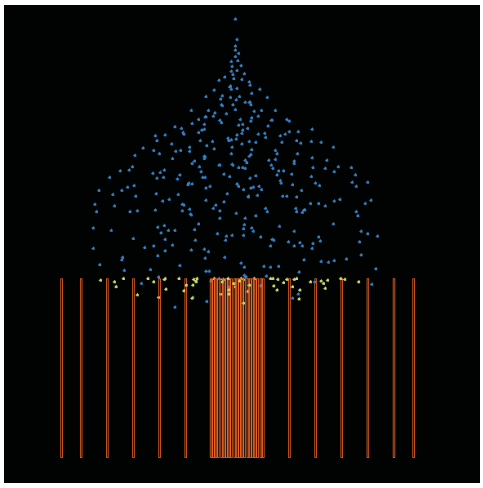
Master Thesis

TU Delft
Faculty of Applied Sciences
Sub faculty Chemical Engineering
Particle Technology Group

Modelling and Measuring Droplet Trajectories in an EHDA Spray



Tom Winkels

Guldeland 49
2291 VH Wateringen
The Netherlands

Supervision:
Ir. K.B. Geerse
Dr. rer. nat. S. Luding
Dr. Ir. J.C.M. Marijnissen

Period:
January 2002 - October 2002

Preface

This report is the Master thesis of the studies Chemical Engineering at the TU Delft. The work has been done in the Particle Technology Group of the sub faculty Chemical Engineering at the faculty Applied Sciences.

For accomplishing the work I want to thank a few people. First I want to thank Kees Geerse for his supervision, cooperation and ideas, especially concerning the measurements done with the PDPA. A second word of thanks to Stefan Luding for his great help developing the model in the C++ programming language and always willing to help me with anything. Without his expertise this work could not have been done. I want to thank Jan Marijnissen too for the coordinating discussions together with Kees and Stefan. And at last a thanks to everybody in the Particle Technology Group who has helped me. For example Sjaak Verdoold helped me a lot with software or computer problems and several other people lent me their computer for doing simulations.

Tom Winkels

Summary

The subject of this Master thesis is both modelling and measuring the trajectories of charged droplets in an ElectroHydroDynamic Atomisation (EHDA) spray.

The experimental set-up existed mainly of a nozzle at 18.0 kV and a grounded metal cylinder about 0.2 m underneath that nozzle, acting as the target for droplet deposition. A mixture of 90.0 $wt\%$ ethanol and 10.0 $wt\%$ triethylene glycol was sprayed at a volume flow rate of 7.0 ml/h. With a Phase Doppler Particle Analyzer (PDPA) velocity and size measurements are performed in a half-plane in the spray. This resulted in a two-dimensional density and velocity "scan" of the spray in the plane through the nozzle and parallel to the length of the target cylinder. A deposition analysis is carried out, showing the elliptical deposition pattern of the dense spray on the target. Also the deposition on the system boundary walls is observed qualitatively and found to be very weak, but agreeing with the PDPA measurements.

A model is developed to simulate the spray from droplet production up to the droplet deposition. Besides several rational assumptions and simplifications, the main idealization is the assumption that the ambient air is at rest. Simulating the real spray is already difficult enough because the collective behaviour of such a spray is very complex, involving the enormous amount of droplets interacting via Coulomb repulsions. The droplet production rate in the order of 10^6 droplets s^{-1} is impossible to simulate with presently available computing power. In order to circumvent the simulation of the enormous amount of droplets a scale-up method influencing the Coulomb interactions is introduced within the framework of the model. This allows to simulate the EHDA spray with fewer droplets than in reality.

From the simulation similar results as from the experiments are obtained and consequently compared to the experimental results. This comparison shows that the model simulates the EHDA spray quite well. The deposition pattern of the droplets on the cylinder target in the simulated situation is an ellipse with almost the same extreme deposition dimensions, because this was a criterion for the choice of the scale-up factor for the final simulation. The shape of the simulated spray resembles the experimental results quite well. Only close to the nozzle (up to about 0.01 m in vertical direction from the droplet production point) the spray is too narrow. As a result the simulated spray stays slightly less broad than the experimental spray until deposition on the target. The qualitative comparison of the velocity profiles of the simulated and real sprays show very good agreement. But the major difference is that the magnitude of the velocities of the droplets in both horizontal and vertical direction is much higher in the real spray than in the model spray.

The initial properties of newly produced droplets are important for the trajectories

of the droplets. Since the measurement of these droplet properties and initial droplet density cannot be done with a PDPA analysis, they are unknown. Therefore a droplet production with limited random position and velocity is assumed.

The result that the simulated spray is too narrow in the area close to the nozzle, and as a result also slightly at higher distance from the nozzle, is assigned to the difference in the Coulomb interactions between the real and model situations, in that area of high droplet density. Further downwards the spray the scale-up method reproduces the effect of the Coulomb interactions well.

The major difference between the model and the reality is the difference in the air hydrodynamics and thus in the drag force. In the model the ambient air is assumed to be at rest. In reality the air flow is expected to be important in all stages of the spray. Close to the nozzle the air velocity and flow are expected to be high and turbulent respectively. Lower in the spray the air velocity could also be significantly high due to an established flow pattern. To what extent the error in the resulting drag force can account for the observed discrepancies can not be determined by this study, but is expected to explain the difference in the magnitude of the velocity, and to some extent will affect the shape of the spray.

From this work it is concluded that the model can simulate the EHDA spray quite well, but it has to be extended including air hydrodynamics and an advanced model for the high droplet density area very close to the nozzle. In the model the scale-up factor is the only adjustable parameter. A definitive value of this parameter might be determined if the scale-up method is validated for extrapolation towards the experimental spray. This can only be achieved if the model is extended with the just mentioned features first. It is expected on the basis of this work that the model including the scale-up method and the yet to be developed extensions, can be more valuable in the future for simulating EHDA sprays.

Contents

Introduction 1

The technological world is very broad consisting of many areas such as chemical engineering, civil engineering, electronical engineering and physics. In all these areas different things are designed: concrete constructions such as reactors or buildings, electronical equipment, or all sorts of combinations involving various processes. These can consist of very complex phenomena. This report concerns the process of "electrospraying". The applications of an electrospray are numerous, such as drug production (for example for asthma patients) and crop protection.

Nowadays a lot of the scientific phenomena are formulated with mathematical equations that only can be solved with the aid of a computer. In that way complex physical events, for example concerning many particles, can be modelled and simulated. This numerical modelling is used in many fields of science and is still penetrating more areas, while computing power is increasing every day. The reason why models are needed is that experiments are often very expensive, very difficult or dangerous to carry out. Advantages of models and simulations are not only the reduced costs and easy execution. Also very little time is needed as compared to doing field experiments (and retrieving data from them). So modelling can be a quick, easy and cheap way of simulating an experiment and subsequently optimising the process by applying predictions from the simulation.

But models have their limitations. Simplifying assumptions have to be made to formulate a model in equations and simulate a complicated process. Finally the model has to be verified with experimental data. So predictions based on a simulation must be verified and they can not be seen as separate entities. Also if discrepancies are uncovered, they can help to understand the consequences of the assumptions made and thus help to reformulate the model. From this point of view it is clear that models should not be too complex but must contain the essential features of the process and are able to describe the process in a realistic way. The just explained way of using a model to simulate an experiment is schematically presented in Figure 1.1.

The graduation work presented in this report is set up in this spirit. The subject is both measuring and modelling the trajectories of charged droplets in an ElectroHydroDynamic Atomisation (EHDA) spray (also known as electrospray). This means a model is made to simulate the complex collective behaviour of such a spray with its enormous amount of droplets involved. Within this framework of the model a scaling correlation between the droplet number and the droplet charge is presented, in order to circumvent the simulation of the enormous amount of droplets. Finally, the model is verified with the results from several experiments.

Chapter 2 treats the experimental subjects. First the process of EHDA spraying will

be discussed. The second section treats the experimental verification measurements that are done with a Phase Doppler Particle Analyser (PDPA). The principles of this analysis, the system set-up and the measurements that are done with the PDPA are subsequently discussed in this section. Finally the verification measurements involving the deposition of the charged droplets are treated.

Chapter 3 involves the modelling part of the work done. At first an overview of the development of the modelling and programming is given. Subsequently the equations and assumptions formulating the model are presented. The rest of the chapter involves some important modelling parameters and methodes. First the implementation of the external electric field present in the system is treated. Next the chosen initial properties for the produced droplets are explained. Finally the main part of the model is presented: the scale-up method that should allow to simulate the EHDA spray behaviour with reasonable computing effort.

The results and discussions are given in Chapter 4. The conclusions from that are drawn in the following chapter. Finally some recommendations for future work are given in Chapter 6.

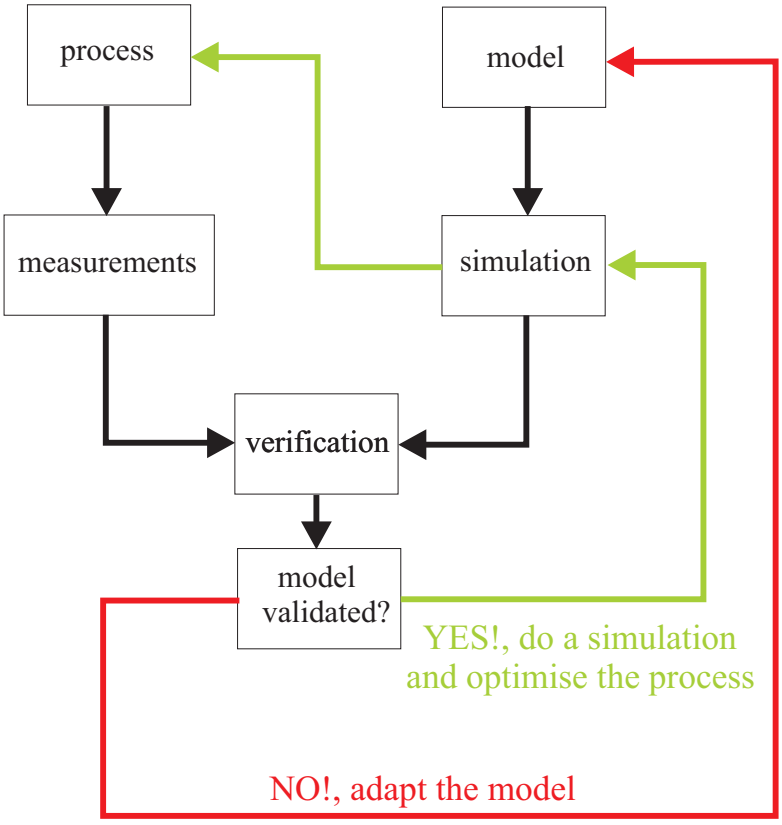


Figure 1.1: Schematic way of using a model to simulate experiments and optimise a process.

Experiment 2

In this chapter all experimental processes and measurements are treated. In the first section the spraying process using EHDA, in the so-called cone-jet mode, is explained. The next section explains the PDPA analysis. At first the principles of the technique are given. Subsequently the experimental set-up is explained, also in relation to the model system set-up. Next the PDPA measurements are discussed in detail. The last section involves a deposition analysis, treating the deposition of the charged droplets on the target and the walls and the ground of the system volume.

2.1 EHDA

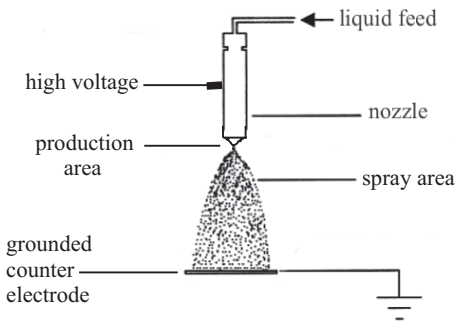


Figure 2.1: Schematical EHDA spraying system.

ElectroHydroDynamic Atomisation (EHDA) is a method to produce very fine charged (of equal sign) droplets from a liquid by using a nozzle and an electric field. A typical experimental set-up is presented in Figure 2.1.

A liquid is pumped at low flow rate, a few ml/h, through a nozzle at which a high voltage is applied. Some grounded counter electrode is placed under the nozzle. This is the target on which the droplets are deposited. Due to the difference in voltage between the nozzle and the target an (external) electric field is created. The Coulomb interactions of the charges inside the liquid, emerging from the nozzle, and the applied electric field, cause an acceleration and consequent atomization of the liquid. A spray of charged droplets is formed, the electrospray or EHDA spray. Depending on the internal properties of the liquid (mainly electrical conductivity, surface tension, viscosity, density and permittivity) and the properties of the external electric field (due to applied voltage, experimental configuration and atmosphere), different modes of EHDA can occur. This can be seen in Figure 2.2 and be read in References [4] and [7].

The process of EHDA spraying can be divided in several areas, as seen in Figure 2.1. The first area is very close to the nozzle, the droplet production area. The liquid emerges from the nozzle and due to some break up mechanism the charged droplets are produced. This production area will be discussed in detail in Subsection 2.1.1.

The second area is then the spray area and is of concern in this report. When the charged droplets are produced they travel through the air influenced by the forces acting on them. Due to the electric field, the charged droplets move towards the grounded target. In Section 3.2 all forces acting on the droplets will be treated in detail. An important force is the Coulomb interaction force between the droplets. Due to the high droplet charges of equal sign the droplets repel each other. Consequently no agglomeration of the droplets occurs and the spray becomes wider. The spray develops through the air until finally the droplets are deposited on the target.

This deposition could be seen as a third area of the EHDA spraying process. This is dependent on the type of counter electrode used. When the conductivity of the counter electrode is low the charge of the deposited droplets stays on the electrode for some time. These charges repel approaching droplets and can alter the trajectories of these droplets. In this case the counter electrode is from metal and the charges from depositing droplets are assumed to flow away instantaneously to earth. Therefore incoming droplets will not be repelled by charge on the target and the deposition is just seen as the end of the spray area.

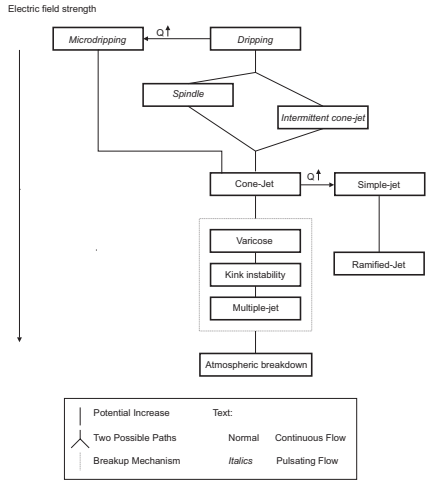


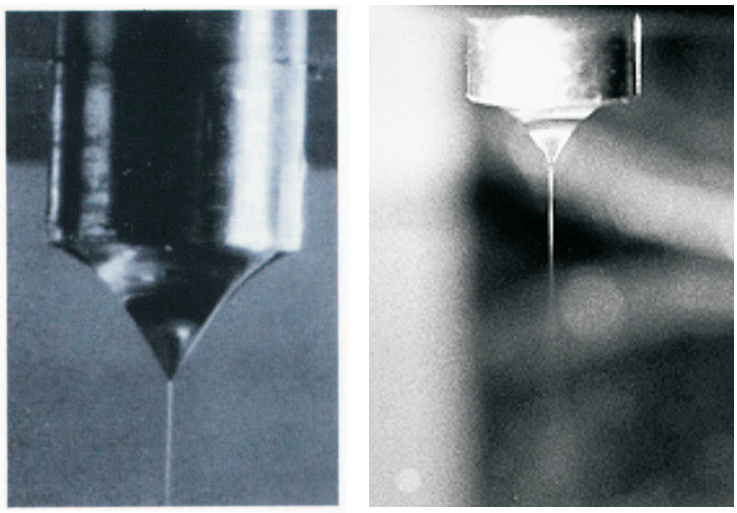
Figure 2.2: Different modes of EHDA.

2.1.1 Cone-jet mode

The mode of EHDA that is of interest is the so-called cone-jet mode. This mode is typically used to produce fine charged droplets. The name of the cone-jet mode will be explained first and after that some properties and applications of the cone-jet mode spray are described.

When a liquid drop emerges from the nozzle it is round due to the surface tension of the liquid. By applying the electric field as discussed above the shape of that drop

changes, namely into a conical shape. This cone is the so-called Gilbert-Taylor cone, named after the persons who wrote down about this phenomena first (Gilbert) and who investigated it extensively (Taylor). From the apex of this cone a jet, a very narrow straight liquid flow, is drawn. Due to a Rayleigh break up mechanism the jet will break up at its end into very small charged droplets forming the electrospray. The nozzle, the Gilbert-Taylor cone and the jet are displayed in Figure 2.3. In Figure 2.3 (b) the jet break up into the spray can be seen, although vaguely.



(a) Gilbert-Taylor cone with emerging jet.

(b) Gilbert-Taylor cone with emerging jet and spray.

Figure 2.3: Fotographs of an EHDA spray in the cone-jet mode.

The cone-jet mode is of importance for three main reasons. First it allows the production of aerosols with a very large range of droplet sizes. Second this mode is achievable for a very wide range of liquids with different properties in terms of conductivity, viscosity and surface tension. Third the size distribution of the droplets produced can be monodisperse, bimodal or polydisperse. Therefore the applications of such an electrospray are numerous, such as electrohydrodynamic mass-spectrometry, thin film production, drug production (for example for asthma patients) and crop protection.

Hartman did a lot of research in the droplet production area of the EHDA spray. Extensive modelling of the Gilbert-Taylor cone and the jet break up has been done [6]. Figure 2.4 shows the Gilbert-Taylor cone model with the forces acting on the fluid. The droplets will be produced with monodisperse size distribution in the so-called varicose jet break up regime. The diameter of the droplets d Hartman calculated with his scaling laws presented in Reference [4]. This diameter is:

$$d_i = 1.76 \left(\frac{\rho \varepsilon_0 \phi^4}{I^2} \right)^{\frac{1}{6}}, \tag{2.1}$$

with 1.76 an experimental prefactor, ρ the liquid density, ε_0 the vacuum permittivity, ϕ the liquid volume flow rate and I the electric current through the Gilbert-Taylor cone.

To calculate the electric current I, ϕ and the following liquid properties are needed: ρ, γ the surface tension and K the conductivity. The expected droplet production time t_{prod} is then given by the ratio of the droplet volume and the liquid volume flow rate:

$$t_{prod} = \frac{\frac{1}{6}\pi d_i^3}{\phi}. \tag{2.2}$$

The work presented in this report can be seen as a next step in the modelling of an EHDA spray, namely the modelling of the trajectories of the charged droplets produced by EHDA in the spray area, from the nozzle to a desired target.

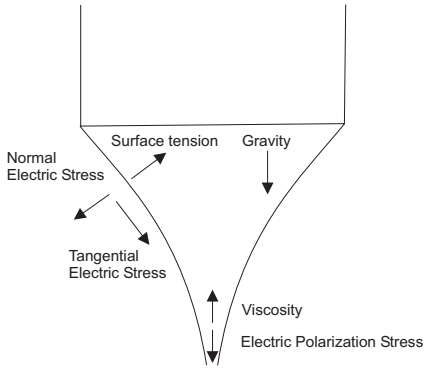


Figure 2.4: The Gilbert-Taylor cone and the forces acting on it.

2.2 PDPA analysis

To create an image of the trajectories of the charged droplets (in the spray area of) the EHDA spray, measurements with a Phase Doppler Particle Analyser (PDPA) are done. The principles of the PDPA analysis method will be discussed briefly. Next the experimental and model system set-up are explained. Finally the measurements done with the PDPA are discussed in detail.

2.2.1 Method principles

The PDPA analysis is based upon light scattering interferometry. This principle uses the wavelength of light as the measurement scale and, as such, the performance is not as easily degraded as it is for systems using light scattering intensity for the estimation of the particle size nor does it require frequent calibration. Therefore PDPA analysis has generally performed well in difficult applications such as in dense sprays produced by gas turbine and rocket injectors and in highly turbulent combustion environments.

According to the schematic presentation given in Figure 2.5 the principles of the PDPA analysis method will be briefly explained. This explanation is done for the equipment used for this work. Further details can be found in the manuals accompanying the equipment.

Initially laser beams are produced with a multi-line Argon-ion laser (Spectra-Physics Stabilite 2017 with Spectra-Physics Model 2250 feed and Spectra-Physics 2670 remote

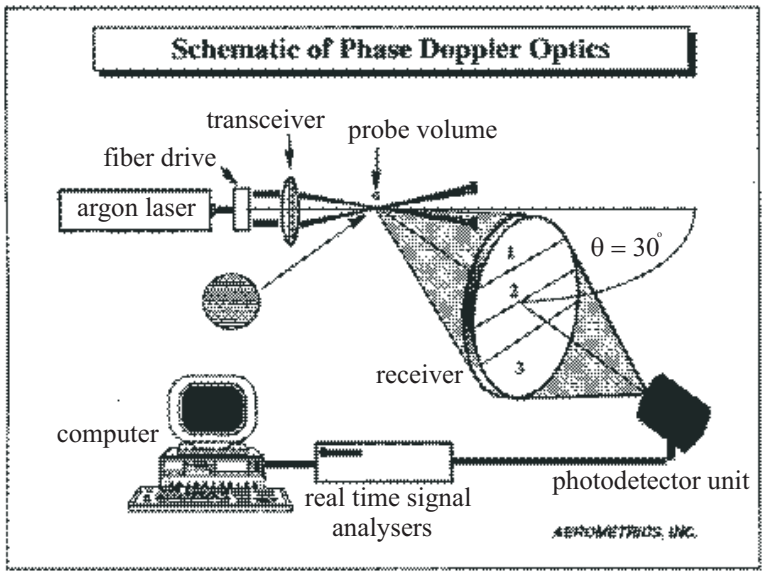
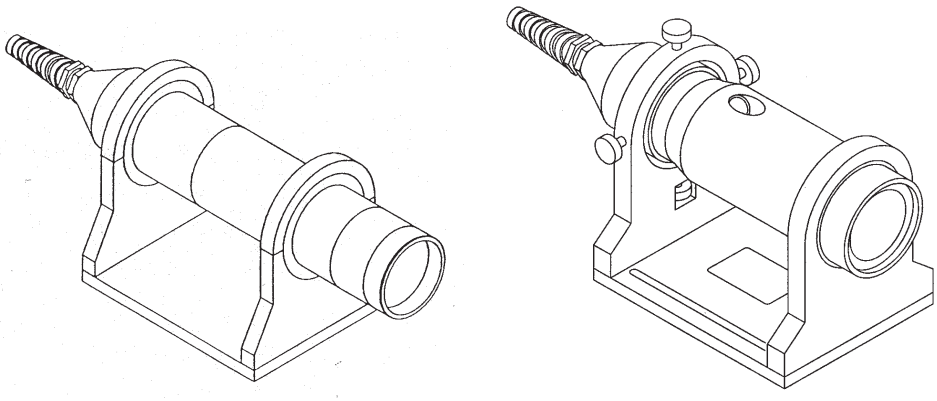


Figure 2.5: Schematic presentation of the PDPA analysis.

control). Laser light with several wave lengths is produced. Subsequently the laser beams are manipulated by a fiber drive (TSI Aerometrics). This instrument provides the laser beams needed for the measurements. Inside the fiber drive a Bragg cell splits the incoming beam into two beams of equal intensity. After that two dispersion prisms separate the beams into three colors: green , blue and violet light. The violet laser light is not used in this set up. Finally, the individual beams are launched into four optical fibers, two for the green and two for the blue beam. The four produced laser beams are transmitted by a transmitter/receiver, the "transceiver". The beams cross exactly at one point at some typical distance away from the transceiver. This is the measuring point and is actually as a small volume with a cross section in the order of 1 mm.



(a) Typical PDPA transceiver (b) Typical PDPA receiver

Figure 2.6: Equipment of a PDPA.

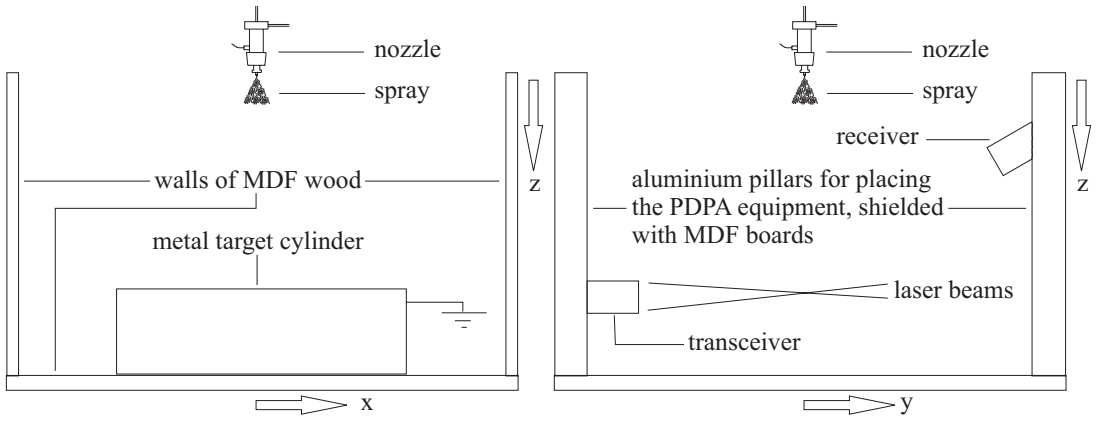
When a droplet travels through this measuring volume the transceiver collects the light scattered by the droplet, while another receiver does the same. This receiver is placed under an angle of 30° with respect to the transceiver, explained in Subsection 2.2.2 too. In Figure 2.6 a typical transceiver (a) and a typical receiver (b) are shown. The signals caught by the receivers are transformed by a Photodetector Unit (TSI Aerometrics) into an electrical current and are subsequently processed by two Real Time Signal Analysers (TSI Aerometrics) before they are sent to a computer displaying the desired data.

With this set-up the droplet size is measured with the receiver. The transceiver measures the velocity in vertical and horizontal direction, both perpendicular to the direction of the laser beams. These droplet properties can be examined in the spray by measuring at different positions. The exact measurements are discussed in Subsection 2.2.4.

2.2.2 Experimental set-up

The experimental set-up is shown in Figure 2.7. It consists of a nozzle, a grounded metal target cylinder, the PDPA equipment and some other necessary equipment such as a liquid feed pump and a high voltage supply. Both Figure 2.7 (a) and (b) are complementary: the nozzle is drawn twice, the rest of the equipment only once. In Figure 2.8 the nozzle is shown in detail.

The EHDA spray is set up symmetrically with respect to the (x, z)-plane and the (y, z)-plane. The target cylinder, with a length of $0.265\ m$ and a diameter of $0.137\ m$, is placed right under the nozzle, at a distance of $20\ cm$ from the apex of the Gilbert-Taylor cone. Several MDF (Medium Density Fibreboard) boards are placed to shield different metal equipment, preventing the attraction of the charged droplets by that metal. The MDF walls are placed at even distance from the cylinder on both sides.



(a) Front view of the experimental set-up. (b) Side view of the experimental set-up.

Figure 2.7: Schematic views of the experimental set-up.

The PDPA equipment is schematically displayed in Figure 2.7 (b). The transceiver and receiver shown in Figure 2.6 are placed in front of and behind of the set-up in Figure 2.7 (a), respectively. The transceiver emits the four laser beams horizontally through the spray. The receiver is placed under an angle of 30° compared to the transceiver. Both instruments can be moved on a "rail" in the horizontal x-direction and also in the vertical z-direction, upwards and downwards the aluminium pillars. In this way the measuring point discussed in Subsection 2.2.1 can be placed in each point in the (x, z)-plane through the nozzle perpendicular to the direction of the laser beams.

The equipment that is not shown in the figures are the liquid feed pump (Harvard Apparatus PHD 2000 Infusion) and the high voltage supply (Heinzinger LCU 20000-05 pos). The liquid that has been sprayed is a 90.0 $wt\%$ over 10.0 $wt\%$ mixture of ethanol and triethylene glycol respectively. The liquid flow is held at $\phi = 7.00\ ml/h$. With the high voltage supply a voltage of 18.0 kV was put on the nozzle to get a cone-jet mode EHDA spray.

The experimental set-up must be as similar as possible to the model system set-up, to be discussed in Subsection 2.2.3. As will be clear from that subsection and as explained above, the set-up is symmetrical. A problem that occurs is that an experiment cannot be built perfectly symmetrically entirely, unlike a model.

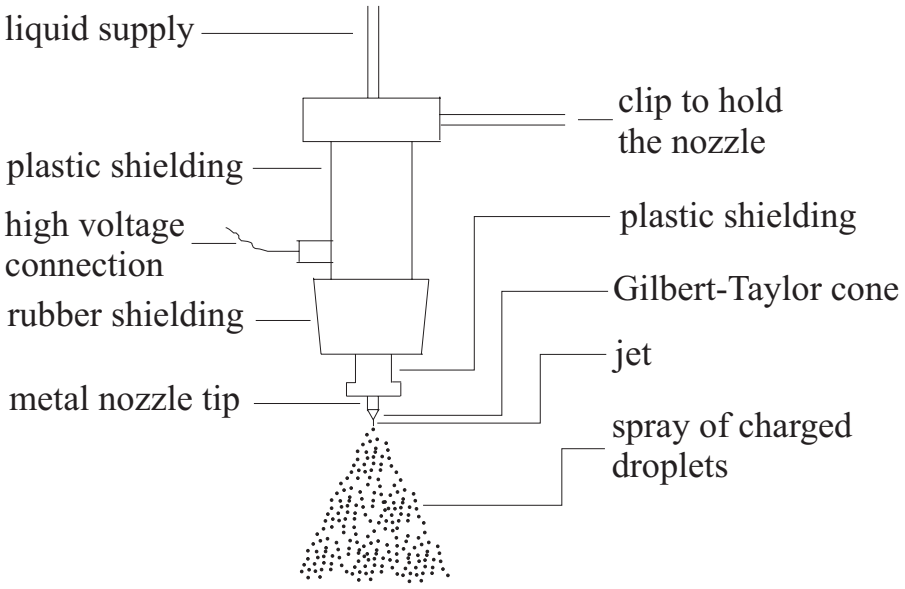


Figure 2.8: Experimental set-up of the nozzle.

As displayed in Figure 2.7 (a) the symmetry of the set-up looking from the front is good. What cannot be seen in this figure is that the left MDF wall is smaller than the right (in length) and that the middle of that wall is a few centimeters in front of the plane perpendicular to the wall through the nozzle. Furthermore the nozzle is hold at its place by a clip that is attached to a 'regulator table'. With this device the position of the nozzle can be changed by hand with very small distances. This device also brings non-symmetry into the set-up. From Figure 2.7 (b) the non-symmetry due to the PDPA equipment can be seen. But in real life the instruments are quite far away from the

spray so the influence is minimal, also because of the shielding of the aluminium with the MDF.

2.2.3 Model system set-up

Because not all experimental details can be and should be modelled, the boundary of the system is idealised. To identify a position in the system volume a grid has been made. The grid is shown in Figure 2.9. The figure shows only one quarter of the total volume. This is because of the symmetry with respect to the (x, z)-plane and the (y, z)-plane, mentioned above. The figure is produced in FEMLAB and will be discussed further in Section 3.4. The vertical axis is the z-axis, with decreasing value of z going downwards. The horizontal direction is the x-axis; left of the center of the cylinder x is negative, right of the center of the cylinder positive. The last direction is the y-axis, also horizontal but perpendicular to the cylinder. In the left top corner the metal tip of the nozzle together with the Gilbert-Taylor cone, shown in Figure 2.8, is recognized (pointed at with the arrow). The top of the metal nozzle tip is the origin of the grid $(x, y, z) = (0,0,0)$. The bottom of the Gilbert-Taylor cone is at $z =$ -0.00815 m. Underneath the nozzle the metal target cylinder is seen. The boundary walls in the (y, z)-plane are at the same distance of 0.2175 m from the nozzle as the MDF walls mentioned in the former subsection. The same distance is used for the other two boundary walls.

