.

- The use of a particle deflector yielded an increased radial transport of particles. Subsequently, the product quality could be improved.
- Due to the insufficient acceleration of the particle cloud in the tangential direction, large particles violently collide with the rotor blades. A blade design that deflects coarse particles back to the spiral housing is hence important.
- The simulation results of the flow between the rotor blades indicate that a vortex between two neighboring blades is strongly influencing the separation efficiency. This finding is in accordance with previous studies [2–4]. However, previous work has not quantified the separation effect of this vortex, which was studied in detail in this thesis. The vortex between the blades has a negative effect, in the sense that the mean radial inflow velocity of the air is increased. This effect can be modeled using a blockage, or open area ratio. However, the vortex also leads to a substantial (local) increase of the tangential velocity at the inner edge of the blades (i.e., the edge that points toward the rotor axis). Consequently, particles can move faster than the circumferential speed of the rotor, leading to an improved separation due to vortex formation.
- Furthermore, the lowest (mean) pressure difference over the blades was obtained in the case with straight blades. The pressure difference using other blade configurations was at least 75% larger.

Due to the lack of radial transport of particles right after injection and the massive radial inflow directly below the deflector, investigation of a relocated particle feed plane were addressed to future work. However, this can be realized only by a redesign of the classifier housing. Also, vertical pipes or guide plates that guide the injected particles further axially could improve particle dispersion. An alternative would be the combination of multiple air nozzles at the particle feed plane and a particle deflector. This combination is expected to increase the particle dispersion and the radial transport of particles.

## 8.1. References

[1] S. Radl, B.C. Gonzales, C. Goniva, S. Pirker, State of the Art in Mapping Schemes for Dilute and Dense Euler-Lagrange Simulations, in: 10th Int. Conf. CFD Oil Gas, Metall. Process Ind., Trondheim, Norway, 2014: pp. 1–8.

[2] K. Leschonski, K. Legenhausen, Investigation of the flow field in deflector wheel classifiers, Chem. Eng. Process. 31 (1992) 131–136.

[3] P. Toneva, P. Epple, M. Breuer, W. Peukert, K.-E. Wirth, Grinding in an air classifier mill — Part I: Characterisation of the one-phase flow, Powder Technol. 211 (2011) 19–27.

[4] S. Radl, B. Mohan, Simulation of Coarse Particle Slip in an Air Classifier, Graz, Austria, 2013.

# 9. List of Symbols

| | | |
|---|---|---|
| $A$ | cross-sectional area | [$m^2$] |
| $a$ | open area ratio | [-] |
| $all$ | all particles | [-] |
| $C$ | coefficient | [-] |
| CG | coarse graining ratio | [-] |
| $Co$ | courant number | [-] |
| **D** | Darcy coefficient | [$m^{-2}$] |
| $d$ | particle size | [m] |
| $d'$ | characteristic particle size of the RRSB-distribution function | [m] |
| **F** | Forchheimer coefficient | [$m^{-1}$] |
| **f** | force | [N] |
| **g** | gravitational acceleration | [m $s^{-2}$] |
| $I$ | moment of inertia | [kg $m^2$] |
| **I** | unity matrix | [-] |
| $k$ | turbulent kinetic energy | [$m^2$ $s^{-2}$] |
| $\tilde{k}$ | constant | [m $s^{-2}$] |
| $L$ | characteristic dimension | [m] |
| $m$ | mass | [kg] |
| $\dot{m}$ | mass rate | [kg $s^{-1}$] |
| $n$ | number of nozzles | [-] |
| $n$ | characteristic steepness parameter of the RRSB-distribution function | [-] |
| **u** | (average) velocity | [m $s^{-1}$] |
| $V$ | volume | [$m^3$] |
| $P$ | power | [W] |
| $p$ | pressure | [Pa] |
| $Q$ | volumetric flow rate | [$m^3$ $s^{-1}$] |
| $Q_3$ | cumulative particle size distribution (related to particle volume) | [-] |
| $r$ | particle radius | [m] |
| **r** | radial distance | [m] |
| $R$ | radius of circular path | [m] |
| $Re$ | Reynolds number | [-] |
| RMD | relative mean deviation | [-] |
| $S$ | rate of strain | [$s^{-1}$] |

| | | |
|---|---|---|
| **S** | sink term taking Darcy-Forchheimer law into account | [kg $m^{-2}$ $s^{-2}$] |
| *Stk* | Stokes number | [-] |
| **t** | torque | [Nm] |
| *t* | time | [s] |

## *Greek Letters:*

| | | |
|---|---|---|
| $\alpha$ , $\beta$ | angle | [°] |
| $\beta$ | drag coefficient | [kg $m^{-3}$ $s^{-1}$] |
| $\Delta$ | filter width | [m] |
| $\delta$ | Kronecker delta | [-] |
| $\varepsilon$ | turbulence dissipation rate | [$m^2$ $s^{-3}$] |
| $\eta$ | dynamic viscosity | [kg $m^{-1}$ $s^{-1}$] |
| $\mu$ | mass loading | [-] |
| $\nu$ | kinematic viscosity | [$m^2$ $s^{-1}$] |
| $\rho$ | density | [kg $m^{-3}$] |
| $\sigma$ | standard deviation of mean particle position | [m] |
| $\boldsymbol{\tau}$ | viscous stress tensor | [Pa] |
| $\tau$ | characteristic time | [s] |
| $\boldsymbol{\Phi}$ | coupling (interaction) force per unit-volume of the mixture | [kg $m^{-2}$ $s^{-2}$] |
| $\phi$ | volume fraction | [-] |
| $\omega$ | specific turbulence dissipation rate | [$s^{-1}$] |
| $\boldsymbol{\omega}$ | angular frequency | [$s^{-1}$] |