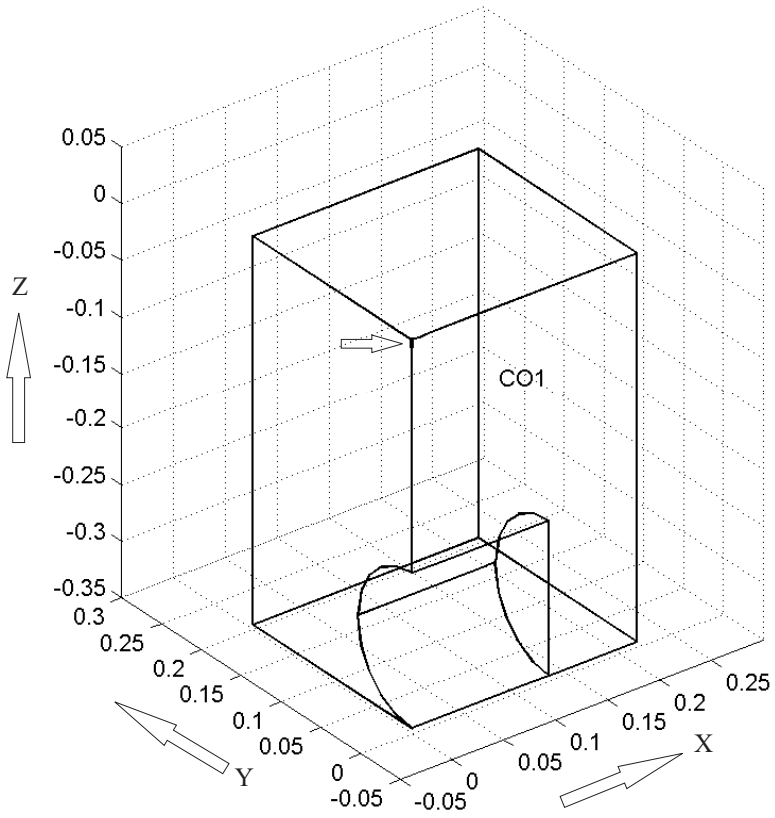


Figure 2.9: The grid edges of a quarter of the system volume.

2.2.4 Measurement procedure

Each measurement is started after adjustment of the equipment and is automatically stopped when the PDPA has analysed 10,000 droplets that passed through the measurement volume. The number of droplets counted for the velocity averaging in x-direction (with the green laser beams) is not the same as the number of droplets in z-direction (with the blue laser beams). The reason for this will be discussed now.

The detection boundary of the transceiver is different for the blue and the green laser beams. This is due to the different wave length and thus different energy of the laser light. The equipment is corrected for this. But this correction only works when the transceiver is exactly "lined out". This means that all four beams exactly cross at one point, which would be the measuring point. The error in the equipment causing the difference in the counted numbers of droplets, will be a combination of errors in the correction for the different laser energy and in the outlining of the transceiver.

Furthermore some droplets are judged to be 'wrong' by the analysis program, these droplets are neglected. For example when the droplets have a very improbable combination of size and velocity they are disposed. Therefore the found number of droplets is not 10,000 droplets but somewhat less (in the direction with the higher number of droplets).

The first measurement done in the (x,z)-plane is right under the nozzle, $x = 0\ m$, and at one centimeter distance above the target cylinder, z = -0.19815 m. The PDPA equipment, both the receiver and the transceiver, is shifted one centimeter upwards for every next measurement. In this way the average size and two velocities are measured at the different positions in the spray. This is proceeded until the end of the jet is reached where the droplets are produced. Then the series of measurements is repeated but for a different value of x, shifting all the PDPA equipment to the negative x-direction, see Figure 2.7 (a). Going to the left of the cylinder, for high enough z no droplets are measured anymore. When this position is reached the next series with the following x is started. This is proceeded until the PDPA hardly counted any droplets what made the measurement time too long and unreliable data are expected measuring less droplets than 10,000. The raw data from the measurements are shown in Appendix A.

Thus at the different positions in the spray the size, vertical and horizontal velocity and also the density of the droplets are measured. This density is the counted number of droplets per unit measuring time. In this way a 'scan' of the earlier mentioned (x,z)-plane through the nozzle is made. An image of droplet trajectories can be obtained combining the average velocity and density data.

2.3 Deposition analysis

The second type of verification measurements concerns a droplet deposition analysis. This treats the end of the spray area discussed in the introduction to this chapter. Two different experiments are done, namely the deposition on the target cylinder and on the walls and the ground.

2.3.1 Target cylinder

The deposition on the target metal cylinder has been investigated using a substance in the sprayed liquid that can be clearly seen when it is exposed to UV-light. Therefore a 12.0 g/l solution of the UV-light sensitive substance Tinopal in water was used. The new mixture that was sprayed consisted of 0.1 $wt\%$ of this Tinopal solution and 99.9 $wt\%$ of the original ethanol/triethylene glycol mixture.

Two light green colored papers were stuck together and folded tightly over the cylinder covering the surface of the cylinder. Also the sides of the cylinder were covered with this kind of paper. It was intended to spray using the same conditions as with the PDPA measurements as given in Subsection 2.2.2. But the high voltage of 18.0 kV seemed not enough to produce a stable spray. To gain this stable spray a higher voltage of 21.5 kV was put over the nozzle.

The cylinder was exposed to the spray over a period of five minutes. After that the paper was carefully removed from the cylinder and the deposition was observed exposing the paper to UV-light in 'Maria's Machine' (UVP Laboratory Products Epi Chemi II Darkroom).

2.3.2 Walls and ground

The deposition on the MDF wooden walls and ground, see Figure 2.7 (a), is looked at qualitatively with water-and-oil-sensitive paper (TeeJet, Spraying Systems Co.). It was tested at first that this paper is sensitive to the ethanol/triethylene glycol mixture used: the yellow paper became blue when the liquid came into contact with it.

Subsequently several paper strips of 2.6 cm width were placed horizontally over the whole length of the left wall. In front of and behind of the cylinder, and between the left wall and the cylinder the same kind of strips were put on the ground. Over a period of a few minutes the liquid was sprayed onto the cylinder. Afterwards the paper strips were removed and deposition of droplets was analysed looking at the change in color of the paper.

Model 3

This chapter treats the modelling part of the work done. In the first section some background information is given. This involves the history of the model development and a description of the final model. Finally many technical programming details are given. For those people who are not interested in this, the first section can be skipped. Section 3.2 and Section 3.3 treat the modelling equations, simplifications and assumptions. The way the external electric field is implemented in the model can be read in Section 3.4. The next section involves the initial properties of the produced droplets, needed to start the trajectory of each droplet. Finally the main part of the model is presented: a scale-up method that should allow to simulate the EHDA spray behaviour with a relative small number of droplets, while preserving the collective properties of the spray as much as possible. Consequently numerical simulations could be carried out within days instead of years. This valuable scale-up method was developed within the framework of the model with all its assumptions and for a given set of parameters.

3.1 Origin, implementation and result

The base for the model is the report of Pitchumani [13], in which the modelling of the trajectories of two ethylene glycol droplets in an electrospray has been described. In the three-dimensional model the Gilbert-Taylor cone is at rest and has the same high voltage as the metal nozzle tip. In Newton's equations of motion the forces acting on each droplet are taken into account, which are the external electrostatic force due to a potential difference between nozzle and target, the drag force and the Coulomb interaction forces between the droplets. The external electric field is obtained using a partial differential equation solver FlexPDE that solves the Laplace equation. Assuming that the electric field is rotation symmetric, it is calculated in two dimensions and the third dimension is taken into account by rotation symmetry. To get the droplet trajectories Newton's equations of motion for each droplet were integrated simultaneously in MATLAB (version 6.0 R12) using the initial positions and velocities of the droplets.

In this study Pitchumani's model was extended with more droplets, all starting at some initial position and moving during some time period towards a grounded target. Two-dimensional simulations with ten droplets took already about two days, so three-dimensional simulations were not an option anymore because of the too high calculation times. Despite of that the solutions were not accurate, they seemed promising and reflected the expected shape of a typical path of a droplet in an EHDA spray. The result of a MATLAB simulation with ten droplets is given in Figure 3.1.

On the basis of the experience of supervisor S. Luding the model was translated into

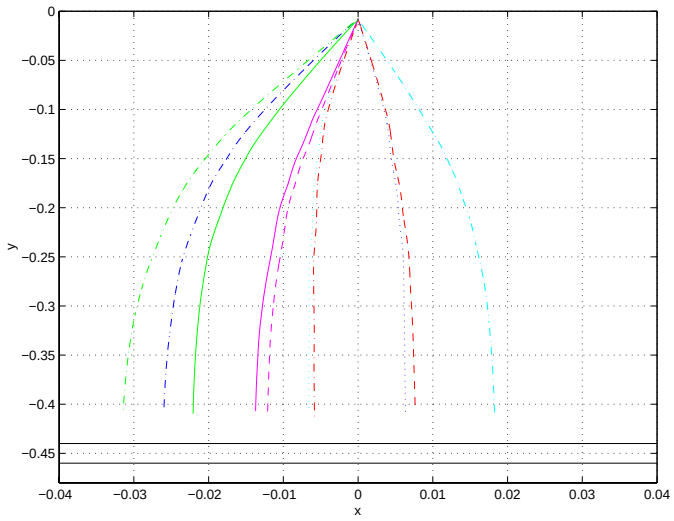


Figure 3.1: Result of an early MATLAB simulation with ten droplets.

the C++ language having advantages compared to the program in MATLAB. Some advantages are that the C++ model was more easy to adapt, significantly faster due to compilation and data were more easily created than in MATLAB. The model was extended with the gravitational force and several other features. Instead of FlexPDE the program FEMLAB is used to solve the three-dimensional external electric field. Furthermore, during the graduation period the model is improved more and more, while new information has been obtained.

The final model simulates an EHDA spray right after the jet break up. The experimental break up is treated in Subsection 2.1.1, and the modelling of the droplet properties right after break up is discussed in Section 3.5. The simulation starts with one droplet that just has been produced. Each numerical time step Newton's equations of motion are integrated and the droplet moves in the x, y, z-volume. After a certain production time step new droplets are produced, one after each time step, and as time elapses a spray is formed. Eventually the droplets will hit the metal target cylinder or maybe one of the boundary walls. When the charged droplets hit one of these objects they disappear instantaneously. This simulates that the charge of a droplet flows instantaneously to earth because the target is grounded. After some time the process has reached the steady state. The number of droplets produced in time is then (approximately) the same as the number of droplets deposited in time. When this state is reached the spray is constant and it can be compared to the experimental results.

In the C++ language some files are written that are subsequently compiled creating the executable program file that has to be run. In this way the program file `exvthlc.exe` is made by compiling the files `exvthlc.cc`, `exvthlc.h` and `exvhlc.h` with the command in the file `exvthlc.bat`. These files are presented in Appendix B. For more information on the program(ming) is referred to T. Winkels (author) and S. Luding (supervisor). Some more details will be discussed now.

The program has some input files to introduce necessary parameters. The file `par.ini` contains parameters concerning time and droplet, air and force properties. The initial

properties for the one initial droplet and the boundary wall parameters are given in `c2d.ini`. Some parameters concerning the linked cell structure that is used are listed in `lcell.ini`. Finally data of the external electric field grid points are read in from `efield.ini`. The production of this last file will be discussed in Section 3.4.

The model also has some output files. In the files `c2d` and `gnu` the properties, such as position and velocity, of each present droplet are stored. These files are updated after a chosen output time step. In the file `deposit` the properties of the droplets that are deposited at the target are stored. A similar file is `wall` that stores the properties of the droplets deposited at the boundary walls or the ground. The last output file is `comment` where bounces between droplets are stored and also the time and date when the simulation has been started and ended. The exact format of the input and output files is given in Appendix C.

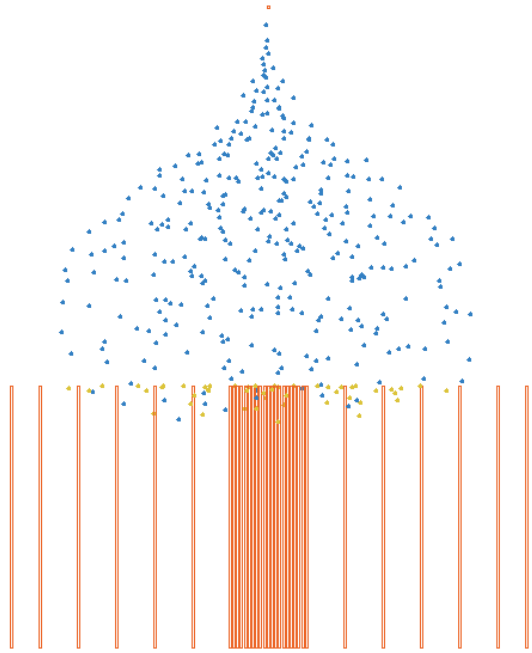


Figure 3.2: Snapshot of a simulated EHDA spray using the x-balls visualisation tool.

To view the simulated EHDA spray a C++ program, `exvc2dtot.exe`, is written to make another output file called `c2dtotal` combining the `c2d`, `deposit` and `wall` files. The program is shown in Appendix D. This file stores all properties of the droplets in air or deposited and consequently the spray can be viewed as a movie using the x-balls snapshot visualisation tool [11]. In Figure 3.2 such a snapshot of a simulated EHDA spray is presented. At the top the point of droplet production is seen (orange). The droplets (blue) travel through the air towards the target cylinder (orange). The deposited droplets stay on the target cylinder in the visualisation and become green. The visualisation can be rotated, creating a three-dimensional movie.

3.2 Equations in discrete formulation

To calculate the trajectories of the charged droplets in the EHDA spray, Newton's equations of motion has to be solved simultaneously for all droplets. For one droplet i with mass m and velocity $\vec{v}$ and at which some forces $\vec{F}$ are working, this reads:

$$\frac{d(m_i\vec{v}_i)}{dt} = \sum_i \vec{F}_i, \tag{3.1}$$

with $\vec{v}_i = \frac{d\vec{r}_i}{dt}$, $\vec{r}$ the position vector of the droplet and t time.

The forces acting on a droplet in the spray are:

- The electrostatic force due to the external electric field that is produced by the high voltage difference between the nozzle and the target,
- The Coulomb interactions between the charged droplets,
- The drag force due to the movement of the droplet in the ambient air,
- The gravitational force.

The first two forces are of electrical nature and therefore involve the charge of a droplet q. The maximum charge on a droplet is given by the Rayleigh limit. At this limit the surface tension of the droplet is just strong enough to compensate the repulsion due to the charges on the droplet, and keep the droplet intact. The charge on a droplet i at the Rayleigh limit is according to Vercoulen [15]:

$$q_{i,R} = 8\pi\sqrt{\varepsilon_0\gamma}\left(\frac{1}{2}d_i\right)^{3/2}. \tag{3.2}$$

Using EHDA the charge on a droplet will be less than the Rayleigh limit and thus some efficiency η_R has to be taken into account. For the cone-jet mode of an EHDA spray this efficiency η_R is set to 70 % according to Hartman [6]. This leads to a droplet charge of $q_i = \eta_R\, q_{i,R}$.

The force due to the external electric field is the multiplication of the charge of the droplet q_i with that electric field $\vec{E}$, which will be discussed later in Section 3.4. The Coulomb force requires the vectorial and absolute distances between all droplets i and j, $\vec{r}_{ij}$ and r_{ij}, because it is a droplet-droplet interaction force:

$$\vec{F}_{coulomb} = \sum_{i\neq j}^{N} \frac{q_i q_j \vec{r}_{ij}}{4\pi\varepsilon_0 r_{ij}^3}. \tag{3.3}$$

The third force acting on a droplet is the drag force describing the drag caused by the movement of ambient air along the droplet. The drag force is a form of Newton's resistance equation and given by:

$$\vec{F}_{drag} = C_D\frac{\pi}{8}\rho_{air}d_i^2(\vec{v}_{air} - \vec{v}_i)\,|\vec{v}_{air} - \vec{v}_i|\,, \tag{3.4}$$

with C_D the drag coefficient, ρ_{air} the air density and $\vec{v}_{air}$ the air velocity. The drag coefficient C_D is dependent on the Reynolds number Re of the droplet, which is

$$Re = \frac{\rho_{air} \left| \vec{v}_{air} - \vec{v}_i \right| d_i}{\eta_{air}}, \tag{3.5}$$

with η_{air} the air viscosity.

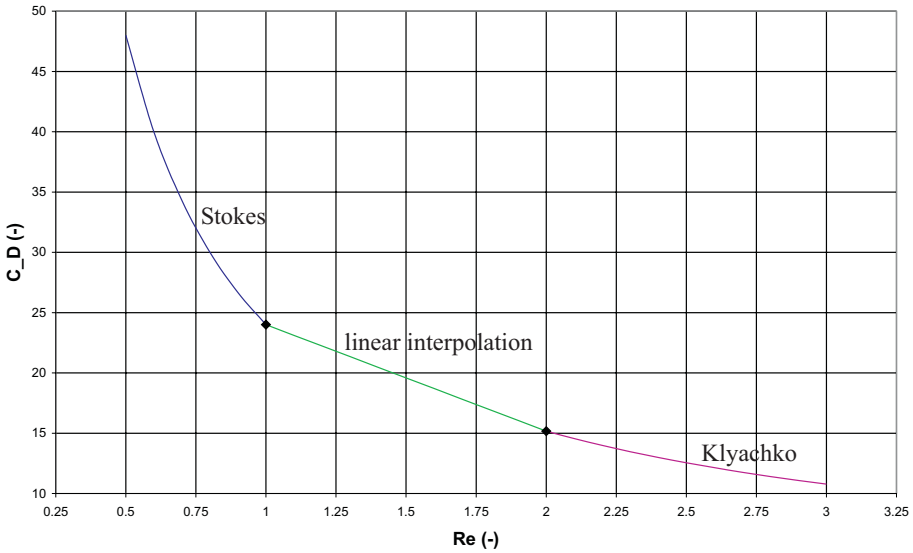


Figure 3.3: Drag coefficient for different regimes of the Reynolds number.

If a droplet is flowing in the (laminar) Stokes regime the Reynolds number is smaller than or equal to unity and the drag force coefficient is

$$C_D = \frac{24}{Re}. \tag{3.6}$$

If the Reynolds number is between 2 and 800 the drag coefficient is assumed according to Klyachko (referred is to Hinds [8] that uses the Klyachko equation):

$$C_D = \frac{24}{Re}\left(1 + \frac{Re^{2/3}}{6}\right). \tag{3.7}$$

In the intermediate region of the Reynolds number between 1 and 2 no generally used equation is known. Therefore a linear function between the drag coefficient of Stokes for $Re = 1$ and the drag coefficient of Klyachko for $Re = 2$ is used. The different regimes can be seen in Figure 3.3.

The last force that acts on a droplet is due to gravity and is obviously the mass of the droplet m_i times the gravitation constant $\vec{g}$.

The four forces lead to the following form of Newton's equations of motion for each of the N droplets i in each direction of the D dimensions (in this case x, y, z):

$$\frac{d(m_i \vec{v}_i)}{dt} = q_i \vec{E} + \sum_{i \neq j}^{N} \frac{q_i q_j \vec{r}_{ij}}{4\pi\varepsilon_0 r_{ij}^3} + C_D \frac{\pi}{8} \rho_{air} d_i^2 (\vec{v}_{air} - \vec{v}_i) \left| \vec{v}_{air} - \vec{v}_i \right| + m_i \vec{g}. \tag{3.8}$$

This is a coupled system of $N \cdot D$ differential equations where the Coulomb forces are responsible for the coupling. To solve the system clearly some initial properties for each droplet are necessary. This means that an initial diameter, position and velocity of the droplet should be introduced. This will be discussed further in Section 3.5. The integration of the diferential equations is done with the Verlet method. For more information on the solving of differential equations for chemical physics is referred to Reference [1].

3.3 Assumptions

Because it is in any case impossible to precisely imitate every detail of the reality in a model, inherent to modelling is making assumptions. In the former section some assumptions are already made. The main assumptions of the model are:

- Monodispersity of the droplets.
- No evaporation of the liquid droplets takes place.
- The charge on a droplet is constant.
- All other physical properties of the liquid droplets are constant during the process.
- The dependency of the drag coefficient on the Reynolds number is reflected by Figure 3.3, Equations (3.6) and (3.7).
- The ambient air has constant properties and is at rest, $\vec{v}_{air} = \vec{0}$.
- The droplets are treated as individual charge carriers that move in a constant external electric field. Therefore the approach that an electrospray has a space charge is not followed.
- No other forces than given in Equation (3.8) are acting on a droplet.

Giving up any of the assumptions is not the goal of this study, but requires further research. However, they are discussed in some detail below.

It is known that the droplet size distribution of an EHDA spray is not monodisperse but polydisperse with a narrow size distribution [4]. On the other hand, the first assumption is done because in the ideal case an EHDA spray will have a monodisperse droplet size distribution. Furthermore modelling a spray with a polydisperse size distribution would be already too difficult in this stage of the research. This distribution would also have to be known for the polydisperse spray.

Naturally some evaporation of the liquid droplets will take place, see References [3] and [12]. Again to hold the size of the droplets constant as a simplification for the model this process is neglected. The first two assumptions thus ensure a constant droplet size.

The third assumption says no charge will leave the droplet. This is realistic even when evaporation takes place, as long as the permittivity of the ambient air is low enough and no other transport processes take place.

During the movement the droplets will change in size and composition because one component will evaporate faster than the other. The influence of this is neglected and

the properties of the liquid droplets are maintained at their initial properties. In Table 3.1 the used properties of the ethanol/triethylene glycol droplets and air are given.

As explained in the former section two equations for the drag coefficient C_D are found in literature. The range of validity of these equations do not cover the whole range of the Reynolds number of the travelling droplets. Namely a gap occurs between the Reynolds number of one and two. As presented in Figure 3.3 the relation in this range is assumed to be linear. From this figure the error made by this interpolation is believed to be small.

Table 3.1: Droplet and air properties.

quantity	symbol (units)	value
droplet density	ρ (kg/m^3)	823
droplet diameter	d (μm)	8
droplet surface tension	γ (mN/m)	23
droplet conductivity	K $(\mu S/m)$	100
air density	ρ_{air} (kg/m^3)	1.2
air viscosity	η_{air} (Pas)	$1.81\ 10^{-5}$
air velocity	$\vec{v}_{air}$ (m/s)	0

A major assumption is made according to the ambient air. The assumption of constant properties may be quite realistic, but the air will not be at rest [5]. Because of the movement of the droplets in air, initially at high velocity (and turbulent), the air between the droplets will move too. It is unknown how the air will flow and with what velocity. Difficult flow pattern simulations have to be done simultaneously with this model to calculate the influence of the air flow on the droplet trajectories. This is not taken into account and air is thus assumed to be at rest.

In the model the droplets are the objects that are tracked in time and are the charge carriers itself. These charge carriers move through the electric field as individual objects interacting with other objects, the other droplets in the spray. Using this approach, the spray is not looked at as a volume with a space charge (in C/m^3). Furthermore, the external electric field is assumed to be constant.

In chemical physics many types of forces exist. The forces acting on the droplets in the EHDA spray are given in Equation (3.8). Some other forces might be included. It is explained here why they are not included.

When the droplets move through the air they undergo the so-called Brownian motion. This is the irregular wiggling motion (a kind of oscillation) of an aerosol particle in still air caused by random variations in the bombardment of gas molecules against the particle. Two characteristic times, τ_1 and τ_2 given in Equations (3.9) and (3.10), determine the influence of the Brownian motion on the displacement of the droplet. The τ_1 is the damping time of the oscillation. The τ_2 is the time the droplet needs to travel a distance in the order of the droplet diameter. The influence of the Brownian motion on the droplet trajectory is then negligible if $\tau_1 << \tau_2$.

$$\tau_1 = \frac{\rho d_i^2}{18\eta_{air}} \tag{3.9}$$

$$\tau_2 = \frac{3\pi\eta_{air} d_i^3}{2kT} \tag{3.10}$$

For the quantities already mentioned and Boltzmann constant k and ambient temperature T, estimated at 293 K, this leads to: $\tau_1 = 1.6\ 10^{-4} s << \tau_2 = 11\ s$. Estimated from Reference [10] the travelled distance of a droplet due to Brownian motion is in the range of a few μm. The Brownian motion is thus negligible.

When the complete theory of electromagnetism is inserted Maxwell's equations should be calculated at each timestep. When the charges in the system are fixed or move as a steady flow these equations can be simplified from electromagnetism to electrostatics. In this case the external electric field, which is present due to the high voltage difference between the nozzle and the target, is constant in time. The spray of moving charged droplets causes a space charge. But by only looking at the steady state of the spraying system, the demand of electrostatics is answered. This excludes the magnetic properties of the system and no magnetic force is included in the model.

In literature on electromagnetics the so-called image force is mentioned. This is a force acting on a charged particle that is near a conducting surface. An imaginary opposite mirror charge will attract the particle towards the surface. In literature such a force is discussed only for one particle. Castle [2] includes this force in his list of forces acting on a charged droplet in a spray. He also mentions the space charge induced force, which is similar to the image force but concerning the whole spray as one charge. Vauge [14] acknowledges the existence of an image force for a single droplet, but denies the existence of an image force in case of a spray with many droplets. He states their will be no overall image force if a high concentration of charged droplets is present, which is the case in an EHDA spray, because this would lead to a conflict with the Gauss theorem. His conclusions hold for a spherical and cylindrical geometry of the target. It is not sure whether an image force has to be taken into account for this work. The existence of it is not generally accepted as stated above. For this work the image force, if it does exist, is neglected.

3.4 External electric field

From former sections every quantity in Newton's equation of motion, Equation (3.8), is known now except for the external electric field $\vec{E}$. At every position where a droplet is present this three-dimensional electric field should be known. Instead of using FlexPDE like Pitchumani, the external electric field is calculated with the program FEMLAB (version 2.2) that runs within MATLAB.

In FEMLAB the desired simulation volume is drawn as seen already in Figure 2.9. Subsequently some boundary conditions and parameters must be filled in to solve the system for the electric field. The metal nozzle including the Gilbert-Taylor cone is set to 18.0 kV. The metal target cylinder is set to 0 V (grounded). All other boundaries are set to a so-called 'insulation' with zero surface charge. The permittivity of the volume is

set to that of the ambient air (the same as for vacuum) and no space charge is put into the volume. To solve the electric field FEMLAB divides the volume into many mesh grid volumes. Those mesh grids are 'Delauney' tetraeders. The density of the mesh grids can be locally adapted. The used mesh grid is shown in Figure 3.4.

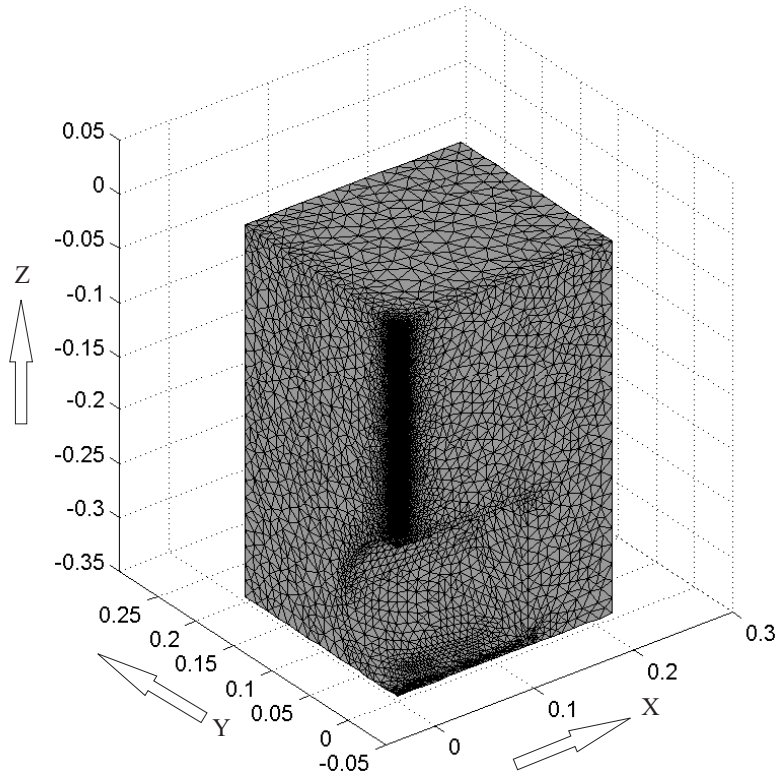


Figure 3.4: The simulation volume divided in Delauney tetraeders.

The density of the grid on the z-axis has been set very dense. This is done to assure a more exact interpolation between the grid points, needed because all droplets start (almost) on the z-axis and because the droplets that move close to this axis (x and y close to zero) should not experience a high artificial electric field forces in the x- and y-direction due to interpolation errors. Also close to the metal cylinder the grid is refined somewhat for a more exact deposition. The electric potential, related to the electric field, calculated with FEMLAB is shown in the Figure 3.5.

From the figures made in FEMLAB it is already clear that in FEMLAB only one quarter of the total volume is evaluated to produce the desired electric field. This is done to produce a symmetric electric field with respect to the (x, z)-plane and the (y, z)-plane, by mirroring the electric field of the modelled quarter of the volume. When the FEMLAB solution is calculated, the electric field in x-, y- and z-direction is calculated at every corner of a Delauney grid tetraeder. The MATLAB script for that evaluation (`cornerevaluation.m`) is given in Appendix E. Because a corner of such a tetraeder is part of several tetraeders the electric field evaluation with `cornerevaluation.m` produces data that occur several times. To exclude these 'double data' the C++ program `makegrid.exe` is written, which is shown in Appendix D. The electric field components are averaged for double data points. Subsequently the created electric field data are mir-

rored with a second MATLAB script (`mirror.m`) which is also given in Appendix E. Data points at the z-axis and in the symmetry planes do not change position due to the mirroring and as a result occur several times again. Therefore the program `makegrid.exe` has to be run again to exclude the 'double grid points'. The resulting data file with the three-dimensional components of the electric field is `efield.ini` used in the program.

As mentioned before the electric field components must be known at the place where a droplet occurs to calculate the electric field force. Therefore an interpolation algorithm is set up to interpolate the electric field between the grid points retrieved from FEMLAB. The interpolation algorithm is discussed in more detail in Appendix F.

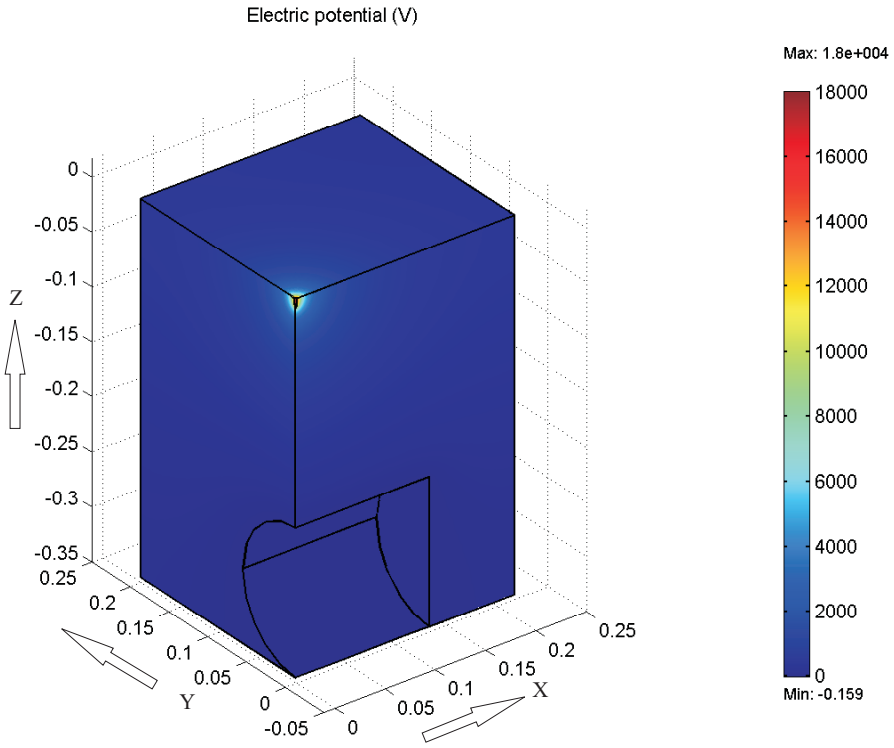


Figure 3.5: The resulting electric potential field.

3.5 Initial droplet properties

The advantage of the three-dimensional model is that the system only needs some initial conditions and from there on the system is tracked in time deterministically. Concretely this means that the droplets that are produced need some initial size, position and velocity. After that the droplets are influenced by forces and move through the system volume according to Newton's equations of motion.

As appears from the used approach the initial properties of a droplet are important and necessary to start the droplet trajectory. Thus for every droplet that is produced the initial s must be close to reality. This is a problem because the production of droplets

due to the jet break up, described in Subsection 2.1.1, is not exactly known because of its complexity. Hartman [4] and [6], investigated this complicated process for his set-up and provided some valuable information, but still further assumptions have to be made to model the production of droplets.

Table 3.2: Dimensions of the nozzle, Gilbert-Taylor cone, jet and target cylinder in the model.

	nozzle	Gilbert-Taylor cone	jet	target cylinder
radius (mm)	1.25	1.25	not modelled	68.5
length (mm)	6.0	2.15	1.0	265

What should be stressed again is that in the model the Gilbert-Taylor cone is one piece with the metal nozzle tip and at the same high voltage. The emerging jet has a constant length and does not influence the electric field. The Gilbert-Taylor cone is thus physically put into the model as a boundary, whereas the jet is not. But the jet is taken into account for droplet production as will be clear later.

The dimensions of the modelled nozzle, Gilbert-cone, jet and target cylinder are presented in Table 3.2. The dimensions of the nozzle and the Gilbert-Taylor cone are taken from Pitchumani [13] because he used the same nozzle type. The dimensions were checked and appeared to be in fact correct. The dimension of the jet was estimated looking at it with spotlights.

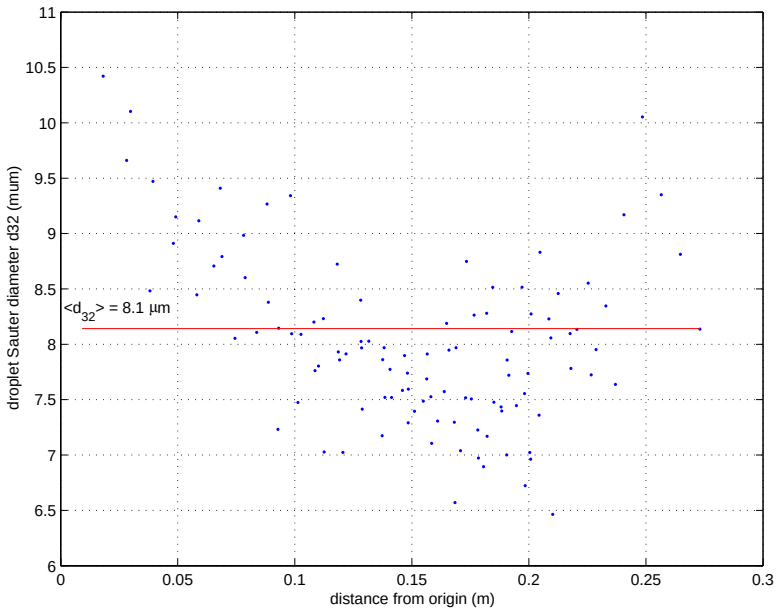


Figure 3.6: Droplet diameter versus distance from the grid origin for all PDPA measurements.

It is measured with the PDPA that the droplet size distribution using EHDA is quite narrow, but not monodisperse as modelled. From the number size distributions

the Sauter mean diameter, defined as $d_{32} = \frac{\sum_i N_i d_i^3}{\sum_i N_i d_i^2}$, is obtained as a representative diameter for the distribution. This quantity is somewhat higher than the mode droplet diameter of the number size distribution due its definition. The narrower the number size distribution, the less the difference between that mode droplet diameter d and the d_{32}. Because the number size distribution is quite narrow, the difference of the Sauter mean and the mode diameter is small.

The average droplet Sauter mean diameter of all PDPA measurements is 8.1 μm, shown in Figure 3.6 with the horizontal line. The droplet size at the droplet production point is $d_{32} = 22$ μm. It appears that close to the nozzle the droplets evaporate very quickly and after that the evaporation rate is much less. The droplet size at the droplet production point is not shown in the figure to give a better overview of the measured droplet sizes. As mentioned in Section 3.3 the size of the droplets in the model is constant. The diameter of the produced droplets is set to 8 μm, according to the average Sauter mean diameter from the PDPA measurements. The size or size distribution of the droplets is not given more attention because of the assumption of monodispersity.

Hartman investigated the oscillation of the jet and measured the initial position (with respect to the earlier mentioned grid) of a droplet. He found that the initial radial position $r_{initial}$, here in x- and y-direction, is less than the jet radius. Furthermore simulation results appeared to be insensitive to the value of the initial position if it is small compared to the jet radius. The exact initial position is therefore not needed but it must be less than the jet radius. To comply with this findings the initial radial position $r_{initial}$ is set to a maximum of 1.0 μm. The distribution of $r_{initial}$ is random. Finally, in the model the droplets are produced at the same axial distance z, namely at the end of the jet at z = -0.00915 meter.

Next to the size and position the third important property of the produced droplets is their initial velocity. Hartman [5] also investigated the initial radial velocity and concluded it is zero for his experiments. Experimental measurements with the PDPA (Appendix A) show that the droplets do have a significant initial radial velocity. This can be explained by the fact that the droplets are accelerated enormously right after their origination and that the position of the PDPA measurements are hard to establish accurately. This can have a large influence on the results, especially close to or at the droplet production point. Hartman's conclusion is possibly right for every EHDA spray but this is not validated for other work than his. Furthermore, the electric field, which is high close to the nozzle, determines greatly the initial radial acceleration and therefore the velocity right after the production of a droplet. This field is very different from the one in Hartman's experiments: not only the magnitude of the electric field is different but also the experimental set-up is very different. In this model Hartman's conclusion is not taken into account and the produced droplets have a non-zero initial radial velocity.

As mentioned before the jet break up is a complex process and not exactly known. Possibly this process is dependent on many parameters that differ from one experiment to the next. In this model the production of droplets from the jet is assumed to be random with a constant total initial (three-dimensional) velocity $|\vec{v}_{tot}|$. This will be explained now in more detail.

In Figure 3.7 a total velocity vector $\vec{v}_{tot}$ in the direction (x, y, z) is drawn with its

components, v_x, v_y, and v_z, and its projections. In the model two angles are defined which are randomly produced for each new droplet: α_1 and α_2. The first angle α_1 is the angle between the total velocity vector and the z-velocity vector, determining the axial direction of the total velocity vector. This angle has a random value in the range $[-\alpha_{1,max}; \alpha_{1,max}]$. The consequence of the random production of α_1 in a certain range is that the simulated spray has a sharp boundary: the edge of the spray is easily detectable. The second angle α_2 is the angle between the projection of the total velocity vector on the (x, y)-plane and the y-velocity vector, determining the radial direction of the total velocity vector. This direction is identical to that of the initial position vector. For symmetry reasons this angle has a random value between 0° and 360°.

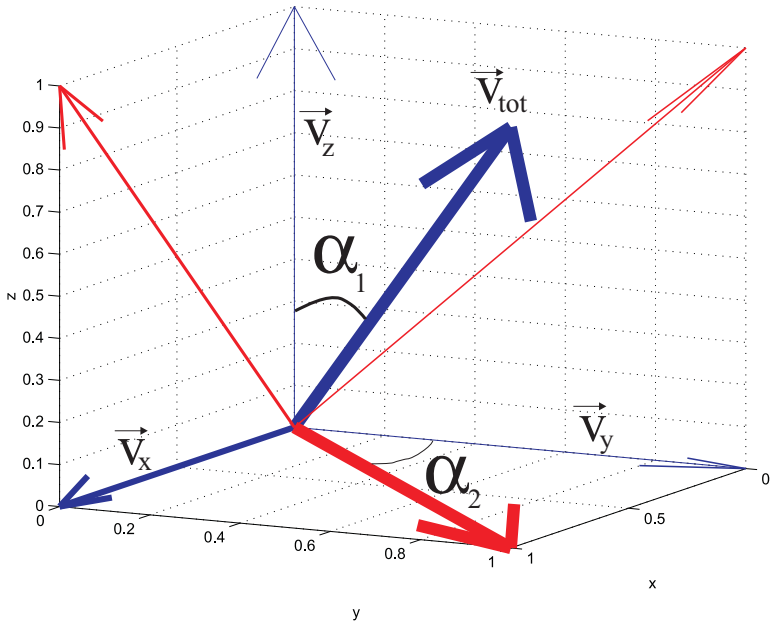


Figure 3.7: Example of a total velocity vector and its components and projections.

Both the $|\alpha_{1,max}|$ and the $|\vec{v}_{tot}|$ are arbitrarily chosen. As will be discussed later it was attempted to make the simulated spray more broad by increasing the value of $|\alpha_{1,max}|$. Therefore a relatively large angle of 40° is chosen. By choosing such a large $|\alpha_{1,max}|$ the spread in the initial v_x, v_y and v_z becomes higher. From the PDPA measurements and knowledge gained from experts in the Particle Technology Group (K.B. Geerse and J.C.M. Marijnissen) a $|\vec{v}_{tot}| = 17\ m/s$ is chosen.

Table 3.3 concludes and summarizes the initial positions (x, y, z) and initial velocities (v_x, v_y, v_z) for produced droplets. The initial displacement in x- and y-direction is determined by α_2. The direction of the initial x- and y-velocities is the same as that of the displacement, also determined by α_2. Finally α_1 determines the three-dimensional direction of the initial velocity. When all these initial properties and the initial size are given, Newton's equations of motion, Equation (3.8), have to be solved for all droplets simultaneously, what leads to the trajectories of the droplets in the EHDA spray.

Table 3.3: Initial properties for produced droplets depending on $r_{initial}$, $|\vec{v}_{tot}|$, α_1 and α_2.

position	
x (μm)	$r_{initial}$ $\sin(\alpha_2)$
y (μm)	$r_{initial}$ $\cos(\alpha_2)$
z (mm)	-0.00915
velocity	
v_x (m/s)	$\|\vec{v}_{tot}\|$ $\sin(\alpha_1)$ $\sin(\alpha_2)$
v_y (m/s)	$\|\vec{v}_{tot}\|$ $\sin(\alpha_1)$ $\cos(\alpha_2)$
v_z (m/s)	$\|\vec{v}_{tot}\|$ $\cos(\alpha_1)$
random initial radial position	
$r_{initial}$ (μm)	-1.0 - +1.0
constant total velocity	
$\|\vec{v}_{tot}\|$ (m/s)	17
random angles	
α_1 $(^\circ)$	-40 - +40
α_2 $(^\circ)$	0 - 360

3.6 Scale-up method

From the former sections the mathematical equations describing the trajectories of the charged droplets of the EHDA spray are clear. The problem of simulating the EHDA spray is that it consists of many droplets. Each second a number in the order of 10^5 - 10^6 droplets are produced. Contemporary computing power cannot handle this in reasonable time. The second, related, problem is that the trajectories of the droplets depend strongly on the number of droplets produced. These problems are tackled using a scale-up method in the model.

It is assumed that only the Coulomb forces in Equation (3.8) are affected by the number of droplets due to the interaction terms. To simulate the real process with much less droplets, the charge of the droplets is increased only for the Coulomb interactions. The increase in charge is not taken into account concerning the electric field force, because an undesired strong change of the droplet behaviour would occur. The charge increase is done by multiplying the real droplet charge with a certain factor, the Coulomb charge factor f_q. With this scale-up method each simulated droplet represents a cluster of real droplets with respect to the Coulomb charge, but keeps its behaviour as a single droplet with respect to the electric field, the drag and gravity.

As a consequence of this method the space charge effect will be different in the simulations and in real life due to the different distribution of charge in space. The goal of the modelling method is to scale-up the simulation in a way such that the real

experimental spray can be simulated with a relative small number of droplets having a relative high Coulomb charge. This implies that the collective properties of the spray should be as much as possible preserved.

Various simulations are done to check the scale-up behaviour of the model. The droplet production time t_{prod} and the Coulomb charge factor f_q are varied in these simulations. In Table 3.4 the used production times and accompanying production frequences are given. In Table 3.5 the Coulomb-charge factors with accompanying real charges on the droplets are given. From these tables it is clear that $5 \cdot 7 = 35$ scale-up simulations are done. In Section 3.5 the initial properties of produced droplets were discussed. For the scale-up simulations a $|\vec{v}_{tot}|$ of 15 m/s, an $|\alpha_{1,max}|$ of 10° and a droplet diameter of 15 μm were used. These values are different than for the final model because of the new information that was obtained meanwhile. All simulations are done for a simulation duration of one second.

Table 3.4: Droplet production times and accompanying frequencies of the up-scale simulations.

production time (s)	0.02	0.01	0.005	0.0025	0.00125
production frequency (Hz)	50	100	200	400	800

Table 3.5: Droplet Coulomb charge factors and accompanying real charges of the scale-up simulations.

Coulomb charge factor (-)	0	0.25	0.5	1	2	4	8
real droplet charge (C)	0	$5.6\ 10^{-15}$	$1.1\ 10^{-14}$	$2.2\ 10^{-14}$	$4.5\ 10^{-14}$	$9.0\ 10^{-14}$	$1.8\ 10^{-13}$

The simulations are analysed with respect to their droplet deposition pattern on the target cylinder, using the MATLAB file `evaluate_deposition.m` in Appendix E. The average distance of droplet deposition from the center point of the cylinder (right under the nozzle) $< r_{dep} >$, is taken as the quantity to analyse. This is an arbitrary choice since other quantities like the maximum distance or the averaged squared distance of droplet deposition lead to similar results.

The simulation results are shown in Figure 3.8. A mathematical analysis is done, stating that the average distance of droplet deposition $< r_{dep} >$ is a function f of the production time and the Coulomb charge factor. This is represented by

$$< r_{dep} >= f(t_{prod}, f_q). \tag{3.11}$$

The $< r_{dep} >$ of the simulations with a lower droplet Coulomb charge than the real charge, $f_q < 1$, and a low t_{prod}, will be influenced significantly by that of the simulation with $f_q = 0$. In this latter simulation the droplets have no Coulomb charge but have some $< r_{dep} >$, as seen in Figure 3.8, due to their initial nonzero α_1 (and $|v_{tot}|$) causing a spread of the spray. This influence is different for different parameter sets of $|\alpha_{1,max}|$ and

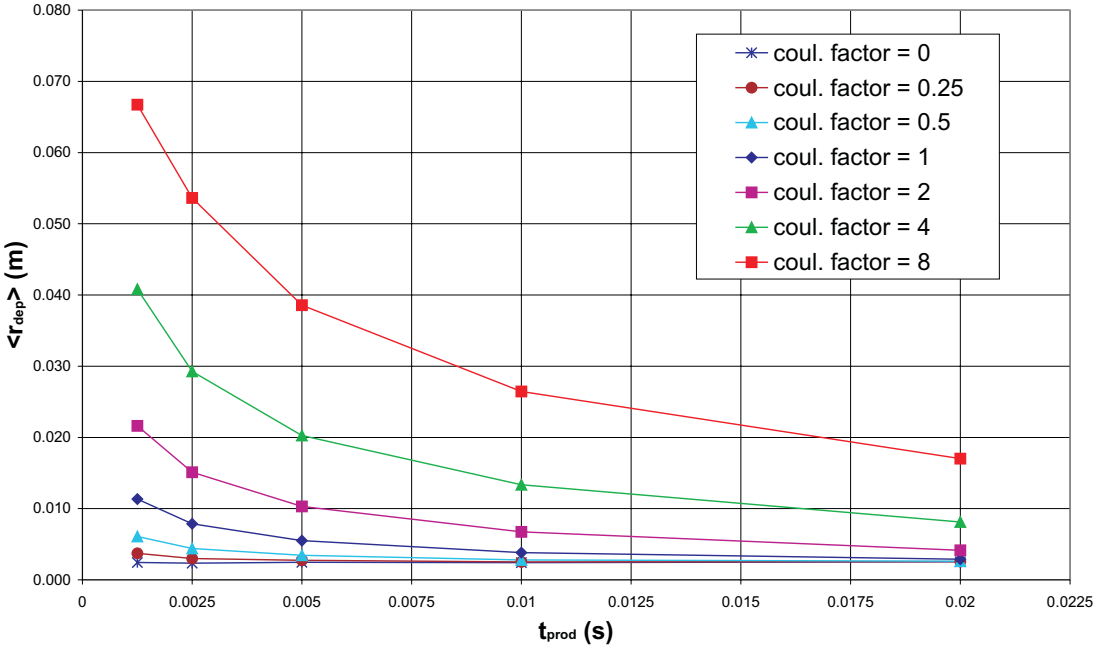
$|v_{tot}|$. To create a scale-up correlation that could be used for different parameter sets, the $< r_{dep} >$ of the simulation with $f_q = 0$ is subtracted from those of the simulations with $f_q \neq 0$. The developed scale-up correlation between the t_{prod} and f_q of two different simulations is:

$$\frac{f_q(sim_1)}{f_q(sim_2)} = \left(\frac{t_{prod}(sim_1)}{t_{prod}(sim_2)} \right)^{\left(\frac{0.57}{1.06}\right)} . \tag{3.12}$$

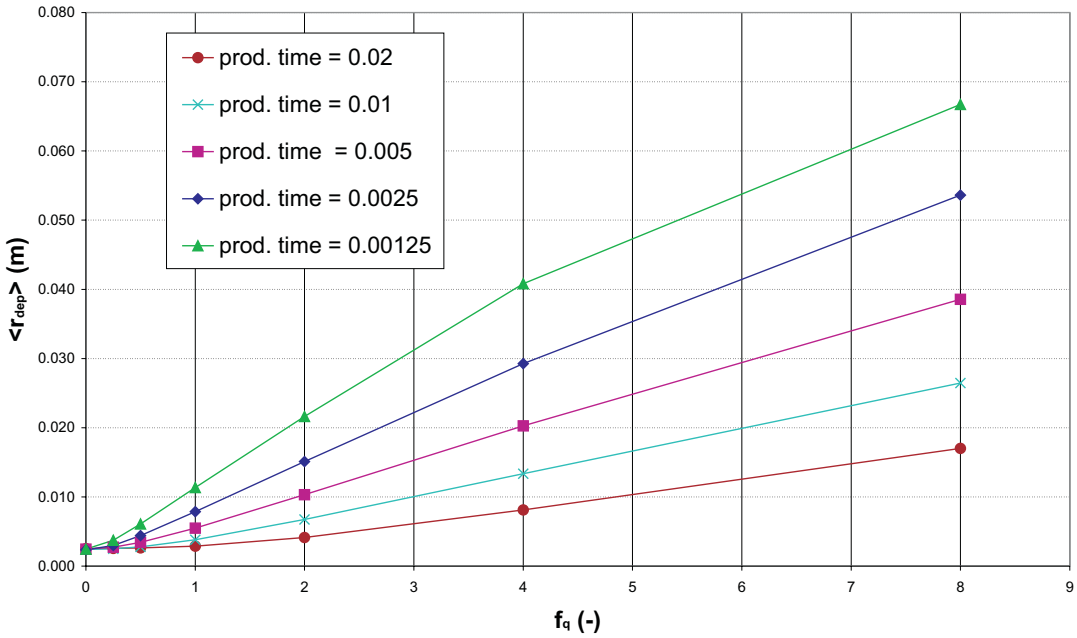
Two simulations with a t_{prod} and f_q that comply with this equation correspond to equivalent simulations and will have the same average distance of droplet deposition from the center point of the cylinder $< r_{dep} >$. This means that a particular simulation with less droplets and higher Coulomb charge will have the same result as the accompanying simulation with more droplets and lower Coulomb charge. But the former will have a shorter calculation time because of the fewer droplets.

As mentioned earlier the scale-up correlation was developed within the framework of the model with all its assumptions and for a given set of parameters. The validity of the correlation is checked for some different parameters, namely the final set mentioned in Section 3.5. Five validation simulations are done for the same values of t_{prod} as in Table 3.4. The results are discussed in Subsection 4.2.1.

Validation of the correlation can also be checked by extrapolation towards the experimental spray with many droplets. The Coulomb charge factor for a real spray is of course $f_q = 1$. According to Hartman [6] the experimental droplet production time can be calculated with Equation (2.2). With these quantities and Equation (3.12), the Coulomb charge factor for a simulation with chosen droplet production time is calculated, and the simulation can be compared to the experimental results. The final simulations are done with a droplet production time $t_{prod} = 1.25\ 10^{-3}\ s$. The results of the validation checks and verification are presented in the next chapter.



(a) Average distance of droplet deposition versus droplet production time.



(b) Average distance of droplet deposition versus Coulomb charge factor.

Figure 3.8: Analysis of average distance of droplet deposition for the up-scale simulations.

Results and discussion 4

In Chapter 2 it is explained that with the PDPA measurements at the different positions in the spray an image of droplet trajectories can be obtained by combining the average velocity and density data. In Chapter 3 the model is discussed that allows the EHDA spray to be simulated and data, similar to the experimental data, to be produced. The results of the experiments and simulations are presented in this chapter. In the first section the results of the PDPA and the deposition analyses are given. The second section involves the results of the simulations. First the validity of the scale-up correlation is checked. Subsequently the results of the final simulations of the EHDA spray are presented. After presenting the experimental and simulation results, the verification of the model is presented in the next section by comparing the experimental and the simulation results. Finally, the discrepancies between the model and the reality are discussed.

4.1 Experiment

In this section the results of the experiments are presented. The first subsection discusses the velocity and density profiles obtained with the PDPA measurements that are described in Subsection 2.2.4. The next section treats the results of the droplet deposition on the target cylinder and the system volume boundaries.

4.1.1 PDPA analysis results

Velocity

The velocity vector plot in the earlier discussed (x, z)-plane is presented in Figure 4.2. Each arrow represents the velocity vector in the (x, z)-plane. The large velocity vector in the upper right corner is the result of a measurement at the point of droplet production. It is concluded impossible to place the measuring volume of the PDPA set-up exactly at the production point of the droplets and thus measure the initial droplet velocities. Therefore several assumptions in Section 3.5 are done. As a result the velocity measurements at this point are considered unreliable and thus have a great error. It should be noticed that also some other measurements do not follow the trend of the rest of the data. It is checked that these results have a large error too, but are not disposed of.

The PDPA velocity profiles are plotted in Figure 4.3. This shows the velocity in x- and z-direction versus the position in those directions. *Note the sign of the quantities! These signs are consequently used in the report because of the model system set-up discussed in Subsection 2.2.3. To prevent any confusion the discussion of the results is done ignoring*

the negative sign of v_x and v_z, since a simple flip of sign allows a more intuitive discussion of the velocities. Furthermore some trend lines for measurements at constant x or z are implemented in the figures. Except involving the measurements with a large error, the trend lines with increasing $|x|$ or $|z|$ do not cross each other. Therefore the series at constant x or z that are not shown can be imagined in a similar way, following the dots in the figures.

In Figure 4.3 (a) v_x is plotted versus x. The trend lines represent measurements at constant z. From these lines it appears that at constant z the v_x increases with increasing $|x|$ (more to the end of the target cylinder). Furthermore at constant x, the v_x decreases with decreasing z. Finally close to the target cylinder, the v_x is almost constant for every x. What also can be seen is that at the side of the cylinder the sign of the v_x turns due to the attraction to the target cylinder, also seen in Figure 4.2.

Figure 4.3 (b) shows v_x versus z. Again trend lines are inserted but now with constant x. The measurements at $x = 0$ m clearly are distinct from the other series, showing a kind of zero velocity line. This is expected due to the symmetry of the EHDA spray. The measurement at the production area of the droplets does not follow this trend. But as it is very hard to measure exactly at that point, this measurement is unreliable. For higher $|x|$-values than zero the trends show a decrease of v_x with decreasing z. At lower $|x|$ the decrease of v_x with decreasing z is fast. With increasing $|x|$ this trend in the decrease is less steep. Again the points at the side of the cylinder with x = -0.15 m are recognized.

From Figure 4.3 (c) and (d) it results that v_z decreases with increasing $|x|$ at constant z. The measurement closest to the nozzle has a very high v_z. Although it is very hard to measure exactly at the point where the droplets are produced, the value of v_z resembles the expectation of experts on EHDA spraying (K.B. Geerse and J.C.M. Marijnissen).

Figure 4.3 (e) and (f) show v_z versus z. The trend lines are from measurements at constant x. The figure shows that the v_z decreases with decreasing z at constant x. But close to the target cylinder, v_z increases again. As clear from the other figures the measurement series at x = -0.15 m shows a different trend.

Density

What cannot be seen from Figure 4.2 and Figure 4.3 is the density of the droplets at each measuring point. With the obtained data (Appendix A) a quantity related to the density is calculated: the number of droplets measured divided by the measuring time (this quantity is called the 'density' in this report). As explained in Subsection 2.2.4 the number of droplets measured in x- and z-direction is different and the largest number of droplets is used to calculate the density for each measurement.

In Figure 4.4 (a) the density is plotted versus x in a semi logarithmic plot. This figure shows some measurement series with constant z. It appears that the density of the droplets decreases with increasing $|x|$-value at a constant z-value (going to the left in Figure 4.2 at one height). This is expected mainly because the high number of droplets go through a larger volume at lower z, resulting in a lower density. A quite sharp drop in the density can be seen between the measurements with -0.07 $m \leq x <$ -0.09 m (note the logarithmic vertical axis).

A similar plot is shown in Figure 4.4 (b), but the density is plotted versus z for constant x. The trend lines at constant x show that density decreases with decreasing z. With increasing $|x|$ the density starts lower as seen in Figure 4.4 (a). In Figure 4.4 (b) a horizontal line is drawn. Below this line the density of the measurements is signifcantly less than that of the rest. All points below this line are of measurements with a x-value less than -0.07 m, except for the (x, z) combinations of (-0.09,-0.19815) and (-0.09,-0.18815). Also from Figure 4.4 (a) can be seen that these two points have a density that is significantly high.

From the PDPA measurements it is thus concluded that the significantly dense spray has a maximum $|x|$-value around 0.09 m.

4.1.2 Deposition analysis results

As explained in Subsection 2.3.1 two pieces of paper were stuck together to cover the whole surface of the target cylinder and the liquid containing very few Tinopal was sprayed onto the paper. These two papers are photographed separately, showing the deposition of the Tinopal. The length of the papers is the same as the length of the cylinder, 0.265 m. The width of the papers is 0.211 m. The photographs of the deposition pattern of Tinopal on the target cylinder are shown in Figure 4.1. From the photographs it is seen that the spray has a vague boundary between the dense and the dilute part of the spray. This was also seen in the density results of the PDPA analysis. On the other hand, the photographs show a reasonably clear border of very high deposition density to very low deposition density is seen.

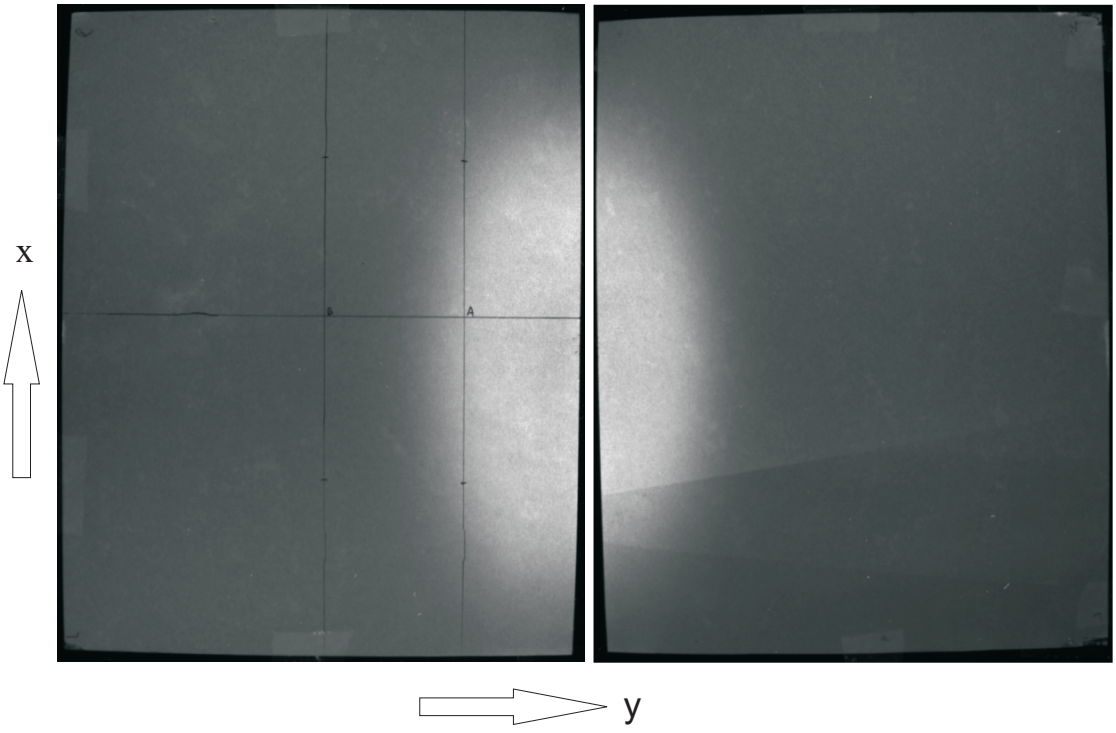


Figure 4.1: Photographs of the deposition pattern of the substance Tinopal.

The deposition pattern is a symmetrical ellipse around the center of the target cylinder. The length of the ellipse is about 0.17 m and the width of the ellipse is about 0.11 m. This agrees with a maximum absolute x-value ${r_{dep}}^{max-x} \simeq 0.085\ m$ and a maximum y-value ${r_{dep}}^{max-y} \simeq 0.049\ m$ (both projected on the (x, y)-plane). This is consistent with the PDPA results giving a ${r_{dep}}^{max-x} \simeq 0.09\ m$. No deposition was found on the sides of the target cylinder.

The qualitative deposition analysis on the walls and the ground with the water-and-oil-sensitive paper is very rough. The liquid that contacts with the paper reacts with the paper and causes the color change. A disadvantage is that the a kind of oil is produced that can diffuse or maybe even flow. Therefore the analysis was done soon after the spraying was stopped.

Some weak color change was seen at the ground in front of the cylinder. At the left wall in Figure 2.7 (a) some deposition is observed. The shape was about a half ellipse with the top at $z \simeq$ -0.16 m and the basis line at $z \simeq$ -0.23 m. These results are in accordance with the PDPA measurements, because the height of the deposited droplets on the wall agrees with that expected from Figure 4.2.