## *Sub- and Superscripts:*

| | |
|---|---|
| $c$ | contact |
| $C$ | Coriolis |
| $cf$ | centrifugal |
| $D$ | drag |
| $div$ | divided void fraction model |
| $f$ | fluid phase (i.e.: air) |
| $fp$ | fluid-particle interaction |
| $h$ | hydraulic |
| $i$ , $j$ | index of summation |
| $in$ | inflow |
| $inject$ | quantity defined at injection |
| $inlet$ | inlet location |
| $k$ | turbulent kinetic energy |
| $max$ | class of largest particles |
| $min$ | class of smallest particles |
| $mix$ | mixture quantity |
| $p$ | particle phase |
| $pr$ | projected |
| $prim$ | primary (physical) quantity |
| $r$ | radial |
| $rel$ | relative quantity |
| $relax$ | relaxation |
| $S$ | Smagorinsky |
| $Stokes$ | quantity related to Stokes flow conditions |
| $SGS$ | subgrid-scale quantity |
| $t$ | terminal |
| $tu$ | Turbulent |
| $wn$ | weighted neighbors void fraction model |
| $\phi$ | Tangential |
| * | critical particle size |

# 10. Appendix A – Setup of the Cross-Flow Simulations

- **controlDict**

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.6                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     cfdemSolverPimpleImEx;

startFrom       startTime;

startTime       0.0;

stopAt          endTime;

endTime         1.0;

deltaT           1.50e-5;

writeControl    adjustableRunTime;

writeInterval   3e-5;

purgeWrite      60;

writeFormat     ascii;

writePrecision  6;

writeCompression compressed;

timeFormat      general;

timePrecision   6;

runTimeModifiable yes;

adjustTimeStep  no;

maxCo           1;

maxDeltaT       1;

libs ( "libfiniteVolumeCFDEM.so" );

functions
(
    probes
    {
        functionObjectLibs ( "libsampling.so" );
        type probes;
        outputControl      timeStep;//
        outputInterval     1;

        probeLocations
        (
            ( 0.000 0.001 0 )
            ( 0.000 0.099 0 )
            ( 0.000 0.199 0 )
            ( 0.000 0.299 0 )
            ( 0.000 0.399 0 )
            ( 0.000 0.499 0 )
            ( 0.000 0.599 0 )
            ( 1.000 0.001 0 )
```

```
            ( 1.000 0.101 0 )
            ( 1.000 0.201 0 )
            ( 1.000 0.301 0 )
            ( 1.000 0.401 0 )
            ( 1.000 0.501 0 )
            ( 1.000 0.601 0 )
        );
        fields
        (
            U UMean voidfractionMean p
        );
    }

    fieldAverage1
    {
        type            fieldAverage;
        functionObjectLibs ("libfieldFunctionObjects.so");
        enabled         true;
        outputControl   outputTime;

        fields
        (
            U
            {
                mean        on;
                prime2Mean  on;
                base        time;
            }

            voidfraction
            {
                mean        on;
                prime2Mean  on;
                base        time;
            }
        );
    }

);


// ************************************************************************* //
```

- **couplingProperties**

```
/*--------------------------------------------------------------------------*\
| =========                 |                                                |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox          |
|  \\    /   O peration     | Version:  1.4                                  |
|   \\  /    A nd           | Web:      http://www.openfoam.org              |
|    \\/     M anipulation  |                                                |
\*--------------------------------------------------------------------------*/


FoamFile
{
    version         2.0;
    format          ascii;

    root            "";
    case            "";
    instance        "";
    local           "";

    class           dictionary;
    object          couplingProperties;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
//=========================================================================//
// sub-models & settings

solveFluidFlow  true;

modelType "A";

couplingInterval 250;

voidFractionModel divided; //centre; //bigParticle;//

locateModel engine; //turboEngine//standard;//

momCoupleModels
(
    implicitCouple
    explicitCouple
    //explicitCoupleSource
);

//forceExplicitForceMapping; //force the use of explicit Lagr-To-Euler force Mapping
//only relevant for Pimple-based solvers

probeModel off;

//smoothingModel off;
//smoothingModel  constDiffSmoothing;
smoothingModel  localPSizeDiffSmoothing;


constDiffSmoothingProps
{
    lowerLimit 0.0;
    upperLimit 1e99;
    smoothingLength 1500e-6;
//    smoothingLengthReferenceField 9000e-6;
}

localPSizeDiffSmoothingProps
{
    lowerLimit 0.0;
    upperLimit 1e99;
    dSmoothingLength 15e-3; //smoothing length for particle diameter smoothing algorithm
                            //should be approximately 2x the length of the coarsest grid cells
    Csmoothing   5.0; //Prefactor for smoothing length for voidfraction, force, Us, and Ksl
//CsmoothingReferenceField  15.0;
//Prefactor for smoothing length for voidfraction, force, Us, and Ksl
    dSmoothingFieldName "dSmoothing";
}
```

```
turbulenceModelType "LESProperties";//"RASProperties";//

IOModel "basicIO";

meshMotionModel   noMeshMotion;

regionModel       allRegion;//differentialRegion;//

dataExchangeModel twoWayMPI;//twoWayFiles;//oneWayVTK;//

averagingModel dense;//dilute;//

clockModel off;

forceModels
(
    //KochHillDrag
    BeetstraDrag
    gradPForce
    viscForce
//    fieldTimeAverage
//    volWeightedAverage
);

//================================================================================//
// sub-model properties

fieldTimeAverageProps
{
    startTime 0.1;

    scalarFieldNames
    (
        "voidfraction"
    );

    vectorFieldNames
    (
        "Us"
    );
}

totalMomentumExchangeProps
{
    implicitMomExFieldName "Ksl";
    explicitMomExFieldName "none";
    fluidVelFieldName "U";
    granVelFieldName "Us";
    densityFieldName "rho";
}

gradPForceProps
{
    pFieldName "p";
    densityFieldName "rho";
    velocityFieldName "U";
    interpolation;
}

viscForceProps
{
    velocityFieldName "U";
    densityFieldName "rho";
    interpolation;
}

ArchimedesProps
{
    densityFieldName "rho";
    gravityFieldName "g";
}

volWeightedAverageProps
```

```
{
    scalarFieldNames
    (
        voidfraction
    );
    vectorFieldNames
    (
    );
    upperThreshold 0.999;
    lowerThreshold 0;
//    verbous;
}

implicitCoupleProps
{
    velFieldName "U";
    granVelFieldName "Us";
    voidfractionFieldName "voidfraction";
}
explicitCoupleProps
{
    //fLimit (0 0 0);
}