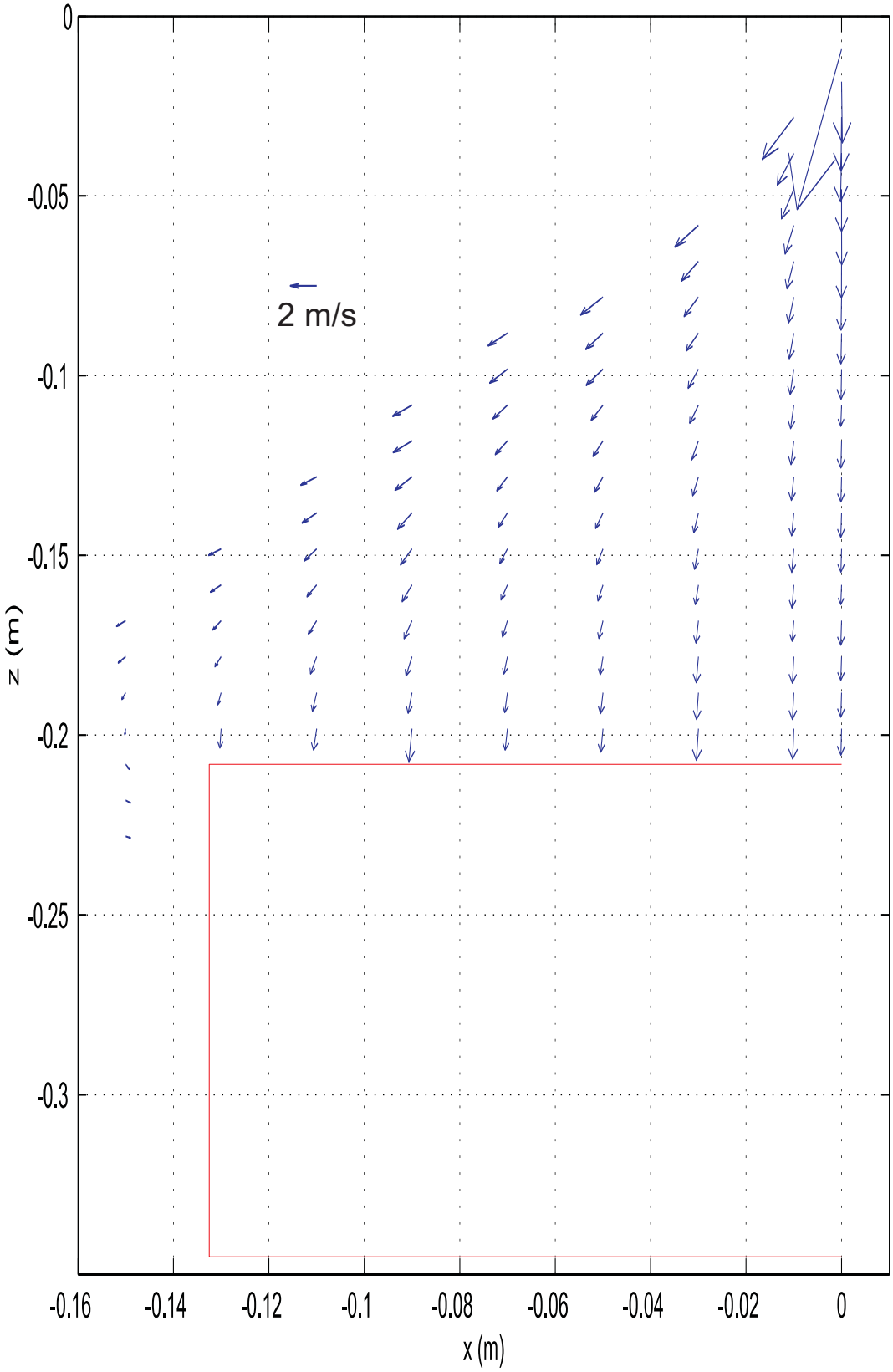
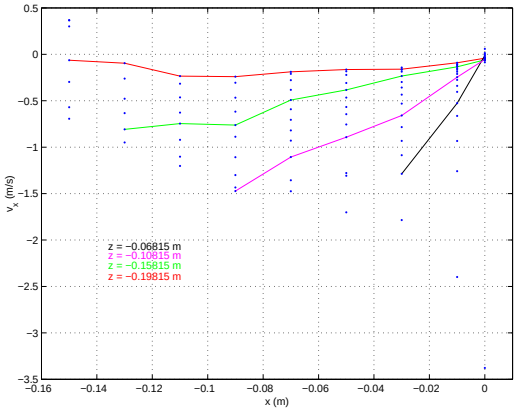
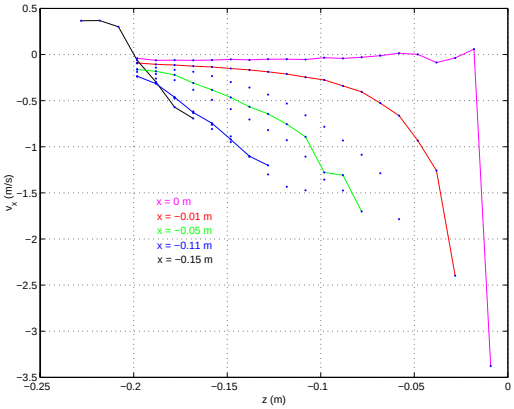


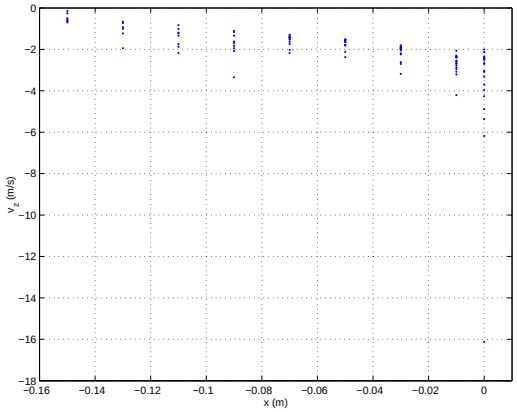
Figure 4.2: Velocity profile from the PDPA measurements.



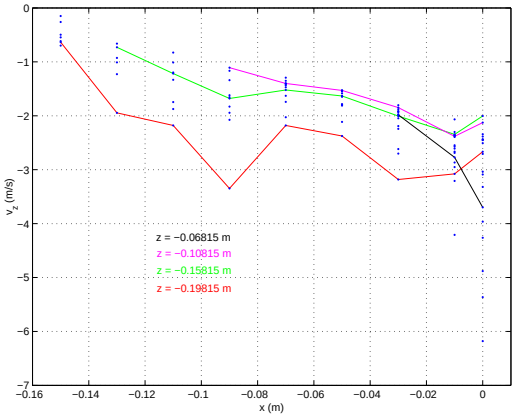
(a) Velocity in x-direction versus position in x-direction.



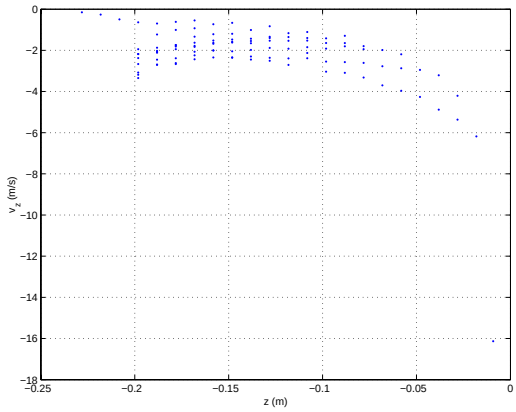
(b) Velocity in x-direction versus position in z-direction.



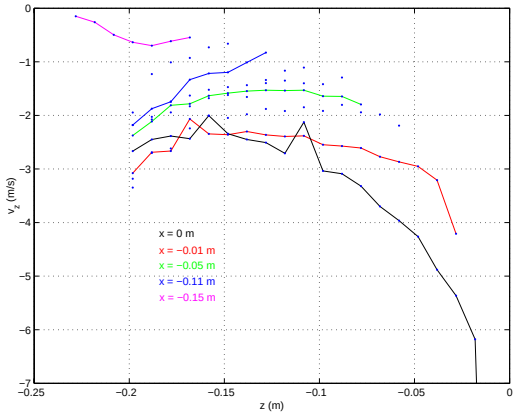
(c) Velocity in z-direction versus position in x-direction.



(d) Subfigure (c) at different scale.

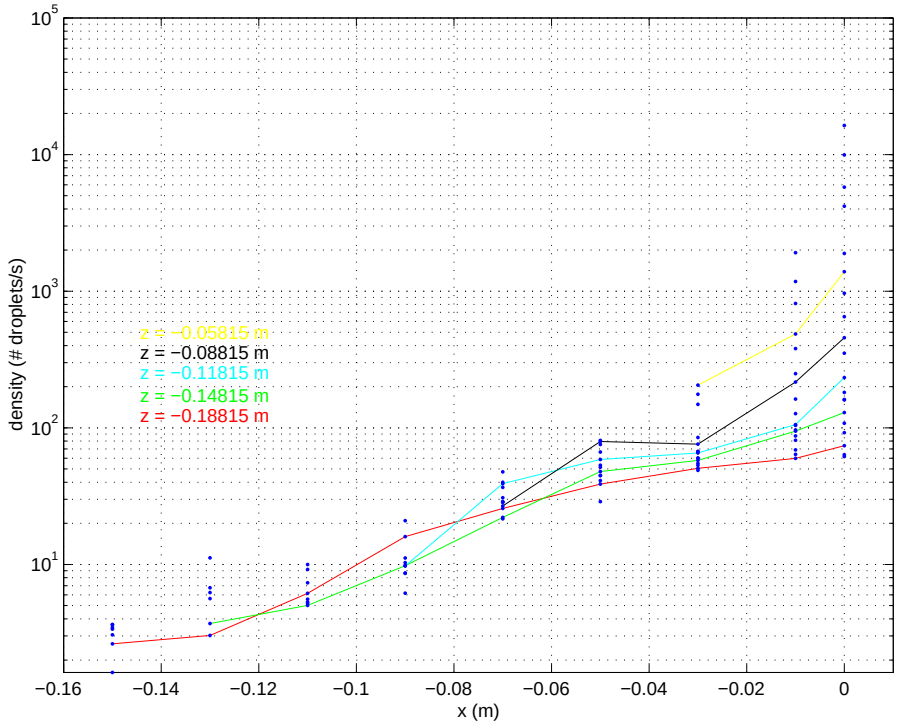


(e) Velocity in z-direction versus position in z-direction.

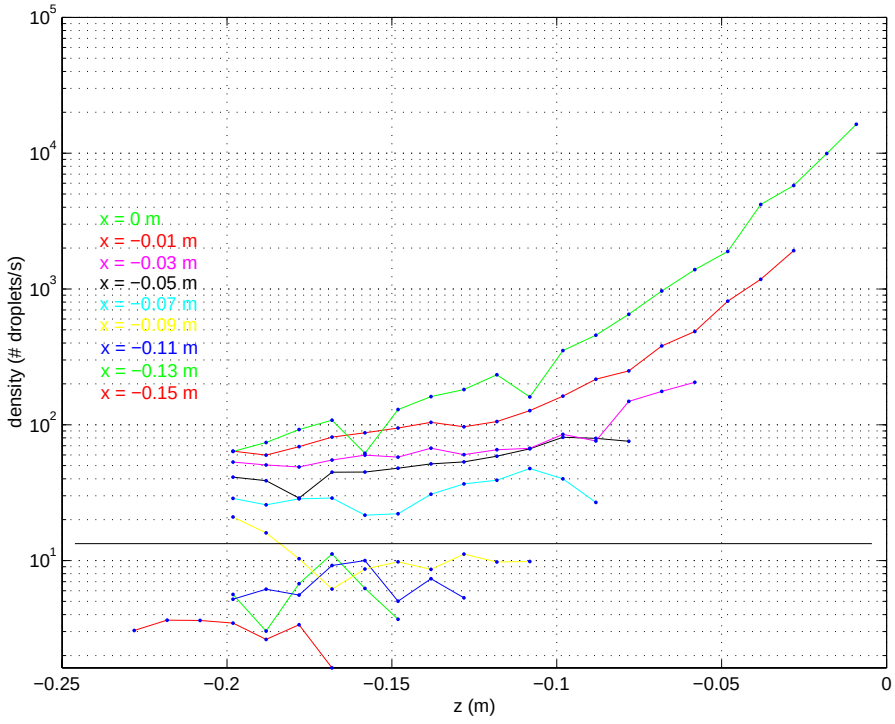


(f) Subfigure (e) at different scale.

Figure 4.3: Velocity versus position from the PDPA measurements.



(a) Droplet density versus x from the PDPA measurements.



(b) Droplet density versus z from the PDPA measurements.

Figure 4.4: Density results of the PDPA measurements.

4.2 Model

This section treats the results obtained with the simulations that are done. In the first section the developed scale-up correlation is checked for the interpolation using the final parameter set, and for extrapolation towards the experimental spray. After that the results of the final simulations are presented.

4.2.1 Scale-up correlation

Validation for the final parameter set

As mentioned in Section 3.6 the set of parameters for the scale-up simulations was different from that of the final simulations. The differences are given again in Table 4.1. The droplet diameter is changed according to the PDPA measurements. The reasons for the changes of α_1 and $|\vec{v}_{tot}|$ are earlier mentioned and discussed again in Subsection .

Table 4.1: Parameter sets for scale-up and final simulations.

	d (μm)	α_1 $(^\circ)$	$\lvert\vec{v}_{tot}\rvert$ (m/s)
scale-up	15	-10 - +10	15
final	8	-40 - +40	17

The validity of the scale-up correlation in Equation (3.12) is checked for the final set of parameters by comparison of the results of five simulations with that new set of parameters. This is done by interpolation: the droplet production times are within the range where the correlation is developed. The first simulation is done with droplet production time $t_{prod} = 1.25\ 10^{-3}\ s$ and Coulomb charge factor $f_q = 6$. The other simulations are done with the same t_{prod} as used for the scale-up simulations mentioned in Section 3.6. The f_q for those simulations is calculated with the correlation using the parameters of the first simulation.

Table 4.2: Results of validity check of the scale-up correlation.

t_{prod} (s)	f_q $(-)$	$< r_{dep} >$ (m)	${r_{dep}}^{max-x}$ (m)	${r_{dep}}^{max-y}$ (m)	N_{dep} $(\#\ droplets)$
$1.25\ 10^{-3}$	6	0.041	0.084	0.053	298
$2.5\ 10^{-3}$	8.71	0.042	0.082	0.054	151
$5\ 10^{-3}$	12.6	0.042	0.079	0.047	77
$10\ 10^{-3}$	18.4	0.041	0.078	0.046	37
$20\ 10^{-3}$	26.6	0.041	0.076	0.048	22

According to Equation (3.11) the same quantity that has been used to find the correlation is used to analyse the validity, namely the average distance of droplet deposition from the center point of the cylinder $< r_{dep} >$. Also the maximum distance of droplet

deposition in x and y direction, ${r_{dep}}^{max-x}$ and ${r_{dep}}^{max-y}$, and the number of deposited droplets N_{dep} are analysed. The simulation results are presented in Table 4.2.

From the $< r_{dep} >$ results in Table 4.2 it is concluded the interpolation of the scale-up correlation works quite well for the new parameter set. The ${r_{dep}}^{max-x}$ and ${r_{dep}}^{max-y}$ of the five simulations are also in a reasonably narrow range, but less than for $< r_{dep} >$. A discussion follows in Subsection 4.4.

Validation with experimental t_{prod}

The goal of the modelling is of course to simulate the EHDA spray that is experimentally measured. Applying the scale-up correlation, the model can be extrapolated towards the experimental spray with many droplets. Extrapolation means that the t_{prod} is outside of the range where the correlation is found. Both experimental and modelling t_{prod} and f_q are needed for the extrapolation. In the experimental situation $f_q = 1$ of course. The t_{prod} for the final simulations is chosen at 1.25 10^{-3} s as a compromise between computing effort and realistic droplet numbers. Finally the experimental t_{prod} is needed to calculate the f_q for the simulation.

This experimental t_{prod} is obtained in three ways. The first two ways are from the PDPA measurements at the point of droplet production $(x, z) = (0\ m, \text{-}0.00915\ m)$. The first way to obtain the t_{prod}, called "PDPA size", is to divide the droplet volume, calculated with the measured droplet size at the end of the jet, $d_{32} = 22\ \mu m$, by the volume flow rate. The second way, called "PDPA density", is to obtain the t_{prod} by dividing the analysis time by the number of droplets analysed at that point. The third way to obtain the experimental t_{prod}, called "Hartman size", is to calculate t_{prod} dividing the droplet volume, using the droplet size calculated with Equation (2.2) from Hartman [6], $d = 24\ \mu m$, by the volume flow rate. The production time estimates and the calculated Coulomb charge factors for the simulations are presented in Table 4.3.

Table 4.3: Experimental parameters from PDPA analysis and Hartman [6].

	$t_{prod}\ (s)$	$\frac{1}{t_{prod}}\ (Hz)$	d or $d_{32}\ (\mu m)$	$f_q\ (-)$
PDPA density	6.1 10^{-5}	1.6 10^{4}	-	5.1
PDPA size	3.0 10^{-6}	3.3 10^{5}	22	26
Hartman size	3.6 10^{-6}	2.8 10^{5}	24	23

From Table 4.3 it appears that the last two estimates for the t_{prod} are agreeing quite well, but the first is very different. A discussion on the reliability of the found droplet production times is presented here. After that a brief conclusion on the validation is given.

The method "PDPA density" is considered to give wrong results. This is because it is impossible to measure exactly at the end of the jet, the production point of the droplets, with a PDPA. The measuring volume cannot be put exactly at that point, already mentioned in Subsection 4.1.1. Furthermore, many droplets going through the measuring volume are considered wrong by the PDPA for this measurement (discussed

in Subsection 2.2.4). The number of droplets analysed is therefore not the real number of droplets going through the volume, but much smaller so that t_{prod} becomes too large. A last reason for an error is that the droplet detection rate of the PDPA could be less than the rate at which the droplets go through the measuring volume (the density). This means the number of counted droplets is too few. The detection rate of the PDPA is not cleared up but is expected to be in the range of the droplet density. For these reasons the measured initial droplet density has a large error and the t_{prod} cannot be calculated according to the PDPA density measurements.

The fact that it is impossible to measure the exact initial droplet properties with a PDPA affects the initial droplet diameter much less than for the density and the velocities. If the measuring point is slightly next to the droplet production point the measured droplet size is a very good estimation for the initial droplet size, because evaporation has very few influence over such a little distance. Therefore the $d_{32} = 22$ μm is a good estimation for the initial droplet diameter.

The initial droplet size according to Hartman's scaling laws is close to the measured droplet diameter, namely $d = 24$ μm. The main reasons for the difference are the errors in both the measurement, the model Hartman developed and the liquid properties in Table 3.1 used. Two other reasons of less influence are mentioned already in Section 3.5. (i) The measured droplet size is the Sauter mean diameter of a narrow polydisperse distribution and Hartman assumes the ideal case of a monodisperse size distribution. (ii) The production of droplets might be different from Hartman's observations, having to do with the difference in experimental set-up.

The difference in droplet diameter of Hartman compared to the PDPA diameter of factor 1.09 results in an increase of f_q of factor 1.13. From the results it appears that the real experimental t_{prod} and thus the f_q for the final simulation, can be estimated quite well with the two available methods based on the initial droplet diameter. As will be discussed in the next subsection a f_q of 5 to 6, for a $t_{prod} = 1.25\ 10^{-3}$ s, will give a simulated spray similar to the experimental one. The f_q's from the experimental methods do thus not simulate the experimental spray with the present model. The droplets have a too high Coulomb charge and the spray becomes too broad due to the repulsions in those situations. A more detailed discussion on the validation with an experimental t_{prod} will follow in Section 4.4.

4.2.2 Final simulation results

The two final simulations are done with the parameters $t_{prod} = 1.25\ 10^{-3}$ s and $f_q = 5$ or $f_q = 6$. These values are chosen because from several (trial) simulations it appeared that these combinations of t_{prod} and f_q will have about the same deposition dimensions as found for the experimental EHDA spray.

As explained in Section 3.1 the model simulates the spray from the start up and after a while the steady state is reached. To calculate the properties of the droplets in the spray the two final simulations are analysed in the steady state only. In Figure 4.5 the number of droplets versus the simulation time is shown for the simulation with $f_q = 6$. From this figure the steady state is determined. The final simulation time t_{stop} and the time when the steady state is reached t_{steady} for both simulations are summarised

in Table 4.4. It appears that for a constant t_{prod} the t_{steady} increases with increasing f_q. This is because the spray becomes broader and the droplets have to travel a larger distance at the edge of the spray.

The velocity and density plots as presented in Subsection 4.1.1 for the PDPA analysis are shown in this subsection for the simulation with $f_q = 6$ only. This is done because both simulations give similar results and because that simulation resembles the experimental spray more. Some important results from the simulation with $f_q = 5$ are presented in Table 4.4 and 4.5.

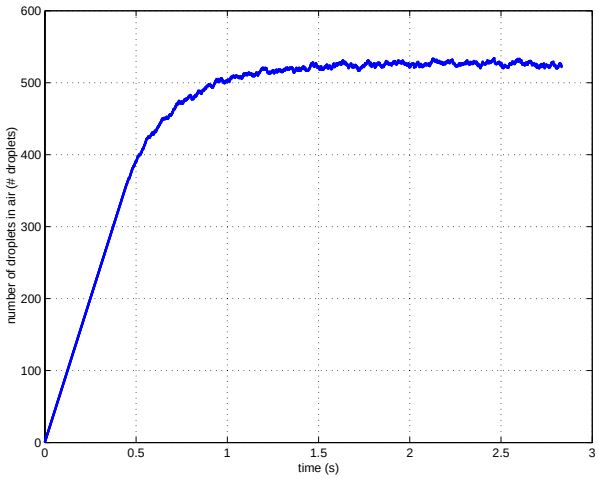


Figure 4.5: Number of droplets versus simulation time; $f_q = 6$.

The method to calculate the results (velocities and density) of the simulations is illustrated in Figure 4.6. The results of the PDPA are measured at a single spot in the (x, z)-plane. Because of the much lower droplet density in the simulations, this is statistically not reliable to do there too. Therefore an imaginary cube is put around each measuring point of the PDPA analysis. The properties of the simulated droplets that are within such a cube in the steady state are used to calculate the average set of properties, representing the result for that measuring point.

Table 4.4: Simulation time parameters for the final simulations.

t_{prod} (s)	f_q $(-)$	t_{stop} (s)	t_{steady} (s)
$1.25\ 10^{-3}$	5	2.38	1.5
$1.25\ 10^{-3}$	6	2.83	1.8

This method of obtaining the results is accompanied with some problems. The first problem is that due to the relative few droplets in the simulated spray, the density of the droplets is relatively low. At some places in the spray no droplets will be present at all while at others several droplets pass by. This distorts the results to a certain extent. The second problem occurs at the edge of the spray. The evaluation cubes in that area are partly never filled with droplets. This means that the density in those cubes is much

less than cubes inside the spray. This will be clear from the density results. When the cubes are made smaller, the volume that is never filled with droplets is decreased, improving the density results. But smaller cubes will naturally contain less droplets, which will increase the unreliability of the velocity results. A cube size of 0.005 m was too small to obtain nice results, while a cube size of 0.01 m is considered to have too bad density results. Therefore a compromise is found at a cube size of 0.0075 m. The density problem at the edge of the spray is still there as will be clear later. All evaluations gave approximately the same velocity results.

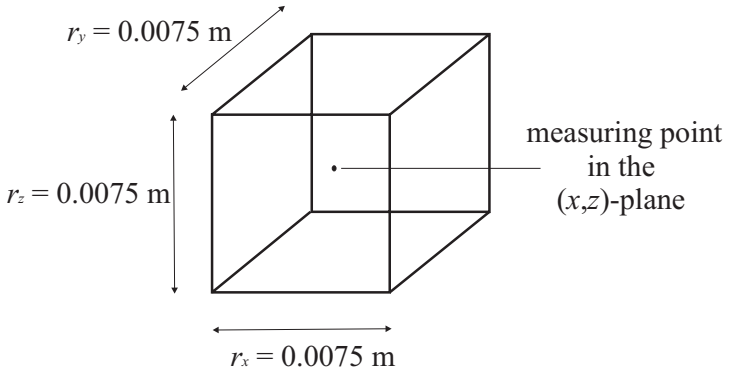


Figure 4.6: Illustration of a volume used to calculate the simulation results.

Deposition results

The deposition pattern of the final results can be characterised by the shape of the pattern and the maximum distance from the center point of the target cylinder of a deposited droplet in x- and y-direction, ${r_{dep}}^{max-x}$ and ${r_{dep}}^{max-y}$. In Figure 4.7 the elliptical deposition pattern of the simulation with $f_q = 6$ is shown. Table 4.5 summarizes the results for both the final simulations.

Table 4.5: Deposition results for the final simulations.

f_q $(-)$	${r_{dep}}^{max-x}$ (m)	${r_{dep}}^{max-y}$ (m)	N_{dep} $(\#\ droplets)$
5	0.077	0.051	1406
6	0.094	0.062	1744

Velocity results

The velocity vector plot is shown in Figure 4.8. The velocity in x- and z-direction versus the position in those directions are presented in Figure 4.9.

The velocity vectors in Figure 4.8 are five times more enlarged than the vectors of the PDPA measurements in Figure 4.2. The initial z-velocity is very high in comparison to the z-velocities further away from the nozzle. This also is seen in Figure 4.9 (c) and

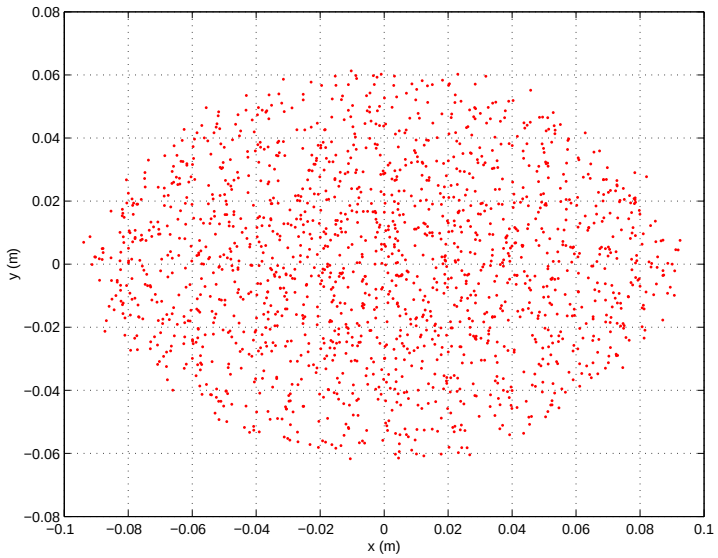


Figure 4.7: Deposition pattern from the final simulation with $f_q = 6$.

(e). The high initial x-velocity of the droplets, treated in Section 3.5, is not seen back in the the profiles in Figure 4.9 (a) and (b). Both last two observations show that the droplets slow down very fast in both directions.

The trends in the subfigures are similar to those for the PDPA in Figure 4.3, taking into account the dense experimental spray as explained in Subsection 4.1.1. Therefore a discussion similar to the PDPA results is omitted here.

Density results

In Figure 4.10 the density results for the final simulation with $f_q = 6$ are presented. The trend lines are shown to indicate how the density develops in x- or z-direction. At the edge of the spray the density drops due to the explained method of obtaining the results. The interpretation of the density trends in the plots is therefore somewhat difficult. From Figure 4.10 (a) it appears that at one height z the density is roughly constant with decreasing $|x|$. Figure 4.10 (b) shows that with decreasing z the density decreases.

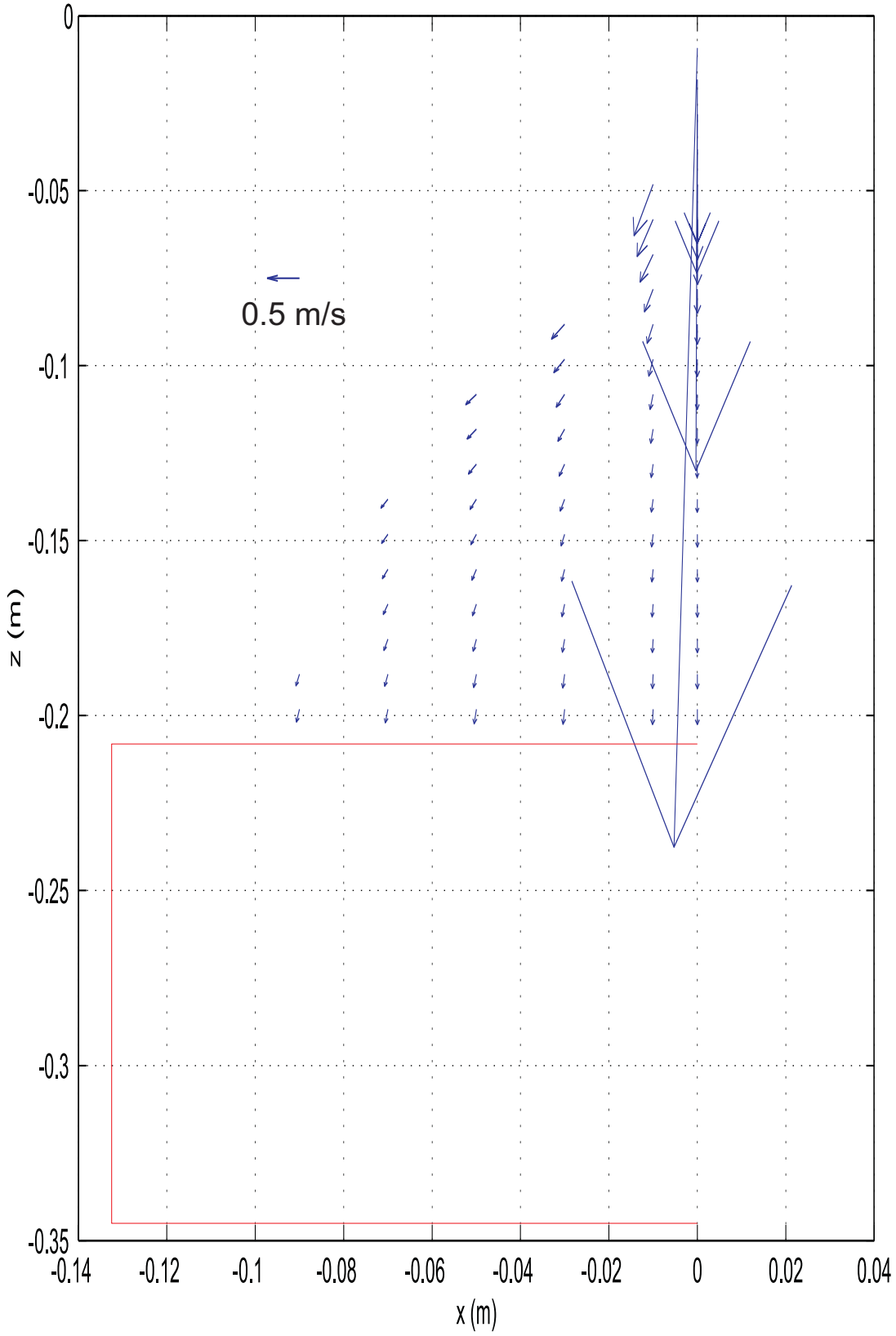
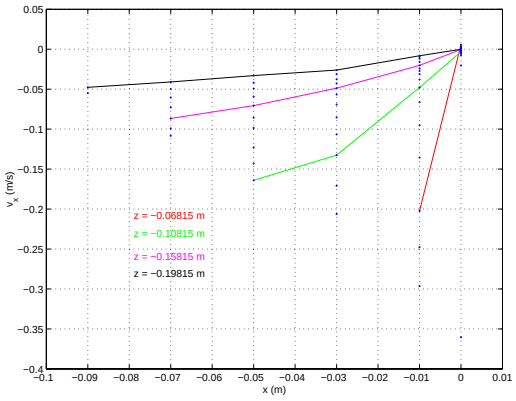
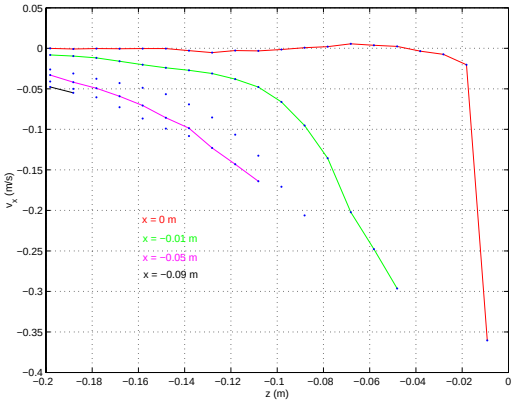


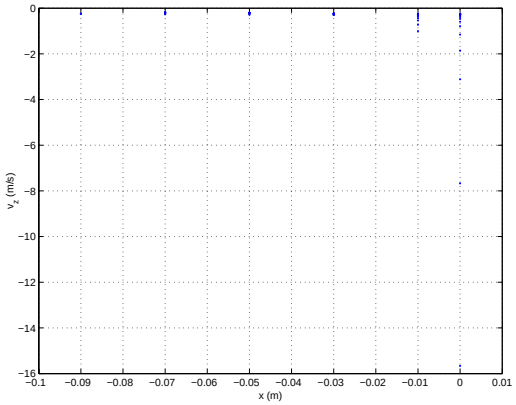
Figure 4.8: Velocity profile from the final simulation with $f_q = 6$.



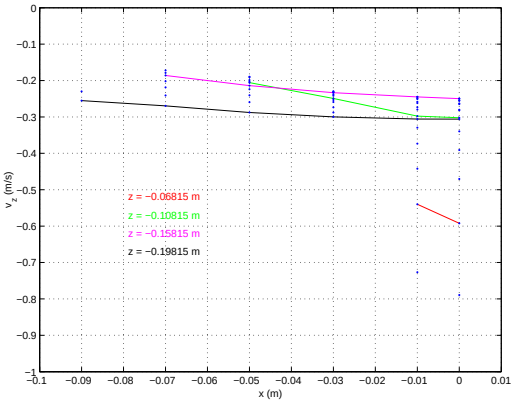
(a) Velocity in x-direction versus position in x-direction.



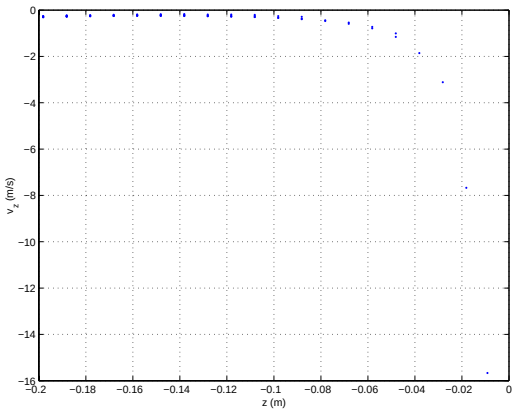
(b) Velocity in x-direction versus position in z-direction.



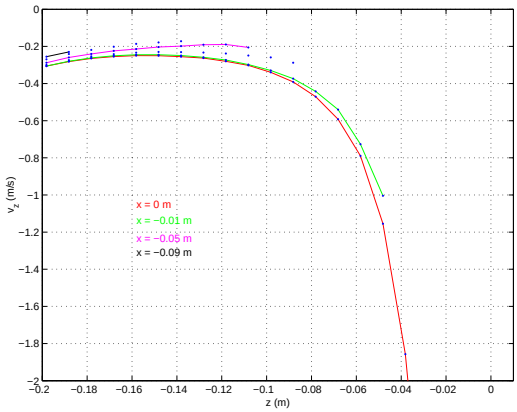
(c) Velocity in z-direction versus position in x-direction.



(d) Subfigure (c) at different scale.

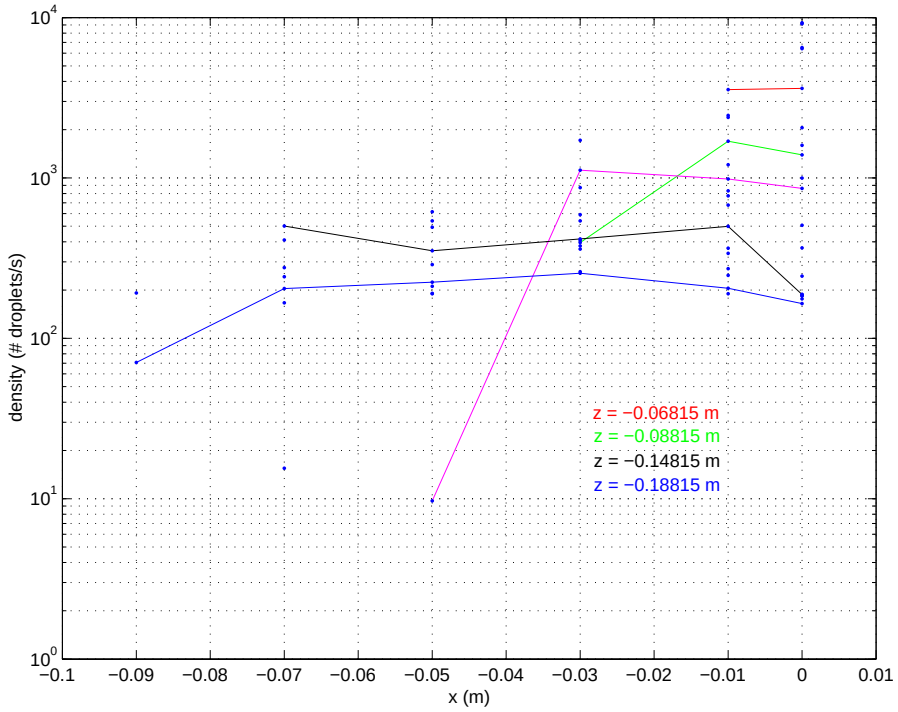


(e) Velocity in z-direction versus position in z-direction.

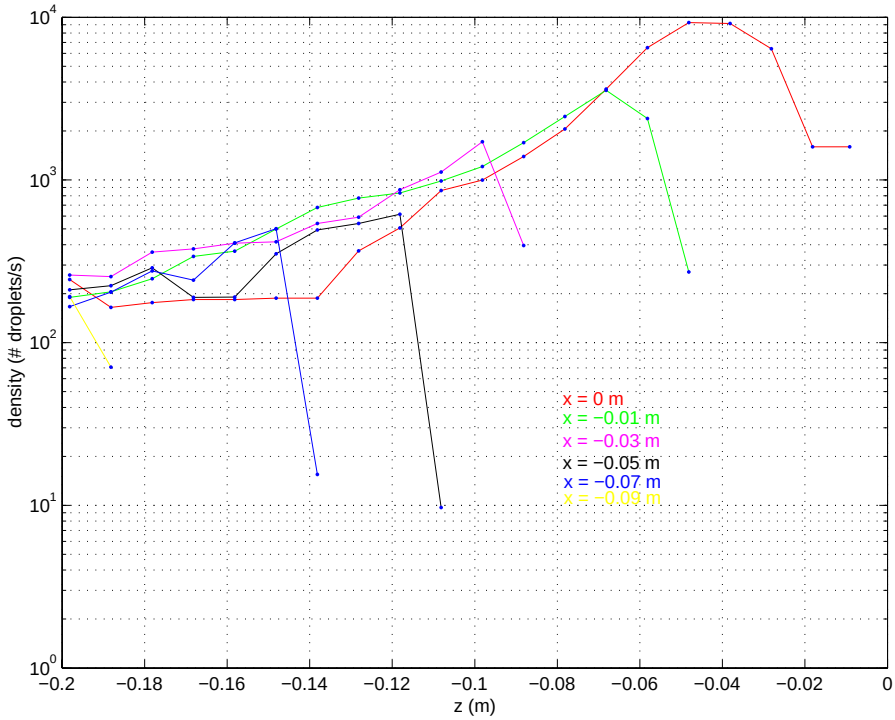


(f) Subfigure (e) at different scale.

Figure 4.9: Velocity versus position from the final simulation with $f_q = 6$.



(a) Droplet density versus x from the final simulation with $f_q = 6$.



(b) Droplet density versus z from the final simulation with $f_q = 6$.

Figure 4.10: Density results from the final simulation with $f_q = 6$.

4.3 Verification

In this section the verification of the model will be discussed. First the shape and the deposition, velocity and density profiles of the experimental and simulated sprays from the Sections 4.1 and 4.2 are compared and discussed. In the next section the from the verification apparent discrepancy between the model and the reality is discussed.

4.3.1 Deposition

The simulations give an elliptical droplet deposition pattern just as the real EHDA spray. In Section 4.2.1 it is explained that the f_q of the final simulations are chosen such that the deposition dimensions of the simulation resembles the experimental findings. In Table 4.6 these deposition parameters are listed. The simulated spray gives a clear deposition boundary because of (droplet production in) the model. The experimental deposition pattern does not have a sharp edge as discussed in Subsection 4.1.1 and 4.1.2, but a distinction is made between the dense and dilute spray according to the density profile of the spray and the deposition pattern.

It can be seen that the ${r_{dep}}^{max-x}$ of the simulation with $f_q = 6$ is close to the experimental dimension. The simulated ${r_{dep}}^{max-y}$ is somewhat too high. This might be due to the different way the droplets are produced compared to the reality (also possible from the density results to be discussed). No conclusions can be drawn from the distribution of the droplets because this is not investigated experimentally. Another possible reason is the fact that no image force is modelled. This force could attract the droplet more to the target cylinder causing a lower ${r_{dep}}^{max-y}$.

Despite of this discrepancy in ${r_{dep}}^{max-y}$ it can be concluded that the shape of the elliptical deposition pattern for the simulation with $f_q = 6$ resembles the experimental pattern well.

Table 4.6: Comparison of experimental and simulated deposition dimensions.

	${r_{dep}}^{max-x}$ (m)	${r_{dep}}^{max-y}$ (m)	${r_{dep}}^{max-x}/{r_{dep}}^{max-y}$ $(-)$
experiments	~ 0.09	~ 0.049	1.8
simulation $f_q = 5$	0.077	0.051	1.5
simulation $f_q = 6$	0.094	0.062	1.5

4.3.2 Shape

From the presented velocity profile and density results it appears that the shape of the simulated sprays resemble the experimental spray quite well. Only in the initial stage, close to the nozzle, the simulated spray is too narrow. With decreasing z the simulated spray becomes more broad due to the Coulomb repulsions and the shape resembles the experimental shape quite well. But because the spray is too narrow initially, it stays slightly less broad than the experimental spray. From this point of view the simulated

spray is shifted downwards compared to the real spray with a distance of a few cm. It is expected that the simulated spray is initially too narrow due to the difference in Coulomb interactions compared to the reality. In that area droplet density is highest and therefore the Coulomb repulsions are highest. The real droplets are close to each other causing a great Coulomb repulsion resulting in a broader spray. Due to the different Coulomb interactions in the model the broadening is different. A detailed discussion on the difference in Coulomb interactions close to the nozzle is given in Section 4.4.

4.3.3 Velocity

Above it is mentioned that the shape of the simulated spray resembles the experimental shape well, except for the shift in z-direction. The similarity in shape can also be seen from the similarity of the directions of the velocity vectors for both situations, shown in Figures 4.2 and 4.8.

The presented velocity profiles, plots of v_x and v_z versus x and z, are also similar for the experimental and simulated spray. This can be seen in Figure 4.3 and 4.9, taking into account the dense part of the experimental spray discussed in the experimental results. One striking resemblance of the experimental and simulation results can be seen in the plots of v_z versus z. Closer to the target the droplets are accelerated towards the target cylinder, while no image force is modelled (Section 3.3). The reason for this observation is not clear. It might be caused by an increasing repulsion acting on the droplets lower in the spray, because a larger amount of space charge is above those droplets.

Although the shape of all presented velocity profiles is similar, the great difference between the velocity profiles of the real and the simulated spray is the magnitude of the velocity. The simulated droplets start with a high z-velocity comparable to the initial velocity from the PDPA measurements, but they decelerate fast unlike the real droplets. A similar high deceleration is seen in the x-direction. The "final" velocity of the simulated droplets becomes therefore much smaller than that of the measured droplets.

4.3.4 Density

Due to the evaluation method to obtain results from the simulation, the density results are distorted at the edge of the spray, discussed in Subsection 4.2.2. A comparison of the density between the PDPA and simulation results is therefore difficult.

The simulation results give an almost constant density for measurements at one height z, whereas the PDPA results show that at one height, the density increases with decreasing $|x|$. A probable reason for this difference in density is the difference in the magnitude of the droplet velocities, explained as following. The simulation shows a fast drop in the velocity (in both directions) and then the droplets travel almost at the same velocity. In the real spray the decrease in velocities is much more gradual. Another possible reason is the different production of droplets. Maybe the droplets are produced according to a Gaussian profile instead of the random profile assumed in the model.

4.4 Discrepancy discussion

In the former subsections the results of the experiments and simulations are compared and the differences are shown. The model including the scale-up method appears to work quite well. The results are promising, but only a few main discrepancies between the simulation and the experiments are observed. In this subsection the possible reasons for those discrepancies are discussed. Both the measurements and the modelling are accompanied with errors. The expected errors in both activities are discussed.

In the following passage the errors in the PDPA experiments are estimated. The flow rate and high voltage are quite exact because of the good equipment used. The measured velocities and sizes have an error. Some measurements clearly do not follow the trend of the rest of the data. Despite of this the measured velocities that follow these trends are expected to have a relatively low error inherent to the analysis method of the PDPA. The density calculated from the measured number of droplets and the measuring time is expected to have a relatively high error. As mentioned in Subsection 2.2.4 the number of droplets analysed with the two colors of laser light are not the same and furthermore several droplets going through the probe volume are not taken into account due to their improbable combination of size and velocity. The density is expected to be higher than measured due to these errors. Despite of the expected significant error, the densities can be compared well. This is because in all measurements the number of counted droplets is about the same and therefore the densities have approximately the same error. This means the distinction between the so-called dense and dilute spray with the obtained densities is possible. The last and main error is in the PDPA measurement at the production point of the droplets, at the end of the jet. It is concluded that it is not possible to measure the initial droplet properties with a PDPA. A few mm away from this production point the droplets can be analysed, but still it is hard to set the equipment in such a position that the measurement is done exactly underneath the nozzle on the symmetry axis.

The way the dimensions of the deposition pattern of the sprayed Tinopal solution are obtained is quite rough. At the sides of the pattern the droplet density is lower, but the distinction between the so-called dense spray and the dilute spray is vague. Therefore an error in the order of 0.01 m is expected when determining the deposition boundary between dense and dilute.

The model for the EHDA spray is developed making several assumptions trying to formulate the process in mathematical equations and simulate the reality in a relatively simple way. Every assumption will be accompanied with an error, whatever the significance of that error. Expected errors of the assumptions made are discussed now.

The modelled droplet charge q is calculated by multiplying the Rayleigh limit charge with an efficiency. Both these quantities are extensively researched by Vercoulen [15] and Hartman [6] and are expected to have a low error. The calculation of the Rayleigh limit requires the properties of the liquid sprayed. Because the liquid mixture of ethanol and triethylene glycol is known, the properties of this mixture can be estimated with a low error.

The external electric field $\vec{E}$ is calculated with FEMLAB. The model system set-up resembles the main parts of the experimental set-up well and therefore it is expected that the error in the external electric field is not high. No verification of the electric field is done. The assumption of constant droplet diameter of one size will cause an error. The measured droplet size distribution is not monodisperse. Consequently the Sauter mean diameter d_{32} is taken as the representing diameter for one measurement. Furthermore evaporation occurs, decreasing the size of the droplets. The mean d_{32} of all PDPA measurements is taken as the droplet diameter for the model. Despite of this, the modelled droplet size does not differ very much of the measured sizes, and therefore this assumption is expected to have a systematic error.

An error could occur because no image force is modelled. As mentioned in Chapter 3 the existence of such a force concerning the deposition of droplets in an EHDA spray is not certain. In the comparison of the deposition pattern of the real and simulated spray it is mentioned that an image force could have influence on the droplet trajectories.

The above mentioned parameters are considered to have a low error. Some main reasons for the discrepancies between the reality and the simulation can be pointed out. The three topics that will be discussed in a separate subsection are: the initial droplet properties, the Coulomb interactions close to the nozzle and the drag and hydrodynamics concerning the air flow. Before these subjects are treated, the major assumption of the model being the scale-up of the Coulomb interactions between the droplets, is discussed. This involves the accuracy, the interpolation and the extrapolation of the scale-up correlation.

4.4.1 Scale-up correlation, accuracy and validation

Several reasons contribute to some inaccuracy while developing the scale-up correlation. These are mainly due to the chosen method of producing it with the deposition results, namely analysing the average distance of droplet deposition from the center point of the cylinder $< r_{dep} >$.

First the simulation time of the scale-up simulations was 1 s. This is reasonably short because in most cases the steady state has not been reached, although sometimes very closely. The simulation time for the scale-up simulations is not increased because of the lack of time for doing the time-consuming simulations! A larger simulation time will cause more droplets to be deposited and the steady state to be reached. This is observed in the interpolation validation check in Subsection 4.2.1, Table 4.1. A difference in ${r_{dep}}^{max-x}$ and ${r_{dep}}^{max-y}$ is seen for the different simulations. All simulated sprays are not in the steady state and also at different time away from that state. Therefore the number of deposited droplets is different. The ${r_{dep}}^{max-x}$ and ${r_{dep}}^{max-y}$ quantities are maximum values and not averages, and can therefore only be compared in a good way in the steady state. Then this difference is expected to be minimal. Because the correlation is produced based on the $< r_{dep} >$ of the scale-up simulations and because this is an average quantity, it is constant for the validation simulations. This actually validates the use of the correlation.

Also the range of the scale-up parameters t_{prod} and f_q for the production of the correlation is limited. This is because the target cylinder is not symmetrical and also

finite. Both the symmetry and the finiteness of the target influence the chosen analysis parameter $< r_{dep} >$. When the spray becomes broader, due to higher t_{prod} and f_q, the influence of the finiteness and non-symmetry becomes bigger and the accuracy of the correlation decreases.

A last comment is that the accuracy of the correlation will be even better if the number of scale-up simulations is higher, although already 35 simulations are done. However, within the just explained limits of the scale-up parameters, the available parameter space was scanned.

Despite of the discussed possible inaccuracies the scale-up correlation is validated for interpolation for a different set of parameters than used for the production of the correlation. The extrapolation towards the experimental spray is not succeeded as presented in Subsection 4.2.1. This does not mean that the scale-up method is a failure. From the verification it resulted that the model including the scale-up method gives nice results. From the similarity of the deposition, shape and velocity profiles of the simulations and the experiments, it is seen that the scale-up method works quite well and is valuable. The only two main discrepancies appear from the final simulation: close to the nozzle the spray does not resemble the real spray and the magnitude of the velocity is too low compared to the reality. These discrepancies in the model will have to be resolved first before the extrapolation of the scale-up correlation can be validated or invalidated.

A foresight is given to illustrate that extrapolation of the scale-up correlation is expected to be more successful after resolving the difference in magnitude of velocity is resolved. The idea is that when the velocity of the simulated droplets is higher, the flux of droplets through the air is higher and thus the number of droplets in the air is less (in the steady state). This will cause lower Coulomb repulsions. The correction for the droplet flux of the high f_q, calculated with the current estimations for t_{prod}, can be imagined by an additional factor multiplied by the Coulomb interaction terms. This could for example be $f_{flux} = \frac{\vec{v}_{drop,model}}{\vec{v}_{drop,reality}}$. The correction of the high f_q of 23 to 26 from Table 4.3, could lead to a value that is near that of the current final simulation, namely $f_q = 6$.

A second reason that causes the extrapolation of the scale-up correlation to be more difficult, of less concern in this stage of the model, is that the values of the determined real t_{prod} have some error. The droplet diameters according to Hartman and with the PDPA measured are in quite good agreement. But an error in the diameter gives an error in the t_{prod} eight times as high according to Equation (2.2). The resulting error in t_{prod} leads to an error in f_q. This could lead to a difference in estimated final f_q and the f_q that will really model the EHDA spray.

Even if the model is adapted and the experimental t_{prod} is known more accurately it might be impossible to validate the scale-up correlation with experimental data. Once again this does not mean that the scale-up method is unusable. The results show that the model is promising and expected to be of great value when the discrepancies are resolved. Furthermore an error in t_{prod} results in an error in f_q that can influence the simulation results greatly.

4.4.2 Initial droplet properties

One of the major assumptions in the model is the calculation of the initial properties of the droplets. The droplets are produced with a random distribution of $r_{initial}$ and angle α_2, which determine the position of the droplet. Another randomly produced angle α_1 determines the initial angle of the velocity of the droplets relative to the z-axis. Furthermore the initial velocity $|\vec{v}_{tot}|$ is constant. Possibly the real distribution of initial position is Gaussian instead of random, caused by the oscillation of the jet. The assumed droplet production is a good approach of reality though.

The maximum initial $|x|$ and $|y|$ of 1.0 μm is believed to be realistic, based on Hartman's research. Also the constant initial z at the end of the jet is a realistic assumption. The range of the initial velocities in all directions is unknown though. In the model this depends on the angle $|\alpha_{1,max}|$. According to Hartman the initial radial velocity is close to zero resulting in a very low $|\alpha_{1,max}|$. In Section 3.5 it is mentioned that this is not taken into account, but a high $|\alpha_{1,max}| = 40°$ was chosen to create a broader spray close to the nozzle spray as experimentally is observed. A larger $|\alpha_{1,max}|$ causes a broader range of the initial velocities in all directions, according to Table 3.3.

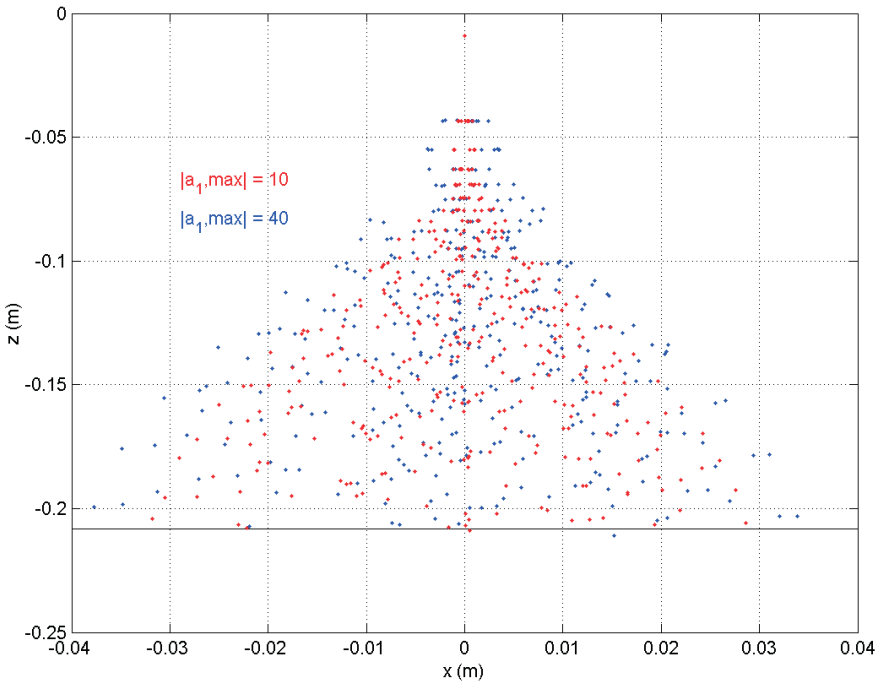


Figure 4.11: Illustration of the influence of $|\alpha_{1,max}|$ on the broadening of the spray.

The influence on the broadening of the spray of an increase in $|\alpha_{1,max}|$ is shown with two simulations with a $f_q = 8$ and $t_{prod} = 1\ 10^{-2}\ s$ are compared. The first simulation has an $|\alpha_{1,max}| = 10°$ (scale-up parameter), the second has an $|\alpha_{1,max}| = 40°$ (final parameter). The results are presented in Figure 4.11 and Table 4.7. From these results it is seen that an increase of factor 4 in $|\alpha_{1,max}|$ causes that the deposition dimensions

become larger with a factor of only 1.2.

Table 4.7: Illustration of the influence of $|\alpha_{1,max}|$ on the broadening of the spray.

$\lvert\alpha_{1,max}\rvert$ (°)	${r_{dep}}^{max-x}$	${r_{dep}}^{max-y}$
10	0.032	0.021
40	0.038	0.028

The influence of $|\alpha_{1,max}|$ is only due to the very initial spread of the droplets. The trajectories of the droplets are not shown close to the nozzle but will be seen in Figure 4.12 in the next subsection. The shape of the spray is initially according to the angle $|\alpha_{1,max}|$. But after a few mm distance in z-direction the v_x and v_y are already very low and the spray does not follow the initial angle anymore. The simulated spray is much more narrow close to the nozzle than is seen experimentally. The influence on the broadening of the spray close to the nozzle of $|\alpha_{1,max}|$ is thus far from that was desired. From these results it follows that some other features are responsible for the experimentally observed broadness of the spray. One possibility that is discussed in the next subsection is the difference in Coulomb interactions close to the nozzle between the reality and the model. A difference in drag force may have an influence too, as discussed in the last subsection.

4.4.3 Coulomb forces close to the nozzle

The scale-up method influencing the Coulomb interactions allows to simulate the spray with much less droplets than in real life. The Coulomb interactions between the droplets are increased by increasing their charge. Comparing the shape of the experimental and simulated spray it appeared the latter is less broad than the former.

Close to the nozzle the simulated spray is too narrow and consequently the spray stays less broad than the experimental spray as mentioned in Subsection 4.3.2. This initial difference in shape is not due to the magnitude of $|\alpha_{1,max}|$, thus it is assigned to the difference in Coulomb interactions between the reality and the simulations. To illustrate this two simulations in the area close to the nozzle are done. One simulation is done with droplets of real charge and a t_{prod} that comes close to the reality. The second simulation is done with a lower t_{prod} and f_q according to the scale-up method.

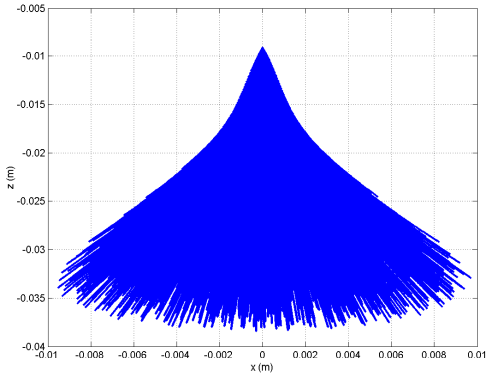
The simulation duration of both simulations was 0.0056 s. The first simulation is done with a Coulomb charge factor $f_q = 1$, and a droplet production time $t_{prod} = 1.25\ 10^{-6}$. This is expected reasonably close to reality, although the real t_{prod} is not cleared up in this work. The second simulation has a droplet production time $t_{prod} = 1.25\ 10^{-4}$ and a Coulomb charge factor $f_q = 12$, according to the scale-up correlation. This means few droplets are produced all having a higher charge, affecting their Coulomb interactions, than in reality. The rest of the parameters are the same for both simulations. The results are displayed in Figure 4.12.

From Figure 4.12 (a) and (b) it is clear that the spray becomes much wider initially (until $z \simeq$ -0.02 m) if more droplets of real charge are produced than if few droplets with relatively high charge are produced. At larger distance from the nozzle the sprays are similar to each other with respect to the direction of movement of the droplets at the edge of the spray. Apparently the scale-up method appears to work well, except in the area very close to the nozzle.

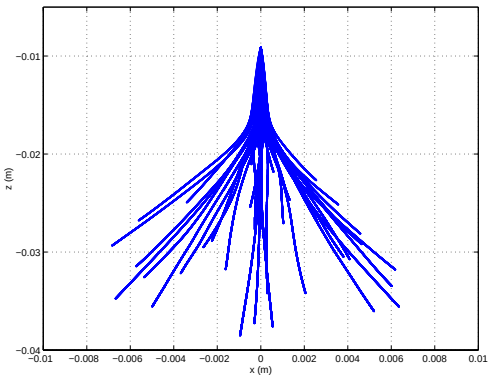
In reality many droplets will be produced and they will be close to each other initially. In every direction around a droplet there is another droplet present due to the high density. This causes a repulsion in all directions. An acceleration in x-direction and z-direction is the result, seen in Figure 4.12 (c) and (e). The same holds for the (y, z)-plane but these results aren't shown. This acceleration due to high Coulomb repulsions in the radial direction causes the broadness of the spray (in x- and y-direction).

In the model situation much less droplets are produced than in reality. The droplets are relatively far away from each other due to the low droplet production rate and are accelerated due to the electric field, seen in Figure 4.12 (d). Due to the resulting low droplet density the Coulomb repulsions are relatively low in x and y-direction. No acceleration in those directions take place and the droplets are faster decelerated (due to drag) than with a higher droplet production rate. As a result the spray is less broad initially (until $x \simeq$ -0.001 m) than in reality.

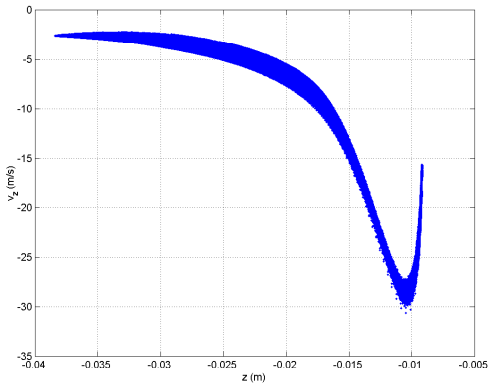
The observation that at larger distance from the nozzle the sprays are similar to each other is also seen in Figure 4.12 (c) and (d), and (e) and (f). The velocity profiles in z-direction are very similar. The velocity profiles in x-direction are also similar, thus except for the area very close to the nozzle. The comparison of the profiles in Figure 4.12 (e) and (f) is somewhat difficult because the last subfigure has much less droplets than the first subfigure. The resemblance is quite clear though.



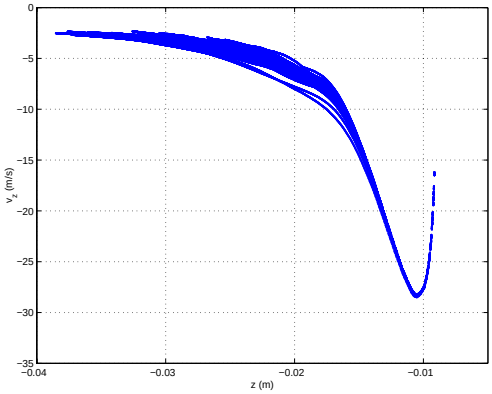
(a) Spray simulated with realistic parameters.



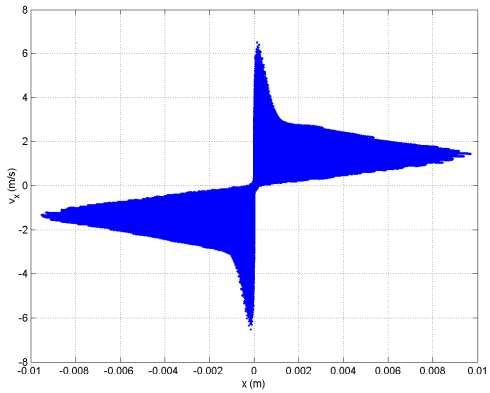
(b) Spray simulated with model parameters.



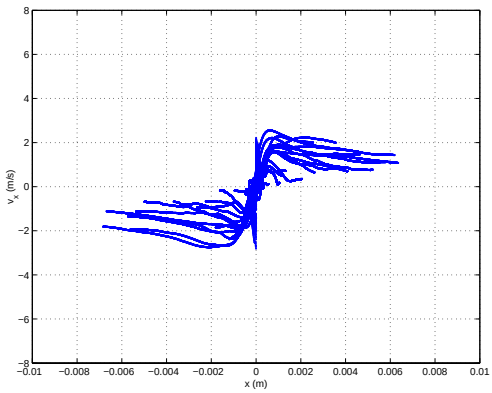
(c) Velocity profile in z-direction simulated with realistic parameters.