WenYuDragProps
{
    velFieldName "U";
    densityFieldName "rho";
    voidfractionFieldName  "voidfraction";
//    dPrim       100e-6;
//    dParcelRef  100e-6;
    interpolation;
//    useEMMSDragModel;
//    verbose;
}

KochHillDragProps
{
    velFieldName "U";
    densityFieldName "rho";
//    dPrim       100e-6;
//    dParcelRef  100e-6;
    voidfractionFieldName "voidfraction";
    interpolation ;
}

BeetstraDragProps
{
    velFieldName "U";
    densityFieldName "rho";
    gravityFieldName "g";
    rhoParticle     1500.;
//    dPrim           20e-6;          // only used in octave postproc!!!
    voidfractionFieldName "voidfraction";
    interpolation ;
/*    useFilteredDragModel ;*/
/*    useParcelSizeDependentFilteredDrag ;*/
/*    k       0.05;*/
/*    aLimit  0.0;*/
//    verbose ;
}


twoWayMPIProps
{
    maxNumberOfParticles 10100;
    liggghtsPath "../DEM/in.jet";
}
bigParticleProps
{
    alphaMin 0.1;
    scaleUpVol 1.0;
    maxCellsPerParticle 500;
}
centreProps
{
    alphaMin 0.1;
}

dividedProps
{
    alphaMin 0.1;
    scaleUpVol 1.0;
}

engineProps
{
    treeSearch true;
}
// ************************************************************************* //
```

- **fvSchemes**

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.6                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default         Euler;
}

gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
    grad(U)         Gauss linear;
}

divSchemes
{
    default         Gauss linear;
    div(phi,U)      Gauss limitedLinearV 1;
    div(phi,k)      Gauss limitedLinear 1;
    div(phi,epsilon) Gauss limitedLinear 1;
    div(phi,R)      Gauss limitedLinear 1;
    div(R)          Gauss linear;
    div(phi,nuTilda) Gauss limitedLinear 1;
    div((viscousTerm*dev(grad(U).T()))) Gauss linear;
    div((nu*dev(grad(U).T()))) Gauss linear;
    div((nuEff*dev(grad(U).T()))) Gauss linear;
}

laplacianSchemes
{
    default         Gauss linear corrected;
    laplacian(viscousTerm,U) Gauss linear corrected;
    laplacian(nu,U) Gauss linear corrected;
    laplacian(nuEff,U) Gauss linear corrected;
    laplacian((1|A(U)),p) Gauss linear corrected;
    laplacian((voidfraction2|A(U)),p) Gauss linear corrected;
    laplacian(DkEff,k) Gauss linear corrected;
    laplacian(DepsilonEff,epsilon) Gauss linear corrected;
    laplacian(DREff,R) Gauss linear corrected;
    laplacian(DnuTildaEff,nuTilda) Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
    interpolate(U)  linear;
}

snGradSchemes
{
    default         corrected;
}

fluxRequired
{
    default         no;
    p               ;
}
```

- **fvSolution**

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.6                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    p
    {
        solver          GAMG;
        tolerance       1e-7;
        relTol          1e-4;
        smoother        DICGaussSeidel;
        nPreSweeps      0;
        nPostSweeps     2;
        nFinestSweeps   2;
        cacheAgglomeration true;
        nCellsInCoarsestLevel 50;
        agglomerator    faceAreaPair;
        mergeLevels     2;
    }

    pFinal //Only relevant if Pimple-type solver used!!
    {
        solver          GAMG;
        tolerance       1e-7;
        relTol          0;
        smoother        DICGaussSeidel;
        nPreSweeps      0;
        nPostSweeps     2;
        nFinestSweeps   2;
        cacheAgglomeration true;
        nCellsInCoarsestLevel 50;
        agglomerator    faceAreaPair;
        mergeLevels     2;
    }

    U
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-05;
        relTol          0;
    }

    UFinal //Only relevant if Pimple-type solver used!!
    {
        $U
        tolerance       1e-05;
        relTol          0;
    }

    k
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-05;
        relTol          0;
    }

    kFinal //Only relevant if Pimple-type solver used!!
    {
```

```
        $k
        tolerance       1e-05;
        relTol          0;
    }

    epsilon
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-05;
        relTol          0;
    }

    R
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-05;
        relTol          0;
    }

    nuTilda
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-05;
        relTol          0;
    }

    voidfraction
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-09;
        relTol           1e-06;
    }

    Ksl
    {
        $voidfraction
    }

    f
    {
        $voidfraction
    }

    fSmooth
    {
        $voidfraction
    }

    Us
    {
        $voidfraction
/*        tolerance       1e-09;*/
/*        relTol           0;*/
    }

    dSmoothing
    {
        $voidfraction
    }


}

PISO
{
    nCorrectors     2;
    nNonOrthogonalCorrectors 0;
    pRefCell        0;
    pRefValue       0;
}
```

```
//Only relevant if Pimple-type solver used!!
PIMPLE
{
    nOuterCorrectors 1;
    nCorrectors     2;
    nNonOrthogonalCorrectors 0;
}

relaxationFactors
{
    fields
    {
        p       0.8;
    }
    equations
    {
        "U.*"           0.7;
        "k.*"            1.0;
        "epsilon.*"    0.1;
    }
}


// ************************************************************************* //
```

- **transportProperties**

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.6                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      transportProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //


transportModel  Newtonian;

nu              nu [ 0 2 -1 0 0 0 0 ] 1.7364e-05;

CrossPowerLawCoeffs
{
    nu0             nu0 [ 0 2 -1 0 0 0 0 ] 1e-06;
    nuInf           nuInf [ 0 2 -1 0 0 0 0 ] 1e-06;
    m               m [ 0 0 1 0 0 0 0 ] 1;
    n               n [ 0 0 0 0 0 0 0 ] 1;
}

BirdCarreauCoeffs
{
    nu0             nu0 [ 0 2 -1 0 0 0 0 ] 1e-06;
    nuInf           nuInf [ 0 2 -1 0 0 0 0 ] 1e-06;
    k               k [ 0 0 1 0 0 0 0 ] 0;
    n               n [ 0 0 0 0 0 0 0 ] 1;
}

// ************************************************************************* //
```

- **LESProperties**

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  1.6                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      LESProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

LESModel        oneEqEddy;

delta           cubeRootVol;

printCoeffs     on;

cubeRootVolCoeffs
{
    deltaCoeff      1;
}

PrandtlCoeffs
{
    delta           cubeRootVol;
    cubeRootVolCoeffs
    {
        deltaCoeff      1;
    }

    smoothCoeffs
    {
        delta           cubeRootVol;
        cubeRootVolCoeffs
        {
            deltaCoeff      1;
        }

        maxDeltaRatio   1.1;
    }

    Cdelta          0.158;
}

vanDriestCoeffs
{
    delta           cubeRootVol;
    cubeRootVolCoeffs
    {
        deltaCoeff      1;
    }

    smoothCoeffs
    {
        delta           cubeRootVol;
        cubeRootVolCoeffs
        {
            deltaCoeff      1;
        }

        maxDeltaRatio   1.1;
    }

    Aplus           26;
    Cdelta          0.158;
}

smoothCoeffs

{
    delta           cubeRootVol;
    cubeRootVolCoeffs
    {
        deltaCoeff      1;
    }

    maxDeltaRatio   1.1;
}


// ************************************************************************* //
```

# 11. Appendix B – Results of the Cross-Flow Simulations

This Section provides further data including void fraction fields, particle size distributions in the channel and flow fields of different cross-flow simulations.

- **Sensitivity to coarse graining ratio**

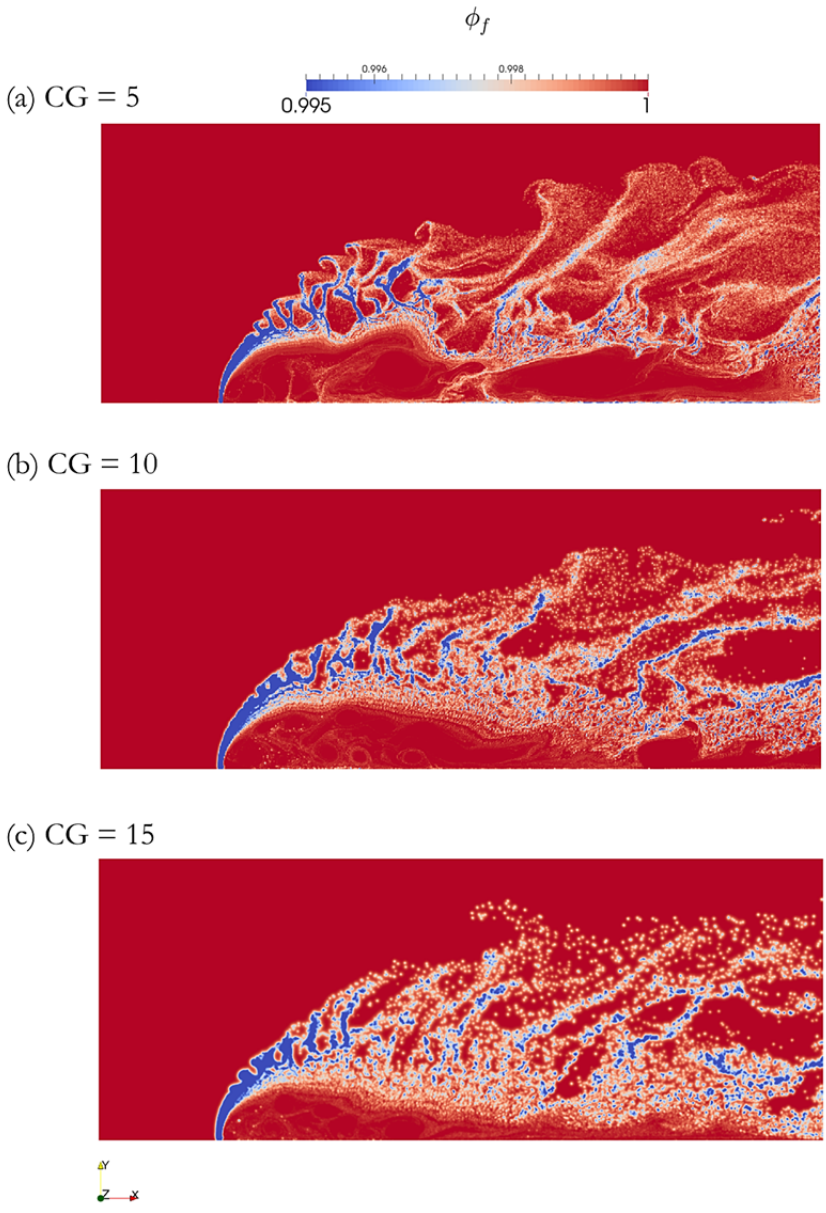


**Figure 11-1: Air volume fraction fields for different coarse graining ratios; particle injection velocity = 15 [m/s]; (a) CG = 5, (b) CG = 10, (c) CG = 15 (for all other parameters see the base case definition in Section 4.1).