(d) Velocity profile in z-direction simulated with model parameters.



(e) Velocity profile in x-direction simulated with realistic parameters.



(f) Velocity profile in x-direction simulated with model parameters.

Figure 4.12: The influence of Coulomb interactions for simulations with realistic parameters $f_q = 1$ and $t_{prod} = 1.25\ 10^{-6}$ (a,c,e) and with model parameters $f_q = 12$ and $t_{prod} = 1.25\ 10^{-4}$ (b,d,f).

4.4.4 Drag and air hydrodynamics

Another major difference between the model and the reality is the drag due to the movement of the droplets in air. In the model it is assumed that the ambient air is at rest, $\vec{v}_{air} = 0\ m/s$. In reality the air velocity will not be zero [5], especially close to the nozzle. In that area the droplet density is very high and the air will have a quite high velocity in the turbulent flow regime. When the droplets are produced in great number they suck air into the spray causing a turbulent air flow. This will create eddies and the direction of the air flow is unknown. The droplets undergo forces due to the air flow, but they push away the air that they meet causing a drag force in the opposite direction of movement. The resulting $\vec{v}_{air}$ will therefore change with position and with time in that area. Further away from the nozzle the droplets have a lower velocity and the spray is less dense. It is expected that no turbulent air flow occurs in this area. But the magnitude of $\vec{v}_{air}$ in that area is also unknown but expected to be significantly high. Finally close to the target the air has to "leave" the spray and flow around the target. This is a third regime of the air flow.

To illustrate the importance of the ambient air the mass of air and droplets in the spray are estimated. This results in the ratio $\frac{m_{air}}{m_{drop}} \sim 10^3$. Only in the relatively small volume close to the nozzle, the droplet density and thus the mass of the droplets compared to that of the air are relatively high. In the rest of the spray the droplet density is much lower and so is the total mass of the droplets compared to that of the air. This difference will cause a different effect of the air flow and the drag force on the spray in the different regimes. This illustrates the thought that the hydrodynamics of a system with particles is strongly dependent on the number of particles involved. This is for example also seen in a settler.

Summarising, the air flow and thus air velocity in the spray are expected to be important and not negligible as modelled. Furthermore the air hydrodynamics is expected to be different in different stages of the spray and thus the drag force will also be different in those stages. The exact influence of the air flow on $\vec{v}_{air}$ in the drag force in Equation (3.4) is unknown, but expected to be significant. Very complex air flow pattern simulations have to be done simultaneously with the model developed in this work to tackle this problem.

Conclusions 5

During the graduation period a model was developed to simulate the trajectories of the charged droplets in an EHDA spray. From first sight it seems that those trajectories are easily obtained by integrating Newton's equation of motion. But it appears that the collective behaviour of an EHDA spray is complex due to the enormous amount of charged droplets that interact via Coulomb repulsions. In order to circumvent the simulation of the enormous amount of droplets, a scaling correlation between the droplet production time t_{prod} and the droplet Coulomb charge, the real droplet charge multiplied by factor f_q, is developed within the framework of this model. A simulation with the parameters $t_{prod} = 1.25\ 10^{-3}\ s$ and $f_q = 6$ reproduces the experimental spray reasonably well with respect to the deposition pattern.

Measuring the shape and density of the spray, and the droplet deposition pattern on the target and the walls, is important. In that way the extreme positions accompanied with the density of the droplets are known. From these results the dense part of the experimental spray is considered and simulated. The simulation results are compared with the results from the PDPA velocity and density measurements and from the deposition analyses. The verification results are summarized now.

The deposition pattern of the droplets on the cylinder target in both the experimental and the simulated situation is an ellipse. Because the simulated extreme deposition positions in x- and y-direction were used as criterion to choose f_q, they resemble the experimental results well. This is a first success since the droplets are produced in a radially symmetrical circular plane. The shape of the simulated spray itself resembles the experimental results also quite well. Close to the nozzle the spray is too narrow though, and as a result it stays slightly less broad than the experimental spray until deposition on the target. In other words, the simulated spray is shifted downwards a few cm compared to the real spray. The qualitative difference in the density profiles can be attributed to the initial density distribution. The velocity profiles of the simulated and real sprays are qualitatively similar, but a main difference is present. Namely, the magnitude of the velocities of the droplets in both x- and z-direction is much higher in the real spray than in the model spray, despite of the high initial velocities.

It appears the model including the scale-up method gives good results. The main discrepancies that appear from the verification are that the simulated spray is initially too narrow and that the magnitude of the velocity is much less than in the real spray. The main reasons for those discrepancies between the model and the reality are discussed here.

For the developed model the initial conditions of the produced droplets are important and should be measured carefully. It is concluded that measuring these droplet properties

and the initial droplet density cannot be done with a PDPA analysis. The initial droplet properties are thus unknown and assumed by a production of droplets with random position and velocity in a limited range, simulating the jet break up. It appears this assumption works out quite well. The initial distribution is not random but more similar to Gaussian though.

On the basis of a comparison between a very short simulation with many real droplets and a simulation with less model droplets some conclusions are done. The observation that the simulated spray is too narrow close to the nozzle, and as a result also lower in the spray, is assigned to the difference in the Coulomb interactions in that area between the real and model situations. Despite of the scale-up of the Coulomb interactions in the model, the method cannot account for the broadness close to the nozzle caused by the repulsion of the enormous amount of droplets in reality. But the method can account for the broadening of the spray after about 1 cm distance in vertical direction.

The second major difference between the model and the reality is found in the drag force for which the air velocity is needed. The air hydrodynamics in the spray are expected to be important and not negligible as modelled. This is illustrated by the fact that the total mass of the air in the spray is much larger than the total mass of the droplets. Furthermore the air flow is expected to be different in different stages of the spray. This is because the density of the droplets is different throughout the spray and the hydrodynamics are influenced by the number of droplets involved. Close to the nozzle the droplet density is high and the influence of the air flow is expected to be large, because of the high velocity and the turbulence. Further away from the nozzle the spray is less dense and the droplets have a lower velocity. Finally close to the target the air has to "leave" the spray and flow around the target. The (average) magnitude of $\vec{v}_{air}$ in those areas is unknown. Very complex air flow pattern simulations have to be done simultaneously with the model developed in this work to tackle this problem. Implementing the air flow in the model is expected to explain the observed difference in the magnitude of the velocity. Possibly it will also have some influence on the spray close to the nozzle.

From this work it is concluded that the developed model including the scale-up method can simulate the EHDA spray quite well. The developed scale-up correlation was also validated for a different set of initial droplet properties. This leads to practically identical results of a simulation with a small number of droplets and a simulation with a 2 to 16 times larger droplet production time, see Table 4.2. The model is not perfect yet because some important discrepancies compared to the reality are observed. Clearly the model has to be adapted before it can be applied to optimise a process, as outlined in the introduction of this thesis.

Because of these discrepancies the extrapolation of the scale-up correlation towards the experimental spray could neither be validated nor invalidated. If more experimental information is obtained, the model can be developed further. If the discrepancies just discussed are disposed of, the scale-up correlation can be improved and a validation or invalidation of the extrapolation will be possible. It is expected that if the observed difference in the magnitude of the velocity and with that the flux of the droplets, is taken into account for the scale-up, the extrapolation towards the experimental spray will be quite successful. A second reason that the extrapolation is made more difficult

is the error in the experimental value of t_{prod}. A small error in the experimental t_{prod} results in an error in f_q for the final simulation, that influences the simulation results greatly.

Considering the results obtained it can be concluded that the model is valuable. The deposition pattern is reproduced well. The shape of the spray, the density and velocity profiles are also similar, except for the discussed shortcomings. It is expected that if the main discrepancies of the model are tackled, it will be even more valuable. The model will then be able to reproduce the real EHDA. It could be used for other applications subsequently. The only parameter that was adapted is f_q. This parameter is chosen such that the deposition pattern of the simulated spray resembles the real deposition pattern. If the discrepancies are tackled and if the t_{prod} is known accurately, the extrapolation of the scale-up correlation towards the experimental spray could be validated. This would result in a model without free parameters, because the f_q will be fixed then. It would be a fantastic achievement if such a model can be applied to other set-up's using a similar EHDA spray. It might be impossible to validate the scale-up correlation with the experimental data, because it is expected that a totally clear proof is very hard to obtain. This does not mean that the model including the scale-up method is unusable for other purposes. The model will have to be used then with one free variable, f_q.

Recommendations 6

The model presented in this report was improved enormously from its early state, but it is still not perfect. The results are very promising so far and it is thus recommended to develop the model further. In Chapter 1 it is explained that the results obtained with a (model-)simulation should be verified with experimental measurements. This report is set up in this spirit and therefore also some recommendations can be done concerning the experimental verification measurements.

The recommendations for the verification experiments are:

- As much as possible verification measurements should be done, preferably throughout the whole spray creating a three-dimensional image of the spray properties.
- The experimental droplet production time t_{prod} and the initial droplet diameter d should be measured accurately.
- The experimental set-up should resemble the model system set-up as much as possible. The influence of necessary but for the model unimportant equipment must be very low and checked for the used experimental set-up.
- The environment of the measurements must be optimal and constant, for example there should be no airflow due to draft and the spray must be stable and reproducible.

In Section 3.3 the main assumptions of the produced model are listed. One important result of this work is that many assumptions have already been checked and implemented, so that for future work only a reduced set is left. It is recommended for future development of the model that it is attempted to tackle this final set of assumptions and clarify the influence and/or find a solution to it. Some main points are:

- The air velocity needed in the drag force is important to know. A second model for the air flow should be coupled to and should run simultaneously with the present model [9].
- The model should be adapted very close to the nozzle because the spray is too narrow in that area. Possibly with new information this problem can be resolved or another model has to be set up for that area. The influence of evaporation and with that the droplet diameter should be investigated for that area, because it appears that the droplet diameter decreases very fast initially.
- The initial properties of the produced droplets should be known more accurately. Next to the already mentioned initial droplet diameter, the velocity profile and the

density of the distribution at the initial droplet positions should be revealed. In this model a constant total velocity vector is assumed with random distribution between some angles, while the distribution also could be Gaussian.

- The evaporation of the liquid could be taken into account, for example according to References [3] and [12].
- When the droplet size distribution is measured this could be implemented in the model, taking away the assumption of monodispersity.
- The existence and influence of an image force in the model should be investigated. In References [2] and [14] the image force in an electrospray is affirmed and denied respectively. The droplet density at the boundary of these approaches can be investigated and compared to the experimental density close to the target.

The present scale-up correlation is set up for a certain set of parameters within the framework of the model with all its assumptions. It is also validated for a second set of initial droplet properties. If the main points of improvement of the model are tackled first, the model should be able to simulate the EHDA spray very well. Subsequently it can be optimised by improving the scale-up correlation. Taking into account the comment in Section 4.4.1, the accuracy of the correlation could be improved somewhat. After that, the correlation should be checked for validation by extrapolation towards the experimental spray. Whether this succeeds or not, as discussed in the conclusions, the model should be tested for another experimental set-up. The ability of the model to simulate a different EHDA spray will then be checked.

Bibliography

[1] M.P. Allen and D.J. Tildesly. Computer simulation in chemical physics. *(NATO ASI series. Series C. Mathematical and physical sciences vol. 397)*, 1993.

[2] G.S.P. Castle and I.I. Inculet. Space charge effects in orchard spraying. *IEEE Transactions on Industry Applications*, IA-19(3):476–480, 1983.

[3] S.J. Cox, D.W. Salt, B.E. Lee, and M.G. Ford. A model for the capture of aerially sprayed pesticide by barley. *Journal of Wind Engineering and Industrial Aerodynamics*, 87:217 – 230, 2000.

[4] K.B. Geerse, R.P.A. Hartman, J.C.M. Marijnissen, and B. Scarlett. An overview on electrohydrodynamic atomization. *Burgers Centre Annual report*, 1998.

[5] R.P.A. Hartman, J.-P. Borra, D.J. Brunner, J.C.M. Marijnissen, and B. Scarlett. The evolution of electrohydrodynamic sprays produced in the cone-jet mode, a physical model. *Journal of Electrostatics*, 47:143 – 170, 1999.

[6] R.P.A. Hartman, D.J. Brunner, D.M.A. Camelot, J.C.M. Marijnissen, and B. Scarlett. Electrohydrodynamic atomization in the cone-jet mode - physical modeling of the liquid cone and jet. *Journal of Aerosol Science*, 30(7):823 – 849, 1999.

[7] R.P.A. Hartman, D.J. Brunner, D.M.A. Camelot, J.C.M. Marijnissen, and B. Scarlett. Jet break-up in electrohydrodynamic atomization in the cone-jet mode. *Journal of Aerosol Science*, 31(1):65 – 95, 2000.

[8] W.C. Hinds. Aerosol technology, properties behavior, and measurment of airborne particles. page 40, 1982.

[9] K. Hofler and S. Schwarzer. Navier-stokes simulation with constraint forces: finite-difference method for particle-laden flows and complex geometries. *Physical Review E*, 61(6 B):7146 – 7160, 2000.

[10] I. Kondor and J. Kertesz. Advances in computer simulation. 1997.

[11] S. Luding. http://www.dct.tudelft.nl/part/welcomeptg.html.

[12] M.L. Mokeba, D.W. Salt, B.E Lee, and M.G. Ford. Simulating the dynamics of spray droplets in the atmosphere using ballistic and random-walk models combined. *Journal of Wind Engineering and Industrial Aerodynamics*, 67 - 68:923 – 933, 1997.

[13] R. Pitchumani. Simulation of droplet trajectories of electrosprays. *Internal report at Particle Technology Group of Faculty Applied Sciences, Chemical Engineering at TU Delft*, May - July 2000.

[14] C. Vauge. On the concept of "image-force". *Journal of Aerosol Science*, 33:829–832, 2002.

[15] P. Vercoulen. Electrostatic processing of particles. *PhD-thesis at Particle Technology Group of Faculty Applied Sciences, Chemical Engineering at TU Delft*, 1995.

List of Symbols

Symbols

C_D	drag coefficient	$-$
d	diameter	m
D	number of dimensions	$-$
E	(external) electric field	V/m
f	multiplying factor	$-$
F	force	N
I	electric current	A
K	electric conductivity	S/m
m	mass	kg
N	number of droplets	$-$
q	charge	C
r	distance or position	m
Re	Reynolds number	$-$
t	time or time step	s
T	ambient temperature	K
v	velocity	m/s
x	position in x-direction	m
y	position in y-direction	m
z	position in z-direction	m
α	angle	$^\circ$
γ	liquid surface tension	N/m
η	viscosity	Pas
ϕ	liquid volume flow rate	m^3/s
ρ	liquid density	kg/m^3
τ	characteristic time	s

Subscripts

air	concerning the ambient air
$coulomb$	concerning Coulomb interactions
$drop$	concerning the droplets in the EHDA spray
dep	concerning the deposition of the droplets
$drag$	concerning the drag due to movement in air
$flux$	concerning the flux of the droplets
i	concerning droplet i
$initial$	of the droplet at his production time
j	concerning droplet j
ij	between droplet i and droplet j
max	maximum
$model$	concerning the model

$prod$	of production of droplets
q	for the droplet charge concerning the Coulomb charge only
R	concerning the Rayleigh limit
$reality$	concerning the reality
$steady$	concerning the steady state of the simulation
$stop$	concerning the end of the simulation
tot	total
32	Sauter mean

Superscripts

$max-x$	concerning the maximum in x-direction
$max-y$	concerning the maximum in y-direction

Constants

g	gravitation constant	9.81 m/s^2
k	Boltzmann constant	1.38 10^{-23} J/K
ε_0	vacuum permittivity	8.854 10^{-12} C^2/Jm

Mathematical signs

$\vec{}$	vector
$\lvert\vec{}\rvert$	size of a vector
$<\ >$	average

Appendices

A Raw data from PDPA measurements

B Final model in C++

C Input and output files of the C++ program

D Additional C++ programs

E Additional MATLAB programs

F Interpolation algorithm

Raw data from PDPA measurements

A

In Table A-1 the raw data from the measurements done with PDPA analysis, explained in Subsection 2.2.4, are presented.

The first two columns are the position of the measuring point. The third column is the Sauter mean diameter d_{32} in μm. The fourth and fifth column show the mean velocity and root mean square of the distribution of the velocity in x-direction. The sixth and seventh column are equal to the former two, but for the z-direction. The next two columns give the number of droplets that are analysed resulting in the velocity data in both directions, as also explained in Subection 2.2.4. The last two columns are the measuring times in both directions, which are almost the same.

Table A-1: Raw PDPA data made with TSI Aerometrics software.

x (m)	z (m)	d_{32} (μm)	$< v_x >$ (m/s)	$rms\ v_x$ (m/s)	$< v_z >$ (m/s)	$rms\ v_z$ (m/s)	N_x $(\#)$	N_z $(\#)$	t_x (s)	t_z (s)
0	-0.19815	7.55471	-0.0413884	0.227077	-2.66582	0.62798	9903	3251	155.654	155.628
0	-0.18815	7.43478	-0.0631786	0.221497	-2.45007	0.562267	9943	2874	134.293	134.261
0	-0.17815	7.22623	-0.0613167	0.209082	-2.3847	0.530636	9950	1806	108.018	107.916
0	-0.16815	7.29539	-0.0631465	0.221666	-2.43513	0.536696	9947	832	91.9366	91.775
0	-0.15815	7.52601	-0.0599675	0.231084	-2.00388	0.55694	9916	5096	160.981	160.849
0	-0.14815	7.73936	-0.0525934	0.234757	-2.3418	0.505181	9955	1073	76.8972	76.8692
0	-0.13815	7.96878	-0.0582554	0.231736	-2.44908	0.505804	9948	725	61.754	61.7102
0	-0.12815	8.39865	-0.0502534	0.240629	-2.50794	0.465506	9941	866	54.6921	54.428
0	-0.11815	8.72395	-0.0503238	0.240638	-2.70731	0.460138	9944	531	42.629	42.5338
0	-0.10815	8.2002	-0.0545666	0.218986	-2.12624	0.542691	9950	3412	62.1305	62.0737
0	-0.09815	9.34235	-0.0343049	0.256405	-3.03589	0.47807	9935	346	28.232	28.0618
0	-0.08815	9.26672	-0.041688	0.257383	-3.0899	0.511517	9945	1124	21.7781	21.761
0	-0.07815	8.98356	-0.0297022	0.263238	-3.31912	0.51494	9944	1402	15.2724	15.2676
0	-0.06815	9.40973	-0.0127485	0.259639	-3.69894	0.571839	9955	1506	10.3152	10.2919
0	-0.05815	8.44716	0.0137413	0.259037	-3.96317	0.605869	9971	1817	7.18346	7.18097

x	z	d_{32}	$< v_x >$	$rms\ v_x$	$< v_z >$	$rms\ v_z$	N_x	N_z	t_x	t_z
(m)	(m)	(μm)	(m/s)	(m/s)	(m/s)	(m/s)	$(\#)$	$(\#)$	(s)	(s)
0	-0.04815	8.91199	0.00161842	0.245396	-4.26166	0.758459	9972	2048	5.27941	5.27822
0	-0.03815	8.48251	-0.0865191	0.226896	-4.87993	0.790759	9975	4048	2.38189	2.38114
-0.01	-0.19815	6.72338	-0.0907963	0.21274	-3.07685	0.648078	9942	605	155.424	154.609
-0.01	-0.18815	7.39802	-0.10636	0.198908	-2.68852	0.690566	9920	3156	165.919	165.879
-0.01	-0.17815	6.97347	-0.113909	0.208611	-2.66538	0.646368	9934	1656	143.944	143.83
-0.01	-0.16815	6.57058	-0.125941	0.20869	-2.0665	0.581664	9926	4106	122.264	122.255
-0.01	-0.15815	7.10525	-0.134658	0.211801	-2.34529	0.576176	9937	2624	113.761	113.737
-0.01	-0.14815	7.29098	-0.151472	0.220186	-2.36022	0.572769	9913	2162	104.731	104.632
-0.01	-0.13815	7.52019	-0.166605	0.226968	-2.3005	0.546008	9899	3564	95.0022	94.9385
-0.01	-0.12815	7.96825	-0.187489	0.231207	-2.36295	0.546865	9888	3105	102.312	102.208
-0.01	-0.11815	7.93139	-0.21148	0.242608	-2.39321	0.563208	9869	2722	93.52	93.4227
-0.01	-0.10815	7.76239	-0.246004	0.248486	-2.38221	0.552235	9897	3496	77.9055	77.9055
-0.01	-0.09815	8.09513	-0.275716	0.254474	-2.54702	0.540645	9885	3243	60.791	60.7581
-0.01	-0.08815	8.37925	-0.339979	0.267931	-2.57145	0.541532	9887	4365	45.673	45.6503
-0.01	-0.07815	8.6018	-0.404384	0.261787	-2.60822	0.518758	9874	4867	39.6289	39.6289
-0.01	-0.06815	8.79201	-0.526411	0.257385	-2.77152	0.504424	9879	4538	25.9774	25.9687
-0.01	-0.05815	9.11462	-0.66332	0.251357	-2.86783	0.491741	9893	3810	20.3659	20.3289
-0.01	-0.04815	9.15052	-0.933267	0.259943	-2.95107	0.482094	9954	5056	12.2491	12.2467
-0.01	-0.03815	9.47112	-1.25875	0.304389	-3.20811	0.56805	9966	4610	8.45144	8.45145
-0.01	-0.02815	10.1024	-2.39766	0.419672	-4.20917	0.623561	9935	6080	5.18508	5.18495
-0.03	-0.19815	7.02313	-0.159594	0.219729	-3.18271	0.601399	9865	296	185.571	184.869
-0.03	-0.18815	7.00144	-0.142224	0.195607	-2.70111	0.510303	9924	340	196.346	196.124
-0.03	-0.17815	6.89495	-0.166078	0.209837	-2.61709	0.474056	9909	279	202.774	201.989
-0.03	-0.16815	7.03909	-0.186848	0.214168	-2.24223	0.49419	9924	553	180.293	180.069
-0.03	-0.15815	7.30628	-0.234251	0.2302	-2.00491	0.456612	9924	989	166.185	165.748
-0.03	-0.14815	7.39531	-0.29863	0.259394	-2.04621	0.437974	9862	614	170.753	170.363
-0.03	-0.13815	7.51921	-0.358237	0.267896	-1.97955	0.452443	9886	847	146.924	146.578
-0.03	-0.12815	8.02822	-0.43444	0.283941	-1.88014	0.439813	9854	1543	163.77	163.566
-0.03	-0.11815	7.91295	-0.531025	0.281711	-1.91731	0.420136	9865	1466	150.771	150.629
-0.03	-0.10815	8.23103	-0.659041	0.30724	-1.84817	0.40065	9817	2829	146.554	146.467
-0.03	-0.09815	8.08986	-0.78363	0.302814	-1.91662	0.398192	9871	2533	115.953	115.896
-0.03	-0.08815	8.14508	-0.932076	0.311006	-1.80342	0.40876	9880	7903	129.903	129.901
-0.03	-0.07815	8.10711	-1.08651	0.319313	-1.94273	0.379065	9968	4041	66.9773	66.9282
-0.03	-0.06815	8.05361	-1.28676	0.33954	-1.98221	0.369068	9979	5029	56.6129	56.5956

x	z	d_{32}	$< v_x >$	$rms\ v_x$	$< v_z >$	$rms\ v_z$	N_x	N_z	t_x	t_z
(m)	(m)	(μm)	(m/s)	(m/s)	(m/s)	(m/s)	$(\#)$	$(\#)$	(s)	(s)
-0.03	-0.05815	8.70713	-1.78504	0.364672	-2.19138	0.325154	9966	6119	48.4765	48.4704
-0.05	-0.19815	7.36016	-0.163566	0.199518	-2.37465	0.615398	9934	2578	241.388	241.052
-0.05	-0.18815	7.44608	-0.182505	0.206609	-2.11404	0.539297	9915	2866	255.648	255.648
-0.05	-0.17815	7.47698	-0.221228	0.244968	-1.81177	0.498341	9813	6657	340.113	340.022
-0.05	-0.16815	7.50582	-0.308791	0.261576	-1.78328	0.45579	9878	2809	220.961	220.913
-0.05	-0.15815	7.948	-0.383693	0.276872	-1.63217	0.437282	9864	4780	219.682	219.652
-0.05	-0.14815	7.68773	-0.465453	0.280061	-1.58453	0.430559	9821	4492	205.367	205.367
-0.05	-0.13815	7.89839	-0.566778	0.298131	-1.54606	0.405428	9828	4777	190.648	190.604
-0.05	-0.12815	7.86121	-0.644238	0.302591	-1.53008	0.386501	9862	4052	185.162	185.125
-0.05	-0.11815	8.02545	-0.755012	0.322162	-1.53659	0.382604	9902	4398	168.504	168.504
-0.05	-0.10815	7.85939	-0.892627	0.329659	-1.52937	0.377337	9887	4806	148.81	148.722
-0.05	-0.09815	7.80307	-1.27854	0.377892	-1.64183	0.320001	875	455	10.8175	10.7881
-0.05	-0.08815	7.4744	-1.30804	0.35914	-1.64535	0.334785	9970	5442	125.353	125.329
-0.05	-0.07815	7.23172	-1.70147	0.427506	-1.79473	0.32523	9963	5395	131.818	131.806
-0.07	-0.19815	6.46387	-0.188389	0.213069	-2.17905	0.583237	9917	5226	345.338	345.317
-0.07	-0.18815	6.96208	-0.210496	0.223566	-2.02701	0.51022	9891	3796	384.289	384.093
-0.07	-0.17815	7.72062	-0.278164	0.271345	-1.74244	0.481082	9791	4423	343.58	343.505
-0.07	-0.16815	7.16875	-0.382772	0.289335	-1.62952	0.446888	9800	4543	339.24	339.201
-0.07	-0.15815	7.51745	-0.49187	0.300899	-1.52153	0.407657	9698	5780	449.337	449.141
-0.07	-0.14815	7.57258	-0.591775	0.314857	-1.472	0.380406	9720	5294	439.667	439.833
-0.07	-0.13815	7.48605	-0.704682	0.320705	-1.43542	0.371124	9809	4717	318.62	318.502
-0.07	-0.12815	7.58289	-0.819201	0.331146	-1.39817	0.348317	9850	4497	268.292	268.242
-0.07	-0.11815	7.17345	-0.929354	0.339808	-1.35128	0.334995	9884	5009	253.194	253.163
-0.07	-0.10815	7.41407	-1.10678	0.360257	-1.40051	0.321671	9935	3942	208.802	208.732
-0.07	-0.09815	7.02351	-1.35628	0.422536	-1.41717	0.33488	9961	5754	249.222	249.172
-0.07	-0.08815	7.02705	-1.47469	0.442513	-1.29278	0.332922	9958	5207	371.171	371.109
0	-0.02815	9.66047	-0.0375865	0.208131	-5.36426	1.114	9901	8722	1.71316	1.71281
0	-0.01815	10.4211	0.0585994	0.22074	-6.17946	1.62714	9956	8661	1.00152	1.00142
0	-0.00915	22.3315	-3.3773	1.34318	-16.1299	0.801969	8452	4442	0.517495	0.51734
-0.09	-0.19815	8.09646	-0.24029	0.24369	-3.34836	0.454751	9867	303	471.419	469.972
-0.09	-0.18815	8.22917	-0.305379	0.261226	-2.07574	0.584801	9790	7396	612.747	612.616
-0.09	-0.17815	7.73663	-0.466636	0.295428	-1.9443	0.517575	9704	9245	940.081	939.974
-0.09	-0.16815	7.85807	-0.617295	0.303747	-1.82986	0.44297	7987	9491	1539.73	1540.21
-0.09	-0.15815	8.27951	-0.762128	0.31116	-1.67933	0.402019	9842	6920	1135.11	1135.11

x (m)	z (m)	d_{32} (μm)	$< v_x >$ (m/s)	$rms\ v_x$ (m/s)	$< v_z >$ (m/s)	$rms\ v_z$ (m/s)	N_x $(\#)$	N_z $(\#)$	t_x (s)	t_z (s)
-0.09	-0.14815	8.74816	-0.887808	0.306897	-1.61917	0.372547	9893	5268	1011.35	1011.26
-0.09	-0.13815	8.18836	-1.10825	0.347695	-1.65539	0.372846	9935	3987	1153.66	1153.73
-0.11	-0.19815	7.72368	-0.235186	0.256265	-2.17882	0.43391	9772	7964	1883.3	1883.25
-0.11	-0.18815	7.78244	-0.31604	0.276688	-1.8743	0.526194	9778	7655	1590.41	1590.33
-0.11	-0.17815	8.05813	-0.463345	0.301996	-1.74626	0.518181	9732	6395	1746.09	1745.6
-0.11	-0.16815	8.27418	-0.627883	0.287836	-1.33368	0.411452	9793	6695	1063.36	1063.35
-0.11	-0.15815	8.11606	-0.745755	0.314712	-1.21748	0.315996	9842	5557	983.895	983.836
-0.11	-0.14815	8.51471	-0.920409	0.303953	-1.19642	0.252756	9871	5761	1965.39	1265.01
-0.11	-0.13815	8.26361	-1.10238	0.360267	-1.01139	0.303426	9894	5664	1345.76	1345.28
-0.11	-0.12815	7.96912	-1.20214	0.318755	-0.827511	0.262926	6968	9482	1783.77	1783.77
-0.09	-0.12815	7.91174	-1.29987	0.359522	-1.33919	0.396836	9063	9464	848.245	848.225
-0.09	-0.11815	7.59564	-1.43407	0.391953	-1.16493	0.374222	7302	9378	960.003	960.004
-0.09	-0.10815	7.77368	-1.47214	0.389927	-1.10786	0.268909	9956	6663	1011.1	1011.11
-0.13	-0.19815	7.63757	-0.0960404	0.235782	-1.94541	0.290646	9905	2925	1761.28	1760.24
-0.13	-0.18815	7.95221	-0.26085	0.276724	-1.22912	0.349502	7561	9372	3101.09	3101.32
-0.13	-0.17815	8.13405	-0.477834	0.302225	-1.00937	0.296839	9638	7738	1426.78	1426.72
-0.13	-0.16815	8.45876	-0.63354	0.300644	-0.926968	0.224874	9761	4270	873.354	873.295
-0.13	-0.15815	8.83065	-0.808136	0.3157	-0.72954	0.243207	9764	9176	1566.28	1566.28
-0.13	-0.14815	8.51585	-0.949501	0.32765	-0.661494	0.251118	6963	9450	2551.78	2552.26
-0.15	-0.16815	8.55151	-0.693475	0.345849	-0.546631	0.22099	4977	4781	3066.19	3066.19
-0.15	-0.17815	8.34672	-0.569994	0.363396	-0.615143	0.231856	4197	9444	2801.08	2804.71
-0.15	-0.18815	9.16941	-0.297208	0.3589	-0.69806	0.236856	2242	9377	3570.1	3570.1
-0.15	-0.19815	10.0528	-0.0633122	0.283322	-0.635808	0.261729	3233	9277	2671.06	2676.93
-0.15	-0.20815	9.351	0.300682	0.289237	-0.497004	0.234436	664	9218	2446.02	2544.92
-0.15	-0.21815	8.81267	0.368111	0.332961	-0.261324	0.1575	588	9188	2521.88	2524.81
-0.15	-0.22815	8.13639	0.365698	0.411667	-0.150288	0.109934	166	9272	2249.92	3035.4

Final model in C++ B

In this appendix the model translated into the C++ language is shown. A simulation is run by running the program `exvthlc.exe`.

exvthlc.exe

The program `exvthlc.exe` is made by compiling the files `exvthlc.cc` and `exvthlc.h` with the line in `exvthlc.bat`.

exvthlc.cc

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>

#include <exvthlc.h>
#include <exvhlc.h>

int main()
{
time_t now;
time(&now); // reads the time at this moment

ierr=get_parameters(); // read the parameters from par.ini
get_electricfield(); // read the electric field from efield.ini

srand( 540 ); // Call only ONE time for initialization of random number production!!!

if(ierr==0)
{
N=get_particles(); // read the particle information from c2d.ini
if( N==0 ) tstop=-1;
else
{
cout << "# Read successfully " << N << " particles \n ";
pp_initialize(); // initialize the particle fields
walls_initialize(); // initialize the walls
compute_parameters();
}
}
else
tstop=-1;
```

```
cout << "# Linked cell procedure started \n"; // start linked cells procedures
lcell_parameters( wx0, wy0, wz0, wx1, wy1, wz1 );

for( int ih=0; ih<H_LEVEL; ih++ )
{
lcell_numcells( ih ); // compute and check cell-sizes etc.
lcell_sort( ih, Nelecfld, xef, yef, zef); // sort gridpoints to lcells
// this is already done in the lcell_sort function
// lcell_count_neighbors( ih );
}

for( ttime=tstart; ttime<=tstop; ttime+=dt ) // main loop with time-step dt
{
deposition(); // remove deposited droplets
dropproduction(); // production of a droplet after certain timestep given in par.ini
loop_init(); // initialize the loop variables
get_hlcell_grid(); // get the distances and electric field components of the four grid points closest to each
particle using the lcell structure
md_xforces(); // account for external forces
md_pforces(); // compute particle-particle forces
//md_wforces(); // compute wall-particle forces; in this case put in subroutine deposition()

if( ttime>=tlogout-ACCURA ) // output to logfile and screen
{
tlogout += dtlogout;
cout << "# logtime: " << ttime << ' ' << N << endl;
enelogout();
}

if( ttime>=tfilmout-ACCURA ) // output to logfile and screen
{
tfilmout += dtfilmout;
cout << "# filmtime: " << ttime << endl;
c2dfilmout();
}

md_integrate(); // do the integration (Verlet)
//evaporation(); // evaporation of the droplets; not yet in the program!!
}

time_t end;
time(&end);
comment<<endl;
comment<<"The initial time is "<< asctime(localtime(&now))<<endl; // put in comment file initial
time
comment<<"The final time is "<< asctime(localtime(&end))<<endl; // put in comment file final time
cout << "The initial time is " << asctime(localtime(&now)); // put on screen initial time
cout << "The final time is " << asctime(localtime(&end)); // put on screen final time

deposit << Ndep <<' '<<tstart<<' '<< ttime-dt <<' '<< wx0 <<' '<< wx1 <<' '<< wy0 <<'
'<< wy1 <<' '<< wz0 <<' '<< wz1 <<endl;
// final line in the deposit - output file -> must be set afterwards as the first line!!!!
wall << Nwall <<' '<<tstart<<' '<< ttime-dt <<' '<< wx0 <<' '<< wx1 <<' '<< wy0 <<' '<<
```

```
wy1 <<' '<< wz0 <<' '<< wz1 <<endl;
// final line in the wall - output file -> must be set afterwards as the first line!!!!

close_files();

cout << "# End of program ... " << x[0] <<' '<< y[0] <<' '<< z[0] <<endl;

if( ttime>=tstop )
return 0;
else
return ierr;
}

// external forces subroutine (gravitational, drag and electrostatic forces)
void md_xforces()
{
double prefactor1, prefactor2, prefactor3, prefactor4, prefactor5, Re_sq; // constants needed for drag
force calculation
double Exfield, Eyfield, Ezfield; // electric field components in each direction
double charge; // charge of one droplet

double g0x, g0y, g0z;
double E0x, E0y, E0z;

double dist_km, E_km, gradx, grady, gradz;
double weightx, weighty, weightz;
double totweightx, totweighty, totweightz;

int j[JNMAX], j1;

prefactor1 = rho_air*rho_air*4/(mu*mu);
prefactor3 = 6*M_PI*mu;

// interpolation between Stokes and Klyachko for 1 < Re < 2: C_D = -8.8252*(Re-1)+24
// 'factor' is the factor between the C_D in Stokes regime and in the regime 1 < Re < 2
// factor = C_D (inter-regime) / C_D (Stokes)
// factor = -0.36771658*Re_sq + 1.36771658*Re

for (int i=0; i<N; i++)
{
// gravitation
fz[i] += gravz*mass[i];

// drag
prefactor2 = prefactor1*rad[i]*rad[i];
Vel = (vx[i]*vx[i] + vy[i]*vy[i] + vz[i]*vz[i]);
Re_sq = prefactor2*Vel;

if(Re_sq <= 1) // Re < 1 : Stokes regime
{
fx[i] -= prefactor3*rad[i]*vx[i];
fy[i] -= prefactor3*rad[i]*vy[i];
fz[i] -= prefactor3*rad[i]*vz[i];
}
```

```
else if((Re_sq > 1)&&(Re_sq < 4))
{
prefactor5 = -0.36771658*Re_sq + 1.36771658*sqrt(Re_sq);
fx[i] -=prefactor3*rad[i]*vx[i]*prefactor5; // 1 < Re > 2 : interpolation between Stokes and Klyachko.
fy[i] -=prefactor3*rad[i]*vy[i]*prefactor5;
fz[i] -=prefactor3*rad[i]*vz[i]*prefactor5;
}

else
{
prefactor4 = (1 + pow(Re_sq, 1./3)/6);
fx[i] -=prefactor3*rad[i]*vx[i]*prefactor4; // 2 < Re < 800 : according to Klyachko (also W.C. Hinds
(1982), page 40.)
fy[i] -=prefactor3*rad[i]*vy[i]*prefactor4;
fz[i] -=prefactor3*rad[i]*vz[i]*prefactor4;
}

// external electric field

g0x=0;
g0y=0;
g0z=0;
E0x=0;
E0y=0;
E0z=0;

for( j1=0; j1<JNMAX; j1++ )
{
j[j1]=jgrid[i][j1];
// compute the center of the JNMAX nearest gridpoints
g0x += xef[j[j1]];
g0y += yef[j[j1]];
g0z += zef[j[j1]];
// compute the mean electric field there
E0x += Ex[j[j1]];
E0y += Ey[j[j1]];
E0z += Ez[j[j1]];
}

g0x=g0x/JNMAX; // mean values!!
g0y=g0y/JNMAX;
g0z=g0z/JNMAX;

E0x=E0x/JNMAX; // mean values!!
E0y=E0y/JNMAX;
E0z=E0z/JNMAX;

gradx = 0; // initialise
grady = 0; // initialise
gradz = 0; // initialise
totweightx = 0; // initialise
totweighty = 0; // initialise
totweightz = 0; // initialise
```

```
for(int k=0; k<JNMAX; k++) // calculate gradients between gridpoints k and m
{
j[k]=jgrid[i][k];

for(int m=k+1; m<JNMAX; m++)
{
j[m]=jgrid[i][m];

dist_km=sqrt((xef[j[k]]-xef[j[m]])*(xef[j[k]]-xef[j[m]])+(yef[j[k]]-yef[j[m]])*(yef[j[k]]-yef[j[m]])+(zef[j[k]]-zef[j[m]])*(zef[j[k]]-
zef[j[m]]));%
E_km=sqrt((Ex[j[k]]-Ex[j[m]])*(Ex[j[k]]-Ex[j[m]])+(Ey[j[k]]-Ey[j[m]])*(Ey[j[k]]-Ey[j[m]])+(Ez[j[k]]-Ez[j[m]])*(Ez[j[k]]-
Ez[j[m]]));

weightx = (xef[j[k]]-xef[j[m]])/dist_km;
gradx += (Ex[j[k]]-Ex[j[m]])/dist_km*weightx*pow(weightx,4);

if(weightx<0) weightx = -weightx;
totweightx = totweightx + weightx;

weighty = (yef[j[k]]-yef[j[m]])/dist_km;
grady += (Ey[j[k]]-Ey[j[m]])/dist_km*weighty*pow(weighty,4);

if(weighty<0) weighty = -weighty;
totweighty = totweighty + weighty;

weightz = (zef[j[k]]-zef[j[m]])/dist_km;
gradz += (Ez[j[k]]-Ez[j[m]])/dist_km*weightz*pow(weightz,4);

if(weightz<0) weightz = -weightz;
totweightz = totweightz + weightz;
}
}

if(totweightx != 0)
{
Exfield = E0x + gradx/totweightx*(x[i]-g0x);
}
else
{
Exfield = E0x;
}

if(totweighty != 0)
{
Eyfield = E0y + grady/totweighty*(y[i]-g0y);
}
else
{
Eyfield = E0y;
}

if(totweightz != 0)
{
```

```
Ezfield = E0z + gradz/totweightz*(z[i]-g0z);
}
else
{
Ezfield = E0z;
}

charge = effrayl*qparam*2*rad[i]*sqrt(2*rad[i]);

fx[i] += charge*Exfield;
fy[i] += charge*Eyfield;
fz[i] += charge*Ezfield;
}
}

// particle particle interaction subroutine
// The simplest version without any neigbor search
void md_pforces()
{
double d0, delta, d_vec;
double nx, ny, nz, mx, my, mz;
double charge, prefactor;

// the coulomb_charge_factor is a factor that increases the charge of the droplets and thus increases
the Coulomb repulsion effect. The goal of this is to compensate for the relative low number of droplets
in the system comparing to reality. So this adaption is NOT done for the external electric field charge!

prefactor = 2*coulomb_charge_factor*coulomb_charge_factor*effrayl*effrayl*qparam*qparam/(M_PI*epsilon);

for( int i=0; i<N; i++ )
{
for( int j=i+1; j<N; j++ )
{
d0=(x[i]-x[j])*(x[i]-x[j]) +(y[i]-y[j])*(y[i]-y[j]) +(z[i]-z[j])*(z[i]-z[j]);

// Coulomb force always
d_vec = (d0*sqrt(d0));
mx=(x[i]-x[j])/d_vec; // compute relative x-distance between p i and j
my=(y[i]-y[j])/d_vec; // compute relative y-distance between p i and j
mz=(z[i]-z[j])/d_vec; // compute relative z-distance between p i and j

// ATTENTION!!!!!
// this force calculation is done for droplets of equal size and equal charge!!!
// for different sizes and charges the force should be changed to:
// f_ij = 1/(4*M_PI*epsilon)*q_i*q_j*m_ij, with q_drop = q_param*effrayl*(2*rad_drop)Î.5

fx[i] += prefactor*rad[i]*rad[i]*rad[i]*mx;
fy[i] += prefactor*rad[i]*rad[i]*rad[i]*my;
fz[i] += prefactor*rad[i]*rad[i]*rad[i]*mz;

fx[j] -= prefactor*rad[j]*rad[j]*rad[j]*mx;
fy[j] -= prefactor*rad[j]*rad[j]*rad[j]*my;
fz[j] -= prefactor*rad[j]*rad[j]*rad[j]*mz;
```

```
// repulsive force on contact
if( d0 < (rad[i]+rad[j])*(rad[i]+rad[j]) )
{
d0=sqrt(d0); // distance d0=—r_i-r_j—
delta=-d0+rad[i]+rad[j]; // overlap delta

nx=(x[i]-x[j])/d0; // compute contact normal
ny=(y[i]-y[j])/d0;
nz=(z[i]-z[j])/d0;

fx[i] += K*delta*nx; // acceleration of p i due to contact with j
fy[i] += K*delta*ny;
fz[i] += K*delta*nz;

fx[j] -= K*delta*nx; // acceleration of p j due to contact with i : negative of fx[i]
fy[j] -= K*delta*ny;
fz[j] -= K*delta*nz;

comment<<"droplet "<<i<<" bounces with particle "<<j<<" at time is "<<ttime<<endl;
}
}
}
}

// MD integration step
// Here Verlet is used
void md_integrate()
{
double sxtmp, sytmp, sztmp;
double accx, accy, accz;

for( int i=0; i<N; i++ )
{
sxtmp=x[i]; // temporary present values
sytmp=y[i];
sztmp=z[i];
accx=fx[i]/mass[i];
accy=fy[i]/mass[i];
accz=fz[i]/mass[i];
x[i]=sxtmp*2-sx[i]+accx*dt2; // Verlet step
y[i]=sytmp*2-sy[i]+accy*dt2;
z[i]=sztmp*2-sz[i]+accz*dt2;
vx[i]=(x[i]-sx[i])/(dt*2)+accx*dt; // velocities interpolation
vy[i]=(y[i]-sy[i])/(dt*2)+accy*dt; // THIS IS UNSAFE WITH VERLET
vz[i]=(z[i]-sz[i])/(dt*2)+accz*dt;
sx[i]=sxtmp; // define old=present coords.
sy[i]=sytmp;
sz[i]=sztmp;
}
}

// Initialize force etc for each loop
void loop_init()
{
```

```
enepot=0;
for( int i=0; i<N; i++ )
{
fx[i]=0;
fy[i]=0;
fz[i]=0;
}
}

// Read particle coordinates from "c2d.ini"
int get_particles()
{
c2dini >> N >> tstart >> wx0 >> wy0 >> wz0 >> wx1 >> wy1 >> wz1;
cout << "N = " << N << ' ' << "tstart = " << tstart << '\n';

if( N < Nmax )
{
for( int i=0; i<N; i++ )
{
c2dini >> x[i] >> y[i] >> z[i]
>> vx[i] >> vy[i] >> vz[i]
>> rad[i] >> xinfo[i];
cout <<i<<' '<<x[i]<<' '<<y[i]<<' '<<z[i]<<' '<<rad[i]<<'\n';
}
}
else
{
cout << "# ERROR - N is larger than Nmax ... " << N << " > " << Nmax;
}
c2dini.close();

tlogout=tstart;
tfilmout=tstart;
tdropprod=tstart+dtdropprod;

if( N <= 0 ) return 0;
return N;
}

// Initialize particles
void pp_initialize()
{
for( int i=0; i<N; i++ )
{
sx[i]=x[i]-dt*vx[i];
sy[i]=y[i]-dt*vy[i];
sz[i]=z[i]-dt*vz[i];
mass[i]=4./3.*M_PI*rad[i]*rad[i]*rad[i]*mdensity;
}
}

// Initialize walls
void walls_initialize()
{
```

```
// define a point inside the walls
// and the normal vector of the wall

wx[0]=wx0; // left wall
wy[0]=wy0;
wz[0]=wz0;
nwx[0]=1; // normal right
nwy[0]=0;
nwz[0]=0;

wx[1]=wx1; // right wall
wy[1]=wy0;
wz[1]=wz0;
nwx[1]=-1; // normal left
nwy[1]=0;
nwz[1]=0;

wx[2]=wx0; // rear wall
wy[2]=wy0;
wz[2]=wz0;
nwx[2]=0; // normal backward
nwy[2]=1;
nwz[2]=0;

wx[3]=wx0; // front wall
wy[3]=wy1;
wz[3]=wz0;
nwx[3]=0; // normal forward
nwy[3]=-1;
nwz[3]=0;

wx[4]=wx0; // bottom wall
wy[4]=wy0;
wz[4]=wz0;
nwx[4]=0; // normal up
nwy[4]=0;
nwz[4]=1;

wx[5]=wx0; // top wall
wy[5]=wy0;
wz[5]=wz1;
nwx[5]=0; // normal down
nwy[5]=0;
nwz[5]=-1;
}

// Get parameters from file "par.ini"
int get_parameters()
{
parini >> tstop >> dt;
dt2=dt*dt;
tstop += ACCURA;
parini >> dtlogout >> dtfilmout >> dtdropprod;
parini >> K >> V;
```

```
parini >> mdensity >> qparam >> effrayl >> coulomb_charge_factor;
parini >> mu >> rho_air >> epsilon;
parini.close();

comment<< "dtlogout " << "dtfilmout " << "dtdropprod " << " coulomb_charge_factor"<<endl;
comment<< dtlogout <<' '<< dtfilmout <<' '<< dtdropprod <<' '<< coulomb_charge_factor<<endl;

cout << "# Read parameter-file \n";
cout << "# time parameters: " << tstop << ' ' << dt << ' ' << dtlogout << ' ' << dtfilmout <<
' ' << dtdropprod << '\n';
cout << "# coll. properties: "<<K<<' '<<V<< '\n';
cout << "# drop properties: dens="<<mdensity<<" q="<<qparam<<" eff="<<effrayl<<
" coulomb_charge_factor="<<coulomb_charge_factor<< '\n';
cout << "# air properties: visc="<<mu<<" dens_air="<<rho_air<<" eps="<<epsilon<<'\n';
return 0;
}

// Get electric field from "efield.ini"
int get_electricfield()
{
fieldini >> Nelecfld;

if( Nelecfld > Nfldmax )
{
cout << "# ERROR - electri field too large: " << Nelecfld;
Nelecfld = Nfldmax;
}

for( int i=0; i<Nelecfld; i++ ) // Nelecfld is the length of the columns in field.ini
{
fieldini >> xef[i] >> yef[i] >> zef[i] >> Ex[i] >> Ey[i] >> Ez[i];
}
fieldini.close();
cout << "# Read electric field-file \n";
return 0;
}

// Compute some parameters for cross-checking
void compute_parameters()
{
double m_12=mass[0]/2;
double eta=V/m_12;
double tc=M_PI/sqrt(K/m_12-eta*eta);
double rn=exp(-eta*tc);
cout << "# tc=" << tc << " rn=" << rn << '\n';
}

// Output log-data to "ene"
void enelogout()
{
double xhcm=0, yhcm=0, zhcm=0;
double vxcm=0, vycm=0, vzcm=0;
double enekin=0, enerot=0;
double mtot=0;
```

```
for( int i=0; i<N; i++ )
{
xhcm=xhcm+x[i];
yhcm=yhcm+y[i];
zhcm=zhcm+z[i];
vxcm=vxcm+vx[i];
vycm=vycm+vy[i];
vzcm=vzcm+vz[i];
mtot=mtot+mass[i];
enekin+=0.5*mass[i]*(vx[i]*vx[i]+vy[i]*vy[i]+vz[i]*vz[i]);
}
xhcm=xhcm/N;
yhcm=yhcm/N;
zhcm=zhcm/N;
vxcm=vxcm/N;
vycm=vycm/N;
vzcm=vzcm/N;
enekin=enekin/N;
enerot=enerot/N;

eneout << ttime << ' ';
eneout << zhcm*mtot*gravz << ' ';
eneout << enekin << ' ';
eneout << enerot << ' ';
eneout << enepot << ' ';
eneout << " 0 ";
eneout << xhcm << ' ';
eneout << vxcm << ' ';
eneout << vycm << ' ';
eneout << endl;
}

// Output coordinates to "c2d" - an xb8 format file
void c2dfilmout()
{
c2dout << N << ' ' << ttime << ' ' << wx0 << ' ' << wy0 << ' ' << wz0 << ' ' << wx1 << ' '
<< wy1 << ' ' << wz1 << endl;

for( int i=0; i<N; i++)
{
gnuout << N << ' ' << ttime << ' '<< x[i] <<' '<< y[i] <<' '<< z[i] <<' '<< vx[i] <<' '<< vy[i]
<<' '<< vz[i] <<' '<< rad[i] <<' '<< xinfo[i] << endl;

c2dout << x[i] <<' '<< y[i] <<' '<< z[i] <<' '<< vx[i] <<' '<< vy[i] <<' '<< vz[i] <<' '<<
rad[i] <<' '<< xinfo[i] << endl;
}
}

// close files ene and c2d
void close_files()
{
eneout.close();
c2dout.close();
```

```
gnuout.close();
comment.close();
deposit.close();
}

// introduce lcell grid
void get_hlcell_grid()
{
double dist;
int icx0, icy0, icz0;
int icx1, icy1, icz1;
int itmp, itmp0, icheck;
int inmin, inmax;
int j;

int ih;

for (int i=0; i<N; i++)
{
for( j=0; j<JNMAX; j++ )
{
grid[i][j] = 2e20; // initialise the distances in 'grid' at a high value
jgrid[i][j] = -1; // initialize the neighbor grid numbers
}

icheck=0;
ih=0; // try highest resolution level 0

itmp0 = lcell_number( ih, x[i], y[i], z[i] ); // obtain the number of the cell where the particle sits
while(( h[ih].lcell_neighbors[itmp0] < FIELD_MIN ) && ( ih < H_LEVEL-1 ))
{
ih+=1; // try next level
itmp0 = lcell_number( ih, x[i], y[i], z[i] );
}

lcell_index( ih, itmp0, &icx0, &icy0, &icz0 ); // get indices of this core cell

j=h[ih].lcell_member[ itmp0 ];
while( j >= 0 ) // continue as long as j points to a gridpoint
{
icheck+=1;
getgg(i,j); // sort in this gridpoint
j = lcell_pnext( ih, j ); // get next gridpoint in cell
}

// run through all neighbor cells (27 in 3D)
// cout << h[ih].num_neighbors << ' ';
for( int neighbor=1; neighbor<h[ih].num_neighbors; neighbor++ )
{
lcell_inext( ih, neighbor, icx0, icy0, icz0, &icx1, &icy1, &icz1 ); // get indices of neighbor cell
itmp=lcell_p_inumber( ih, icx1, icy1, icz1 ); // get number of this cell

j=h[ih].lcell_member[ itmp ];
```

```
while( j >= 0 ) // continue as long as j points to a gridpoint
{
icheck+=1;
getgg(i,j); // sort in this gridpoint
j = lcell_pnext( ih, j ); // get next gridpoint in cell
}
}
if( icheck < inmin ) inmin=icheck;
if( icheck > inmax ) inmax=icheck;

if( icheck != h[ih].lcell_neighbors[itmp0] )
cout << "# WARNING: num_neigbors missmatch " << ih << "H " << icheck << " =?  " <<
h[ih].lcell_neighbors[itmp0] << endl;
if( icheck <= JNMAX )
cout << "# WARNING: particle " << i << " with neighbors " << icheck << endl;

} // i-loop
}

// production of a new droplet
void dropproduction()
{
double Vinit = 17; // total velocity of produced droplet

if( ttime >= tdropprod-ACCURA ) // introduce new droplet
{
tdropprod += dtdropprod;

angle1 = (((double(rand())/RAND_MAX)-0.5)*80)/180*M_PI; // Gives you a random number in the
range [ deg ; deg]
angle2 = (((double(rand())/RAND_MAX))*360)/180*M_PI; // Gives you a random number in the range
[0 : 2 pi]
initdisturb = (((double(rand())/RAND_MAX)-0.5)*2e-6); // Gives you a random number in the range
[-1e-6 ;1e-6]

// properties of the new droplet ; must be in accordance with c2d.ini !!!!!!!!!
x[N] = initdisturb*sin(angle2);
y[N] = initdisturb*cos(angle2);
z[N] = -0.00915;
vx[N] = Vinit*sin(angle1) *sin(angle2); // should be random positive and negative, indep. from vy
vy[N] = Vinit*sin(angle1) *cos(angle2); // should be random positive and negative, indep. from vx
vz[N] = -Vinit*cos(angle1); // should always be negative
rad[N] = 4e-6;
xinfo[N] = 0;
sx[N]=x[N]-dt*vx[N];
sy[N]=y[N]-dt*vy[N];
sz[N]=z[N]-dt*vz[N];
mass[N]=4./3.*M_PI*rad[N]*rad[N]*rad[N]*mdensity;
N=N+1;
cout << "# produced particle " << N << " a1,2 = " << angle1 << ' ' << angle2 << " with v=("
<< vx[N-1] << ' ' << vy[N-1] << ' ' << vz[N-1] << ")" << endl;
}
}
```

```
// Find closest JNMAX points for each droplet
void getgg( int i, int j )
{
double dist;
int j1, j2;

dist=sqrt( (x[i]-xef[j])*(x[i]-xef[j]) + (y[i]-yef[j])*(y[i]-yef[j]) + (z[i]-zef[j])*(z[i]-zef[j]));

for( j1=JNMAX-1; j1>=0; j1- )
{
if(dist <= grid[i][j1])
{
for( j2=0; j2<j1; j2++ )
{
grid[i][j2] = grid[i][j2+1]; // distances of gridpoints to particle
jgrid[i][j2] = jgrid[i][j2+1]; // corresponding gridpoint number
}
grid[i][j1] = dist;
jgrid[i][j1] = j;

j1=-1; // Terminate the loop if the distance was smaller
}
}
}

// deposition of droplets on the grounded target (cylinder) and on the walls
void deposition()
{
double circ;

// target deposition
for (int i=0; i<N; i++)
{
if (z[i] <= -0.20815) // condition for y-position
{
if(( -0.1325 < x[i] ) && ( x[i] < 0.1325 )) // condition for z-position
{
if(( -0.0685 < y[i] ) && ( y[i] < 0.0685 )) // condition for x-position
{
circ = (z[i] + 0.27665)*(z[i] + 0.27665) + y[i]*y[i];
if (circ <= 0.00469225) // condition for circ-position ; 0.00469225 is the square of the radius of the
target cylinder
{
Ndep += 1; // number of deposited droplets increases by one!
cout << "# the "<< Ndep <<"th droplet has been deposited at the target at " << x[i] <<' '<< y[i]
<<' '<< z[i] << endl;
deposit << x[i] <<' '<< y[i] <<' '<< z[i] <<' '<< vx[i] <<' '<< vy[i] <<' '<< vz[i] <<' '<<rad[i]<<'
'<< ttime <<' '<< i <<endl;

x[i] = x[N-1]; //replace deposited droplet by last one (copy last droplet!)
y[i] = y[N-1];
z[i] = z[N-1];
vx[i] = vx[N-1];
vy[i] = vy[N-1];
```

```
vz[i] = vz[N-1];
rad[i] = rad[N-1];
xinfo[i] = xinfo[N-1];
sx[i]=x[N-1]-dt*vx[N-1];
sy[i]=y[N-1]-dt*vy[N-1];
sz[i]=z[N-1]-dt*vz[N-1];
mass[i]=4./3.*M_PI*rad[N-1]*rad[N-1]*rad[N-1]*mdensity;
N = N-1; //finally remove the last (copied) droplet
}
}
}
}

// wall deposition
if((x[i]<wx0) || (x[i]>wx1) || (y[i]<wy0) || (y[i]>wy1) || (z[i]<wz0) || (z[i]>wz1))
{
Nwall += 1; // number of deposited droplets increases by one!
cout << "# the "<< Nwall <<"th droplet has been deposited at the wall at " << x[i] <<' '<< y[i]
<<' '<< z[i] << endl;
wall << x[i] <<' '<< y[i] <<' '<< z[i] <<' '<< vx[i] <<' '<< vy[i] <<' '<< vz[i] <<' '<<rad[i]<<'
'<< ttime <<' '<< i <<endl;
x[i] = x[N-1]; //replace deposited droplet by last one (copy last droplet!)
y[i] = y[N-1];
z[i] = z[N-1];
vx[i] = vx[N-1];
vy[i] = vy[N-1];
vz[i] = vz[N-1];
rad[i] = rad[N-1];
xinfo[i] = xinfo[N-1];
sx[i]=x[N-1]-dt*vx[N-1];
sy[i]=y[N-1]-dt*vy[N-1];
sz[i]=z[N-1]-dt*vz[N-1];
mass[i]=4./3.*M_PI*rad[N-1]*rad[N-1]*rad[N-1]*mdensity;
N = N-1; //finally remove the last (copied) droplet
}
}
}

// evaporation of droplets; just an idea: not yet investigated and added to the program
void evaporation()
{
double factor1, factor2;
factor1 = -7.43e-11;
factor2 = 183.2;

for (int i=0; i<N; i++)
{
Vel = (vx[i]*vx[i] + vy[i]*vy[i] + vz[i]*vz[i]);
rad[i] += (factor1/rad[i]*(2 + factor2*sqrt(Vel*rad[i])))*dt;
}
}

// particle wall interaction subroutine
// not used in this program because in subroutine deposition() droplets are deposited at the wall
```

```
void md_wforces()
{
double wdist, delta;

for( int iw=0; iw<NWalls; iw++ )
{
for( int i=0; i<N; i++)
{
wdist=(x[i]-wx[iw])*nwx[iw] +(y[i]-wy[iw])*nwy[iw] +(z[i]-wz[iw])*nwz[iw];
// r ... particle vector
// r_w ... wall vector
// n_w ... wall normal
// wall-distance = (r-r_w)*n_w
delta=wdist-rad[i]; // delta = overlap
if(delta < 0)
{

fx[i] -= K*delta*nwx[iw]; // linear force law p̃ropto overlap fy[i] -= K*delta*nwy[iw];
fz[i] -= K*delta*nwz[iw];

comment<<"particle "<<i<<" bounces with the wall at time "<<ttime<<endl;
}
}
}
}
```

exvthlc.h

```
const int Nmax=10000, Wmax=6, NWalls=6;
const int Nfldmax=95000;
int Nelecfld;
int Ndep = 0;
int Nwall = 0;
const double ACCURA=1e-12;
const int JNMAX=4; // number of gridpoints used for field calculation/interpolation
const int FIELD_MIN=2*JNMAX; // this much gridpoints should be in the neighborhood

int ierr=0; // error code

// Input and output file streams
ifstream parini("par.ini");
ifstream c2dini("c2d.ini");
ifstream fieldini("efield.ini");
ofstream eneout("ene",ios::trunc);
ofstream gnuout("gnu",ios::trunc);
ofstream c2dout("c2d",ios::trunc);
ofstream comment("comment",ios::trunc);
ofstream deposit("deposit",ios::trunc);
ofstream wall("wall",ios::trunc);

using std::cout;
using std::cin;

long int N; // particle number
```

```
double x[Nmax], y[Nmax], z[Nmax];
double vx[Nmax], vy[Nmax], vz[Nmax];

double wx0, wx1, wy0, wy1, wz0, wz1; // container edges
double wx[Wmax], wy[Wmax], wz[Wmax];
double nwx[Wmax], nwy[Wmax], nwz[Wmax];

double rad[Nmax], xinfo[Nmax];
double sx[Nmax], sy[Nmax], sz[Nmax], mass[Nmax];

double fx[Nmax], fy[Nmax], fz[Nmax];

double ttime, dt, dt2, tstart, tstop; // time variables
double tlogout, dtlogout, tfilmout, dtfilmout, tdropprod, dtdropprod;
double mdensity, qparam, mu, rho_air, effrayl, coulomb_charge_factor, epsilon, K, V;
double enepot;
double angle1, angle2, initdisturb;
double Vel;

const double gravz=-9.81; // gravity constant

double xef[Nfldmax], yef[Nfldmax], zef[Nfldmax];
double Ex[Nfldmax], Ey[Nfldmax], Ez[Nfldmax];

int jgrid[Nmax][JNMAX];
double grid[Nmax][JNMAX];

int get_particles(); // function prototypes
int get_parameters();
int get_electricfield();

void pp_initialize();
void walls_initialize();
void compute_parameters();

void loop_init();

void md_integrate();
void md_xforces();
void md_xxforces();
void md_wforces();
void md_pforces();

void enelogout();
void c2dfilmout();
void close_files();

void get_hlcell_grid();
void getgg( int i, int j );

void dropproduction();
void deposition();
void evaporation();
```

exvhlc.h

```
const int HLMAX=5;
int H_LEVEL;

const int NCELL_US=100;
const int NCELLMAX=NCELL_US*2+1;
const int iperiodic=0;

const int lcell_stat_length=Nfldmax+1;

void lcell_parameters( double wwx0, double wwy0, double wwz0, double wwx1, double wwy1, double wwz1 );

void lcell_numcells( int ih );
void lcell_init( int ih );
void lcell_index( int ih, int numcell, int *jxc, int *jyc, int *jzc );

// integer return value functions

int lcell_number( int ih, double xc, double yc, double zc );
int lcell_inumber( int ih, int ixc, int iyc, int izc );
int lcell_p_inumber( int ih, int ixc, int iyc, int izc );
int lcell_pnext( int ih, int j0 );