**

- **Influence of particle injection velocity**

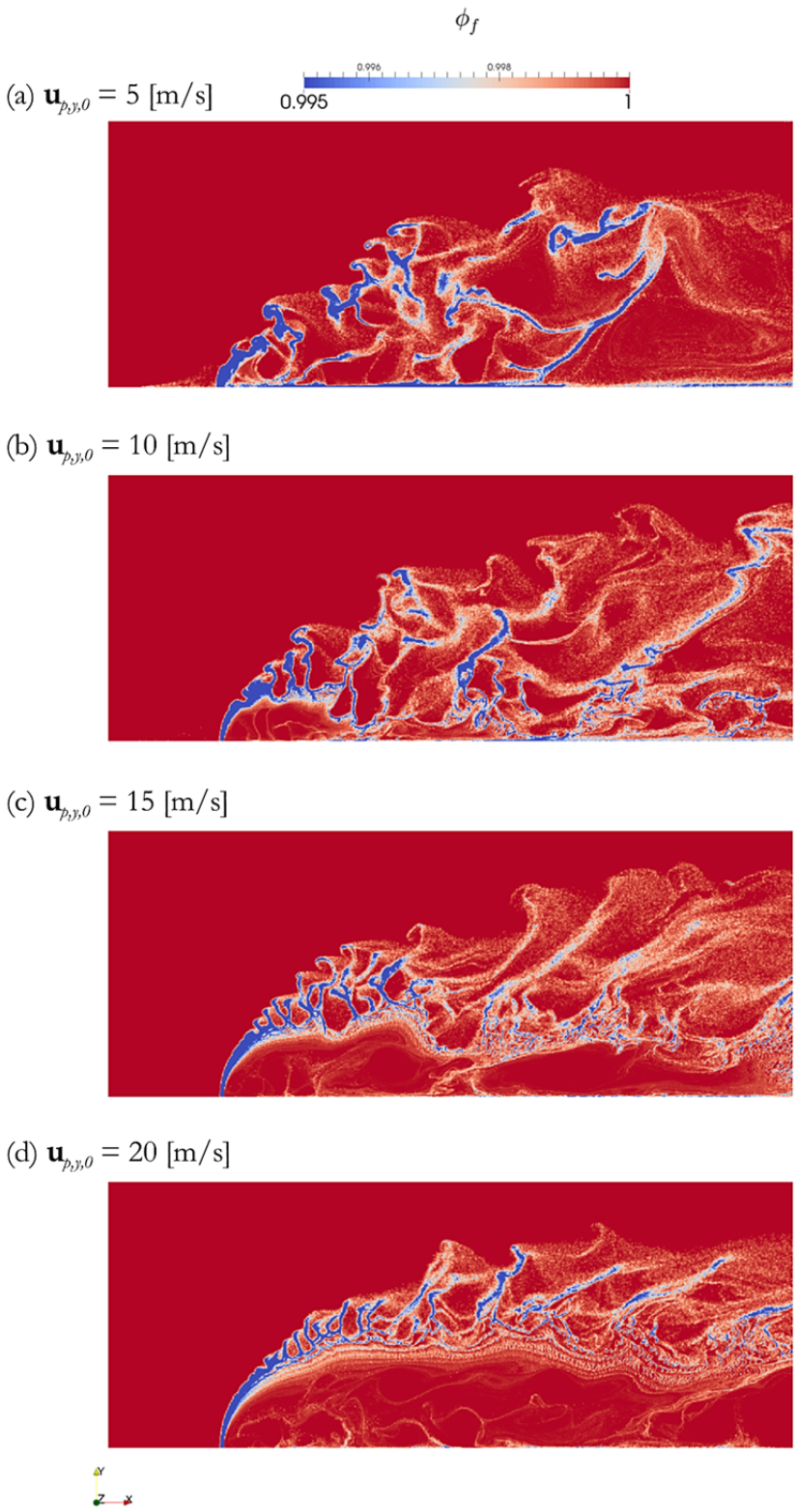


**Figure 11-2: Air volume fraction fields for different particle injection velocities; CG = 5; (a) $u_{p,y,0}$ = 5 [m/s], (b) $u_{p,y,0}$ = 10 [m/s], (c) $u_{p,y,0}$ = 15 [m/s], (d) $u_{p,y,0}$ = 20 [m/s] (for all other parameters see the base case definition in Section 4.1).**

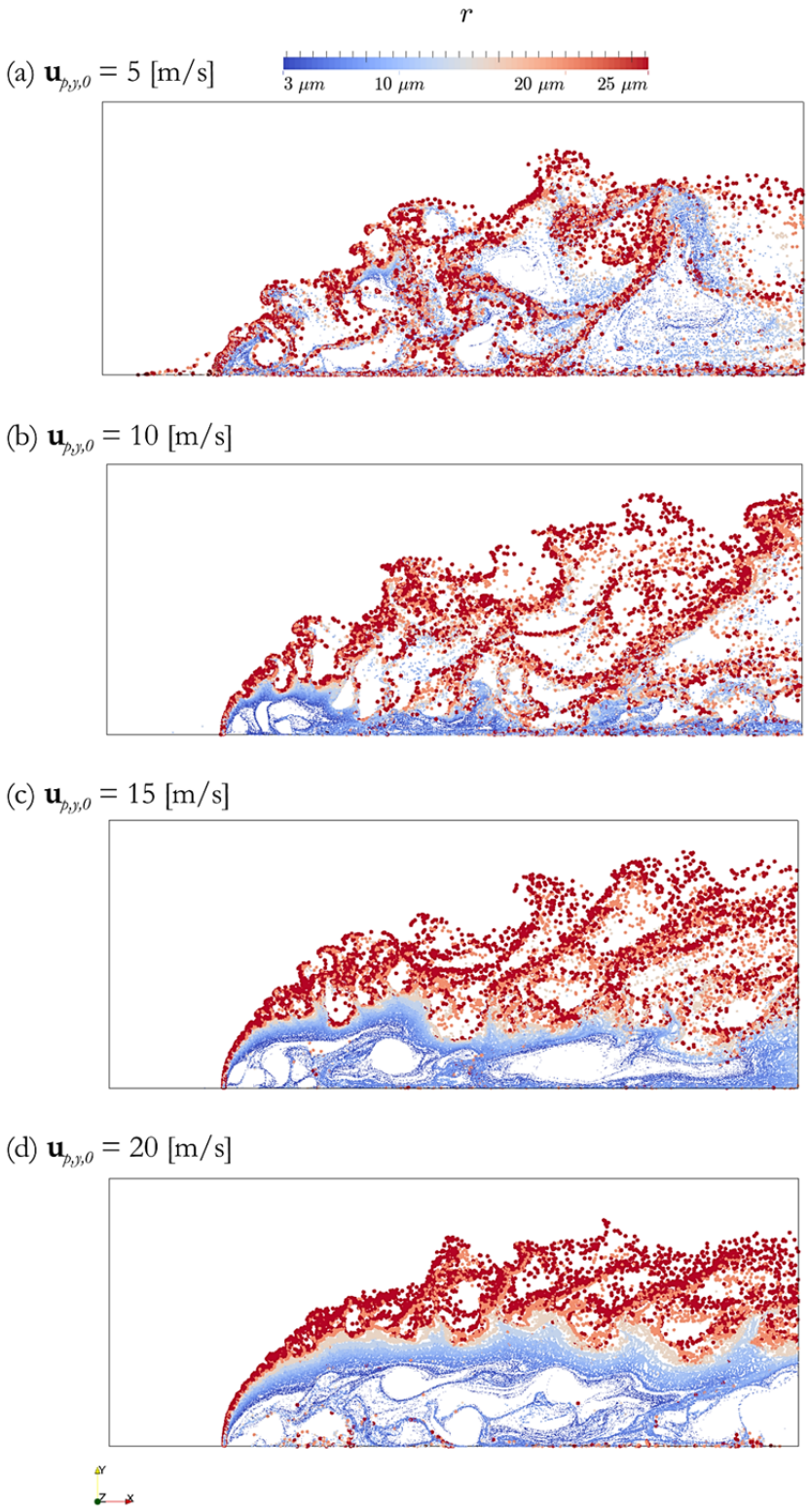


**Figure 11-3: Particle distribution in the channel for different injection velocities; CG = 5; (a) $u_{p,y,0}$ = 5 [m/s], (b) $u_{p,y,0}$ = 10 [m/s], (c) $u_{p,y,0}$ = 15 [m/s], (d) $u_{p,y,0}$ = 20 [m/s] (particles are colored according their size; for all other parameters see the base case definition in Section 4.1).**

- **Effect of different particle injection angles**

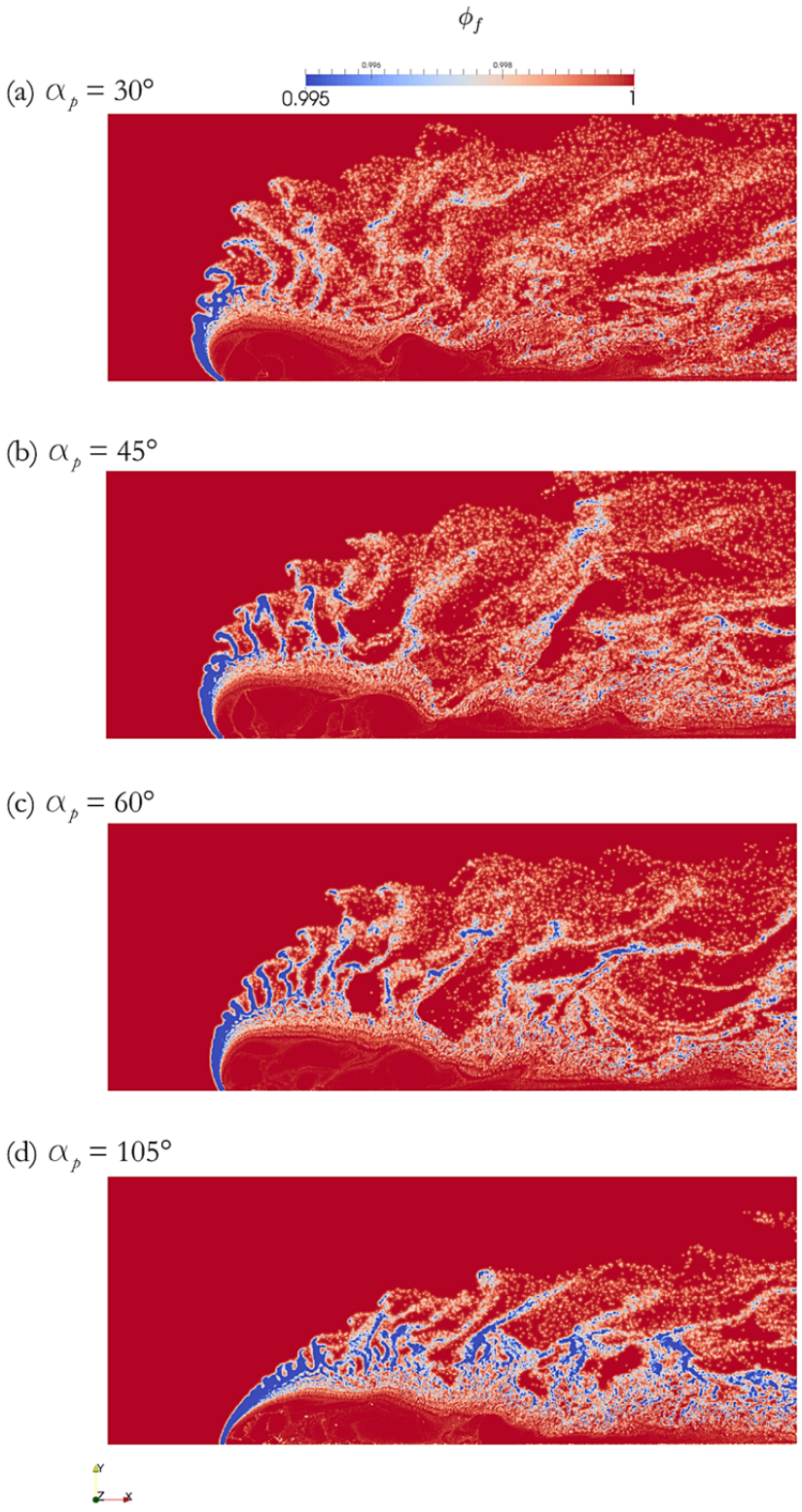


**Figure 11-4: Air volume fraction fields for different particle injection angles; CG = 10; (a) $\alpha_p$ = 30°, (b) $\alpha_p$ = 45°, (c) $\alpha_p$ = 60°, (d) $\alpha_p$ = 105° (for all other parameters see the base case definition in Section 4.1).**

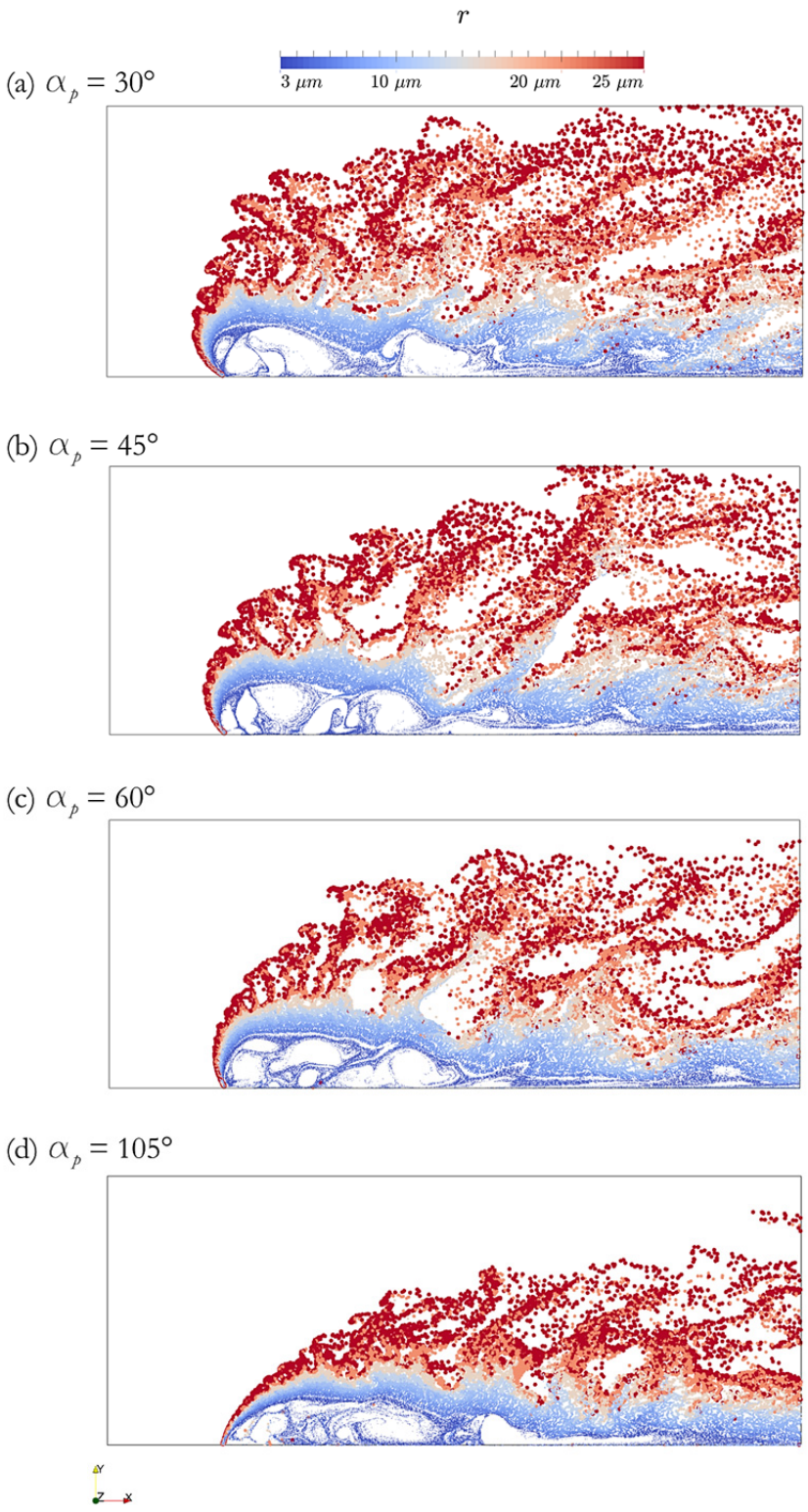


**Figure 11-5: Particle distribution in the channel for different injection angles; CG = 10; (a) $\alpha_p$ = 30°, (b) $\alpha_p$ = 45°, (c) $\alpha_p$ = 60°, (d) $\alpha_p$ = 105° (particles are colored according their size; for all other parameters see the base case definition in Section 4.1).**

- **Effect of multiple nozzles on particle dispersion and flow field**

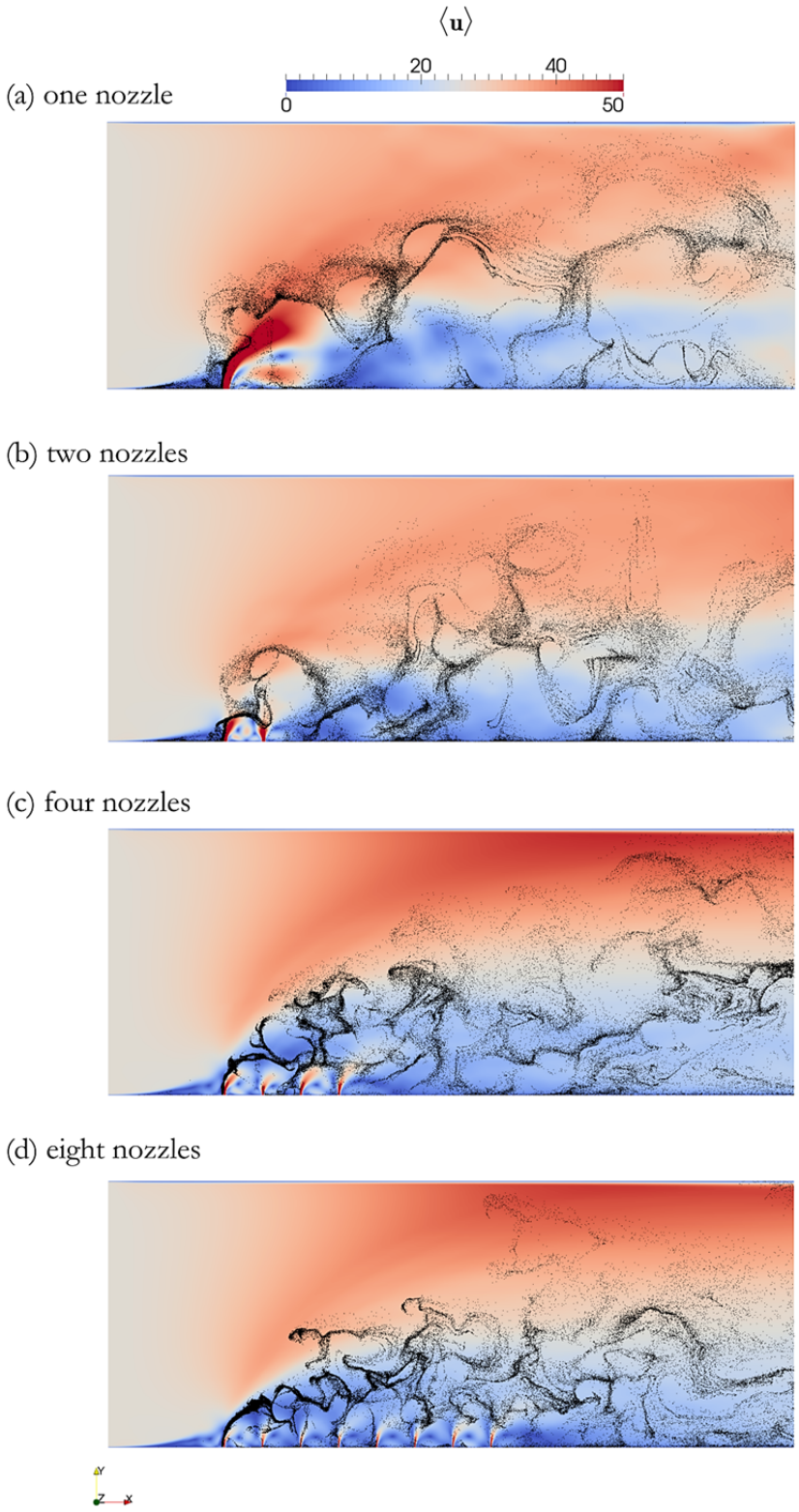


**Figure 11-6: Time averaged air velocity fields for different nozzle configurations; CG = 10, particle injection velocity = 5 [m/s], $u_{mix}$ = 10 [m/s]; (a) one nozzle, (b) two nozzles, (c) four nozzles, (d) eight nozzles (for all other parameters see the base case definition in Section 4.1).**

# 12. Appendix C – Results of the Classifier Simulations

Additional results of the annular gap region simulations, such as the position and velocity statistics of particles entering the rotor region are given in this section.

- **Typical velocity and position statistics of particles approaching the rotor**

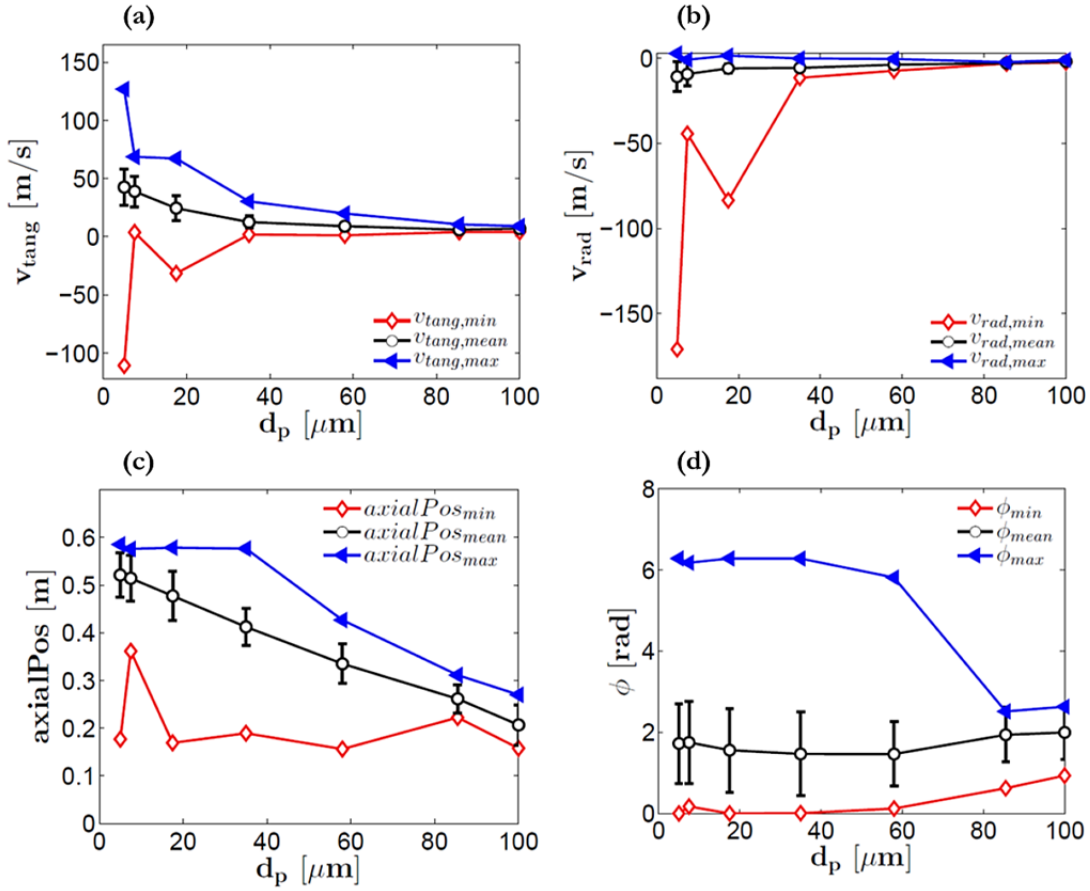


**Figure 12-1: Velocity and position statistics of particles approaching the rotor region; (a) tangential velocity, (b) radial velocity, (c) axial position and (d) radial position relative to the x-axis (anticlockwise).**