// int lcell_cnext( int i0, int nca );

void lcell_inext( int ih, int neighbor, int icx0, int icy0, int icz0, int *icx1, int *icy1, int *icz1 );

void lcell_sort( int ih, int lfield, double *xc, double *yc, double *zc );
void lcell_count_neighbors( int ih );

// ATTENTION - THESE ROUTINES USE Nelecfld ...

struct {
// lcell system boundaries
double lcwx0, lcwx1, lcwy0, lcwy1, lcwz0, lcwz1;

// global variables concerning cell numbers and sizes
int ncellx, ncelly, ncellz, ncell_total;
double scellx, scelly, scellz, min_scell;
double sicellx, sicelly, sicellz;

// field with all cells pointing to the first pp in cell
int lcell_member[(NCELLMAX+2)*(NCELLMAX+2)*(NCELLMAX+2)];
// forward and backward pointers for all particles (-1 means empty)
int lcell_point1[Nfldmax], lcell_point2[Nfldmax];

// number of neighbor points to be stored in this field
int lcell_neighbors[(NCELLMAX+2)*(NCELLMAX+2)*(NCELLMAX+2)];

int insort, incont;
int num_neighbors;
} h[HLMAX];
```

```
// Index steps for ALL neighbor cells
int num_ndefault=27;
const int icxs[27]={ 0, 1, 1, 0, -1, -1, -1, 0, 1, 0, 1, 1, 0, -1, -1, -1, 0, 1, 0, 1, 1, 0, -1, -1, -1, 0, 1 };
const int icys[27]={ 0, 0, -1, -1, -1, 0, 1, 1, 1, 0, 0, -1, -1, -1, 0, 1, 1, 1, 0, 0, -1, -1, -1, 0, 1, 1, 1 };
const int iczs[27]={ 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1 };

// Index steps for pp-contact neighbor cells
// int num_neighbors=10;
// const int icxs[10]={ 0, 1, 1, 0,-1, 0, 1, 1, 0,-1 };
// const int icys[10]={ 0, 0, 1, 1, 1, 0, 0, 1, 1, 1 };
// const int iczs[10]={ 0, 0, 0, 0, 0, 1, 1, 1, 1, 1 };

// parameters file "lcell.ini"
ifstream lcellini("lcell.ini");

// read in parameters from file "lcell.ini"

void lcell_parameters( double wwx0, double wwy0, double wwz0, double wwx1, double wwy1, double wwz1 )
{
lcellini >> H_LEVEL;
if( H_LEVEL > HLMAX )
{
cout << "# ERROR - hierarchy level set to max: " << HLMAX;
H_LEVEL = HLMAX;
}
if( H_LEVEL < 1 ) H_LEVEL=1;

for( int ih=0; ih<H_LEVEL; ih++ )
{
lcellini >> h[ih].ncellx >> h[ih].ncelly >> h[ih].ncellz;
// read in number of cells for level ih

h[ih].lcwx0=wwx0; // set system boundaries
h[ih].lcwx1=wwx1;
h[ih].lcwy0=wwy0;
h[ih].lcwy1=wwy1;
h[ih].lcwz0=wwz0;
h[ih].lcwz1=wwz1;
}
}

// check number of cells and return possible values

void lcell_numcells( int ih )
{
int ncx=h[ih].ncellx;
int ncy=h[ih].ncelly;
int ncz=h[ih].ncellz;

if(ncx<1) ncx=1; // minimum number of cells
if(ncy<1) ncy=1;
if(ncz<1) ncz=1;
```

```
if(ncx==2) ncx=1; // do not allow 2 cells
if(ncy==2) ncy=1;
if(ncz==2) ncz=1;
if(ncx>NCELLMAX) ncx=NCELLMAX; // maximum number of cells
if(ncy>NCELLMAX) ncy=NCELLMAX;
if(ncz>NCELLMAX) ncz=NCELLMAX;

h[ih].scellx=(h[ih].lcwx1-h[ih].lcwx0) / ncx; // compute cellsize
h[ih].scelly=(h[ih].lcwy1-h[ih].lcwy0) / ncy;
h[ih].scellz=(h[ih].lcwz1-h[ih].lcwz0) / ncz;

if( iperiodic == 0 ) // set system boundaries
{
ncx+=2;
ncy+=2;
ncz+=2;
h[ih].lcwx0=h[ih].lcwx0-h[ih].scellx;
h[ih].lcwx1=h[ih].lcwx1+h[ih].scellx;
h[ih].lcwy0=h[ih].lcwy0-h[ih].scelly;
h[ih].lcwy1=h[ih].lcwy1+h[ih].scelly;
h[ih].lcwz0=h[ih].lcwz0-h[ih].scellz;
h[ih].lcwz1=h[ih].lcwz1+h[ih].scellz;
cout << "# non-periodic boundaries in lcell-structure "
<< ncx * ncy * ncz - (ncx-2) * (ncy-2) * (ncz-2)
<< " empty outside cells ..." << endl;
}

h[ih].scellx=(h[ih].lcwx1-h[ih].lcwx0+ACCURA) / ncx; // compute cellsize
h[ih].scelly=(h[ih].lcwy1-h[ih].lcwy0+ACCURA) / ncy; // with some tiny margin
h[ih].scellz=(h[ih].lcwz1-h[ih].lcwz0+ACCURA) / ncz; // to account for round-off

h[ih].sicellx=1.0/h[ih].scellx;
h[ih].sicelly=1.0/h[ih].scelly;
h[ih].sicellz=1.0/h[ih].scellz;

cout << "# Initialize linked cells (level " << ih << ") " <<
ncx << ' ' << ncy << ' ' << ncz << '\n';
cout << "# Linked cells sizes : " << h[ih].scellx << ' ' << h[ih].scelly << ' ' << h[ih].scellz << '\n';
cout << "# Linked cells bounds : " << h[ih].lcwx0 << ' ' << h[ih].lcwy0 << ' ' << h[ih].lcwz0 << '
' << h[ih].lcwx1 << ' ' << h[ih].lcwy1 << ' ' << h[ih].lcwz1 << endl;

h[ih].min_scell=h[ih].scellx;
if( h[ih].scelly < h[ih].min_scell ) h[ih].min_scell=h[ih].scelly;
if( h[ih].scellz < h[ih].min_scell ) h[ih].min_scell=h[ih].scellz;

h[ih].ncell_total = (ncx+2) * (ncy+2) * (ncz+2);

h[ih].num_neighbors=num_ndefault;
if( ncz==1 )
{
if( num_ndefault==27 ) h[ih].num_neighbors=9;
if( num_ndefault==10 ) h[ih].num_neighbors=5;
}
```

```
h[ih].ncellx=ncx;
h[ih].ncelly=ncy;
h[ih].ncellz=ncz;
}

void lcell_init( int ih )
{
cout << "# Initialize field structure " << ih << ' ' << Nelecfld << ' ' << h[ih].ncell_total << endl;

for( int i=0; i<Nelecfld; i++ )
{
h[ih].lcell_point1[i]=-1;
h[ih].lcell_point2[i]=-1;
}
for( int i=0; i<h[ih].ncell_total; i++ )
{
h[ih].lcell_member[i]=-1;
}
}

// Funtion returns cell indices from the number
void lcell_index( int ih, int numcell, int *jxc, int *jyc, int *jzc )
{
int ntmp;

jzc=numcell-int(numcell/h[ih].ncellz)*h[ih].ncellz;
ntmp=(numcell- *jzc)/h[ih].ncellz;
jyc=ntmp-(ntmp/h[ih].ncelly)*h[ih].ncelly;
jxc=(ntmp- *jyc)/h[ih].ncelly;
}

// Function returns the number of the cell with indices (ixc,iyc,izc)
// ATTENTION - no check for boundaries

int lcell_inumber( int ih, int ixc, int iyc, int izc )
{
return izc + h[ih].ncellz * ( iyc + h[ih].ncelly * ixc );
}

int lcell_p_inumber( int ih, int ixc, int iyc, int izc )
{
if( iperiodic )
{
if( ixc < 0 ) ixc=h[ih].ncellx-1;
if( iyc < 0 ) iyc=h[ih].ncelly-1;
if( izc < 0 ) izc=h[ih].ncellz-1;
if( ixc >= h[ih].ncellx ) ixc=0;
if( iyc >= h[ih].ncelly ) iyc=0;
if( izc >= h[ih].ncellz ) izc=0;
}
else
{
if(( ixc < 0 )||( ixc >= h[ih].ncellx )||
( iyc < 0 )||( iyc >= h[ih].ncelly )||
```

```
( izc < 0 )||( izc >= h[ih].ncellz ))
{
cout << "# ERROR - out of lcell-range (non-periodic) " << ih << ' ' << ixc << ' ' << iyc << ' '
<< izc << '\n';
}
}
return izc + h[ih].ncellz * ( iyc + h[ih].ncelly * ixc );
}

// Function returns the number of the cell with coordinates (xc,yc,zc)

int lcell_number( int ih, double xc, double yc, double zc )
{
int inum_cell, ix, iy, iz;

if( xc < h[ih].lcwx0 ) xc+=(h[ih].lcwx1-h[ih].lcwx0);
if( yc < h[ih].lcwy0 ) yc+=(h[ih].lcwy1-h[ih].lcwy0);
if( zc < h[ih].lcwz0 ) zc+=(h[ih].lcwz1-h[ih].lcwz0);
if( xc > h[ih].lcwx1 ) xc-=(h[ih].lcwx1-h[ih].lcwx0);
if( yc > h[ih].lcwy1 ) yc-=(h[ih].lcwy1-h[ih].lcwy0);
if( zc > h[ih].lcwz1 ) zc-=(h[ih].lcwz1-h[ih].lcwz0);

ix=int((xc-h[ih].lcwx0)*h[ih].sicellx); // get integer values corresp. to coords
iy=int((yc-h[ih].lcwy0)*h[ih].sicelly);
iz=int((zc-h[ih].lcwz0)*h[ih].sicellz);

inum_cell = iz + h[ih].ncellz * ( iy + h[ih].ncelly * ix );
// compute integer number corresp. to cell
if( inum_cell >= h[ih].ncell_total )
{
cout << "# WARNING - lcell_number too large " << h[ih].ncell_total << ' ' << inum_cell << ' ' <<
ix << ' ' << iy << ' ' << iz << '\n';
cout << "# Pos.: " << xc << ' ' << yc << ' ' << zc << '\n';
}

return inum_cell;
}

void lcell_sort( int ih, int lfield, double *xc, double *yc, double *zc )
{
int j, icnum, itmp;
int lcell_particles;

int lcell_stat[lcell_stat_length];

for( j=0; j<lcell_stat_length; j++ )
lcell_stat[j]=0;

lcell_init( ih ); // make the fields empty first

cout << "# sort field len (h=" << ih << ")= " << lfield << '\n';
h[ih].insort=0;
for( int i=0; i<lfield; i++ )
{
```

```
icnum = lcell_number( ih, xc[i], yc[i], zc[i] );

if( h[ih].lcell_member[ icnum ] >= 0 ) // already a particle in cell
{
itmp=h[ih].lcell_member[ icnum ]; // get first particle in cell
for( int jj=0; jj<Nelecfld; jj++ ) // search end of linked list in this cell
{
if( h[ih].lcell_point2[ itmp ] < 0 )
{
h[ih].lcell_point1[ i ] = itmp;
h[ih].lcell_point2[ i ] = -1;
h[ih].lcell_point2[ itmp ] = i;
jj=Nelecfld;
h[ih].insort++;
}
else
{
itmp=h[ih].lcell_point2[ itmp ];
}
}
}
else // first particle in cell
{
h[ih].lcell_member[ icnum ] = i;
// put particle i as first pp in cell
h[ih].lcell_point1[ i ] = icnum+Nelecfld;
h[ih].insort++;

if( h[ih].lcell_point2[ i ] < 0 )
h[ih].lcell_point2[ i ] = -1;
else
cout << "# Warning - lcell_point 2 " << i << ' ' << h[ih].lcell_point2[ i ] << endl;
}
}

// output the result
h[ih].incont=0;
for( int ix=0; ix<h[ih].ncellx; ix++ )
for( int iy=0; iy<h[ih].ncelly; iy++ )
for( int iz=0; iz<h[ih].ncellz; iz++ )
{
lcell_particles=0;
itmp = lcell_number( ih, h[ih].lcwx0+(double(ix)+0.5)*h[ih].scellx, h[ih].lcwy0+(double(iy)+0.5)*h[ih].scelly,
h[ih].lcwz0+(double(iz)+0.5)*h[ih].scellz );
j = h[ih].lcell_member[ itmp ];
if( j >= 0 )
{
lcell_particles++;
// count particles per cell
j = lcell_pnext( ih, h[ih].lcell_member[ itmp ] );
while( j >= 0 )
{
j = lcell_pnext( ih, j );
lcell_particles++;
```

```
}
h[ih].incont+=lcell_particles;

if( lcell_particles > lcell_stat_length-1 )
lcell_particles=lcell_stat_length-1;
lcell_stat[ lcell_particles ] += 1;
}
else
{
lcell_stat[ 0 ] += 1;
}
}

cout << "# control numbers : " << h[ih].insort << ' ' << h[ih].incont << '\n';
h[ih].incont=0;
for( j=0; j<lcell_stat_length; j++ )
{
if( lcell_stat[ j ] > 0 )
{
cout << "# gridpoints in cell: " << j << " cells: " << lcell_stat[ j ] << '\n';
h[ih].incont += j * lcell_stat[ j ];
}
}
cout << "# sum over statistics : " << h[ih].incont << '\n';

// count the fields and neighbors etc.
lcell_count_neighbors( ih );
}

// get next particle in the neighborhood

int lcell_pnext( int ih, int j0 )
{
return h[ih].lcell_point2[ j0 ];
}

void lcell_inext( int ih, int nn, int icx0, int icy0, int icz0, int *icx1, int *icy1, int *icz1 )
{
icx1=icx0+icxs[nn];
icy1=icy0+icys[nn];
icz1=icz0+iczs[nn];

// check if out of bounds – periodic boundaries
// lcell_index( ih, incell, icx1, icy1, icz1 );
}

// This is an example how to run through all neighbors of one cell

void lcell_count_neighbors( int ih )
{
int j;
int inmin=Nelecfld;
int inmax=0;
int itmp, itmp0, icheck;
```

```
int ix, iy, iz;
int icx1, icy1, icz1;

if( iperiodic == 0 )
{
for( ix=1; ix<h[ih].ncellx-1; ix++ )
for( iy=1; iy<h[ih].ncelly-1; iy++ )
for( iz=1; iz<h[ih].ncellz-1; iz++ )
{
icheck=0;
itmp0 = lcell_number( ih, h[ih].lcwx0+(double(ix)+0.5)*h[ih].scellx, h[ih].lcwy0+(double(iy)+0.5)*h[ih].scelly,
h[ih].lcwz0+(double(iz)+0.5)*h[ih].scellz );
j = h[ih].lcell_member[ itmp0 ] ;
while( j >= 0 )
{
icheck+=1;
j = lcell_pnext( ih, j );
}

// run through all neighbor cells (27—10 in 3D)
for( int neighbor=1; neighbor<h[ih].num_neighbors; neighbor++ )
{
lcell_inext( ih, neighbor, ix, iy, iz, &icx1,&icy1,&icz1 ); // get indices of neighbor cell
itmp=lcell_p_inumber( ih, icx1, icy1, icz1 ); // get number of this cell
j = h[ih].lcell_member[ itmp ] ;
while( j >= 0 )
{
icheck+=1;
j = lcell_pnext( ih, j );
}
}
if( icheck < inmin ) inmin=icheck;
if( icheck > inmax ) inmax=icheck;

h[ih].lcell_neighbors[ itmp0 ] = icheck;
}
cout << "# min-max occupied cells in the neighborhood: " << inmin << ' ' << inmax << '\n';
} // do it only for non-periodic boundaries
else
{
// periodic boundaries - to be done
}
}
```

exvthlc.bat

```
gxx exvthlc.cc -o exvthlc.exe -O -I. -lm
```

Input and output files of the C++ program

C

This appendix explains the input files needed to run the simulation program. The output files of the simulation are also discussed.

Input files

par.ini

```
tstop dt
dtlogout dtfilmout dtdropprod
K V
mdensity qparam effrayl coulomb_charge_factor
mu rho_air epsilon
```

first line: tstop = time when the simulation is stopped; dt = timestep of numerical calculation/integration.
secondline: dtlogout = timestep for output to c2d; dtfilmout = timestep for output to gnu; dtdropprod = timestep for the production of droplets.
thirdline: K = constant for a spring equation; V = damping factor for an spring oscillation : BOTH QUANTITIES ARE NOT USED IN THE MODEL.
fourth line: mdensity = density of liquid; qparam = constant factor in the equation for the charge of a droplet; effrayl = efficiency for the charge of a droplet with respect to the Rayleigh limit; coulomb_charge_factor = factor with which the charge of a droplet is multiplied for droplet-droplet interactions (Coulomb forces).
fifth line: mu = viscosity of ambient air; rho_air = density of ambient air; epsilon = vacuum permittivity.

c2d.ini

```
N tstart wx0 wy0 wz0 wx1 wy1 wz1
x[i] y[i] z[i] vx[i] vy[i] vz[i] rad[i] xinfo[i]
```

first line: N = number of droplets at initial time; tstart = initial time; wx0 = lower x-wall boundary; wy0 = lower y-wall boundary; wz0 = lower z-wall boundary; wx1 = higher x-wall boundary; wy1 = higher y-wall boundary; wz1 = higher z-wall boundary;
second to (N+1)th line: x[i] = x-position of initial droplet i; y[i] = y-position of initial

droplet i; z[i] = z-position of initial droplet i; vx[i] = initial velocity in x-direction of droplet i; vy[i] = initial velocity in y-direction of droplet i; vz[i] = initial velocity in z-direction of droplet i; rad[i] = radius of initial droplet i; xinfo[i] = information parameter e.g. for distinction between droplet in air or deposited at target or wall.

lcell.ini

```
H_LEVEL
h[ih].ncellx h[ih].ncelly h[ih].ncellz
```

first line: H_LEVEL = level of hierarchy for the linked cell structure.
second to (H_LEVEL+1)th line: h[ih].ncellx = number of cells in x-direction for level ih; h[ih].ncelly = number of cells in y-direction for level ih; h[ih].ncellz = number of cells in z-direction for level ih.

efield.ini

```
Nelecfld
xef[i] yef[i] zef[i] Ex[i] Ey[i] Ez[i]
```

first line: Nelecfld = number of grid points at which the external electric field is known from FEMLAB.
second to (Nelecfld+1)th line: xef[i] = x-position of grid point i; yef[i] = y-position of grid point i; zef[i] = z-position of grid point i; Ex[i] = electric field in x-direction at gridpoint i; Ey[i] = electric field in y-direction at gridpoint i; Ez[i] = electric field in z-direction at gridpoint i;

Output files

c2d

Idem to *c2d.ini* instead now *tstart* is the running time *ttime* and therefor all properties of the droplet present in air are at that present time *ttime*. The file is filled up (with the same lines) every *dtlogout*.

gnu

```
N ttime x[i] y[i] z[i] vx[i] vy[i] vz[i] rad[i] xinfo[i]
```

All parameters are identical to those mentioned in *c2d*, only a different way of storing is used. The file is filled up (with the same lines) every $dtfilmout$.

deposit

`x[i] y[i] z[i] vx[i] vy[i] vz[i] rad[i] ttime i`

Obviously the parameters concern droplets deposited at the metal target cylinder. All parameters are mentioned before except voor i, which is the number of the droplet that has deposited given at it's production. The file is filled up every dt.

wall

Idem to *deposit* but for droplets deposited at the boundary walls.

comment

`'dtlogout' 'dtfilmout' 'dtdropprod' 'coulomb_charge_factor'`
`dtlogout dtfilmout dtdropprod coulomb_charge_factor`

`droplet` i `bounces with the wall at time` $ttime$

`The initial time is 'Type of day' 'Month' 'Day' 'time' 'year'`

`The final time is 'Type of day' 'Month' 'Day' 'time' 'year'`

First some typical parameters for the simulation are put out. Next bounces between droplets are written down. Finally at the end of the program the initial and final time of the simulation can be read in the file.

Additional C++ programs

D

Besides the main C++ program representing the model some other C++ programs are written that were needed during the modelling. These are presented here.

exvc2dtot.exe

The program `exvc2dtot.exe` is made by compiling the files `exvc2dtot.cc` and `exvc2dtot.h` with the line in `exvc2dtot.bat`. The program produces the output file `c2dtotal` which consist of properties of droplets in air and deposited at the metal target cylinder or boundary walls throughout the simulation time.

exvc2dtot.cc

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>

#include < exvc2dtot.h >

int main()
{
int count_d = 0;
int count_w = 0;

N = 0;

// read in the deposit file
// !!!!!!!!!!!!!!! WATCH OUT !!!!!!!!!!!!!!!!!
// the last line of the original deposit and wall files must be the first now!!! these are the following lines:

deposit >> N_dep_tot >> tstart >> tstop >> wx0 >> wx1 >> wy0 >> wy1 >> wz0 >> wz1;
wall >> N_wall_tot >> tstart >> tstop >> wx0 >> wx1 >> wy0 >> wy1 >> wz0 >> wz1;

cout<<"# Start of making c2dtotal"<<endl;
cout<<endl;
cout<<"N_dep_tot = "<<N_dep_tot<<' '<<"N_wall_tot = "<<N_wall_tot<<endl;
cout<<endl;

for(int j=0; j<N_dep_tot; j++)
{
deposit >> xd[j] >> yd[j] >> zd[j] >> vxd[j] >> vyd[j] >> vzd[j] >> radd[j] >> deptime[j] >>
```

```
depnumber[j];
}

for(int k=0; k<N_wall_tot; k++)
{
wall >> xw[k] >> yw[k] >> zw[k] >> vxw[k] >> vyw[k] >> vzw[k] >> radw[k] >> walltime[k] >>
wallnumber[k];
}

c2d >> N >> ttime >> wx0 >> wx1 >> wy0 >> wy1 >> wz0 >> wz1;
N_dep = 0; // initialisation of number of deposited droplets at the target (at tstart)
N_wall = 0; // initialisation of number of deposited droplets at the wall (at tstart)
N_tot = N + N_dep + N_wall;
c2dtotal <<N_tot<<' '<<ttime<<' '<<wx0<<' '<<wx1<<' '<<wy0<<' '<<wy1<<' '<<wz0<<'
'<<wz1<< endl;

while((N > 0)&&(N<Nmax))
{
// c2d
for(int i = 0; i<N; i++) //c2d-loop
{
c2d >> x[i] >> y[i] >> z[i] >> vx[i] >> vy[i] >> vz[i] >> rad[i] >> xinfo[i];
c2dtotal <<x[i]<<' '<<y[i]<<' '<<z[i]<<' '<<vx[i]<<' '<<vy[i]<<' '<<vz[i]<<' '<<rad[i]<<' '<<xinfo[i]<<
endl;
}

// deposit : output to c2dtotal
for(int p=0; p<N_dep_tot; p++)
{
if(deptime[p]<ttime) //deposit-loop
{
c2dtotal <<xd[p]<<' '<<yd[p]<<' '<<zd[p]<<' '<<vxd[p]<<' '<<vyd[p]<<' '<<vzd[p]<<' '<<radd[p]<<'
'<<"1"<<endl;
}
}

// wall : output to c2dtotal
for(int m=0; m<N_wall_tot; m++)
{
if(walltime[m]<ttime) //deposit-loop
{
c2dtotal <<xw[m]<<' '<<yw[m]<<' '<<zw[m]<<' '<<vxw[m]<<' '<<vyw[m]<<' '<<vzw[m]<<'
'<<radw[m]<<' '<<"2"<<endl;
}
}

N = 0; // reinitialise number of droplets in air

c2d >> N >> ttime >> wx0 >> wx1 >> wy0 >> wy1 >> wz0 >> wz1; // read in new c2d-
line

// deposit : counter of deposited droplets at the target
for(int p=0; p<N_dep_tot; p++)
{
```

```
if(deptime[p]<ttime) //deposit-loop
{
count_d += 1;
}
}

// wall : counter of deposited droplets at the wall
for(int m=0; m<N_wall_tot; m++)
{
if(walltime[m]<ttime) //deposit-loop
{
count_w += 1;
}
}

N_dep = count_d; // number of at the target deposited droplets until ttime
N_wall = count_w; // number of at the wall deposited droplets until ttime
N_tot = N + N_dep + N_wall; // needed for output tot c2dtotal

if((N > 0)&&(N<Nmax)) // prevent to write a last line with N = 0!!
{
c2dtotal <<N_tot<<' '<<ttime<<' '<<wx0<<' '<<wx1<<' '<<wy0<<' '<<wy1<<' '<<wz0<<'
'<<wz1<< endl;
// write out new c2dtotal-line
}

count_d = 0; // reinitialise
count_w = 0; // reinitialise
}
close_files();

cout << "# End of program ... " <<endl;
}

void close_files()
{
c2d.close();
deposit.close();
c2dtotal.close();
}
```

exvc2dtot.h

```
const int Nmax=15000;
const int Wmax=6;
// Input and output file streams
ifstream c2d("c2d");
ifstream deposit("deposit");
ifstream wall("wall");
ofstream c2dtotal("c2dtotal",ios::trunc);

using std::cout;
using std::cin;
```

```
int N; // number of droplets in air
int N_dep; // number of deposited droplets at the target
int N_wall; // number of deposited droplets at the wall
int N_tot; // total number of droplets, thus in air and deposited
int N_dep_tot; // total number of deposited droplets at the target at tstop
int N_wall_tot; // total number of deposited droplets at the wall at tstop
int depnumber[Nmax]; // the number of the deposited droplets at the target
int wallnumber[Nmax]; // the number of the deposited droplets at the wall
double deptime[Nmax]; // time of deposition of a droplet at the target
double walltime[Nmax]; // time of deposition of a droplet at the wall

double x[Nmax], y[Nmax], z[Nmax]; // for droplets in air
double vx[Nmax], vy[Nmax], vz[Nmax];
double rad[Nmax], xinfo[Nmax];

double xd[Nmax], yd[Nmax], zd[Nmax]; // for deposited droplets
double vxd[Nmax], vyd[Nmax], vzd[Nmax];
double radd[Nmax];

double xw[Nmax], yw[Nmax], zw[Nmax]; // for deposited droplets
double vxw[Nmax], vyw[Nmax], vzw[Nmax];
double radw[Nmax];

double wx0, wx1, wy0, wy1, wz0, wz1; // container edges
double wx[Wmax], wy[Wmax], wz[Wmax];

double ttime, tstop, tstart; // time variables
double tlogout, dtlogout, tfilmout, dtfilmout, tdropprod, dtdropprod;

void close_files();
```

exvc2dtot.bat

```
gxx exvc2dtot.cc -o exvc2dtot.exe -O -I. -lm
```

makegrid.exe

The program `makegrid.exe` is made by compiling the files `makegrid.cc` and `makegrid.h` with the line in `makegrid.bat`. The program removes grid points of the external electric field that occur more than once and creates one remaining grid point with the average electric field components of all 'double grid points'. The program creates a new file `grid` with the remaining grid points.

makegrid.cc

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <math.h>
#include <iomanip.h>
#include <stdlib.h>
#include <time.h>

#include <makegrid.h>
```

```
int main()
{
get_field();
exclusion();
cout << "# ready" << endl;
}

// SUBROUTINES

// Get electric field from "field.ini"

int get_field()
{
fieldini >> Nelecfld;
cout << Nelecfld <<endl;

for( int i=0; i<Nelecfld; i++ ) // Nelecfld is the length of the columns in field.ini
{
fieldini >> xef[i] >> yef[i] >> zef[i] >> Ex[i] >> Ey[i] >> Ez[i];
}
fieldini.close();
cout << "# Read field-file \n";
return 0;
}

void exclusion()
{
double diffx, diffy, diffz, sum;
int count[Nelecfld];

cout << "start exclusion procedure" << endl;

for (int i=0 ; i<Nelecfld ;i++)
{
newfld[i][0] = xef[i];
newfld[i][1] = yef[i];
newfld[i][2] = zef[i];
newfld[i][3] = Ex[i];
newfld[i][4] = Ey[i];
newfld[i][5] = Ez[i];
count[i] = 1;
}

int i = 0;
int i0 = 0;
int j = 0;
int replaced = 1;

while(replaced > 0)
{
replaced = 0;
i = i0;
while(i < Nelecfld)
```

```
{
j = 0;
while(j < i)
{
diffx = newfld[i][0] - newfld[j][0];
diffy = newfld[i][1] - newfld[j][1];
diffz = newfld[i][2] - newfld[j][2];
sum = diffx + diffy + diffz;

if (sum == 0)
{
if ((diffx == 0) && (diffy == 0) && (diffz == 0))
{
count[j] += 1; // count the number of double grid points for every j

newfld[j][3] = (newfld[i][3] + newfld[j][3]); //replace electric field with sum of the values of the 'double
grid points'
newfld[j][4] = (newfld[i][4] + newfld[j][4]);
newfld[j][5] = (newfld[i][5] + newfld[j][5]);

Nelecfld = Nelecfld-1; //Nelecfld is reduced by one by leaving out the 'double grid point'
newfld[i][0] = newfld[Nelecfld][0]; //replace 'double grid point' with last one
newfld[i][1] = newfld[Nelecfld][1];
newfld[i][2] = newfld[Nelecfld][2];
newfld[i][3] = newfld[Nelecfld][3];
newfld[i][4] = newfld[Nelecfld][4];
newfld[i][5] = newfld[Nelecfld][5];

i0 = i;
i = Nelecfld + 1;
j = Nelecfld + 1;
replaced = 1;
}
}
j++;
}
i++;
}
}
cout << Nelecfld <<endl;

grid << Nelecfld << endl;

for (int i=0 ; i<Nelecfld ;i++)
{
newfld[i][3] = newfld[i][3]/count[i]; //replace electric field with the average values of the 'double grid
points'
newfld[i][4] = newfld[i][4]/count[i];
newfld[i][5] = newfld[i][5]/count[i];

grid <<setprecision(16)<<newfld[i][0]<<' '<<newfld[i][1]<<' '<<newfld[i][2]<<' '<<newfld[i][3]<<'
'<<newfld[i][4]<<' '<<newfld[i][5]<<endl;
}
}
```

makegrid.h

```
const int Nfldmax=150000;
int Nelecfld;
const double ACCURA=1e-12;

int ierr=0; // error code

// Input and output file streams
ifstream fieldini("field.ini");
ofstream grid("grid",ios::trunc);

using std::cout;
using std::cin;

double xef[Nfldmax], yef[Nfldmax], zef[Nfldmax];
double Ex[Nfldmax], Ey[Nfldmax], Ez[Nfldmax];

double newfld[Nfldmax][6];

int get_field();
void exclusion();
```

makegrid.bat

```
gxx makegrid.cc -o makegrid.exe -O -I. -lm
```

Additional MATLAB programs

E

During the modelling several programs are written in MATLAB to evaluate certain data. This appendix includes these MATLAB files.

cornerevaluation.m

This program calculates the electric field in x-, y- and z-direction at every corner of a Delauney grid tetraeder, using the solution calculated with FEMLAB.

```
% First clear the workspace in the MATLAB command window
% Second export the FEM structure 'fem' from FEMLAB

% posteval evaluation
[x,y,z,ex,ey,ez] = posteval(fem, 'x', 'y', 'z', 'Ex', 'Ey', 'Ez', 'spoint','corner');

X=x{1};
x1=X(1,:,1)';
x2=X(1,:,2)';
x3=X(1,:,3)';
x4=X(1,:,4)';
xtot = [x1;x2;x3;x4];

Y=y{1};
y1=Y(1,:,1)';
y2=Y(1,:,2)';
y3=Y(1,:,3)';
y4=Y(1,:,4)';
ytot = [y1;y2;y3;y4];

Z=z{1};
z1=Z(1,:,1)';
z2=Z(1,:,2)';
z3=Z(1,:,3)';
z4=Z(1,:,4)';
ztot = [z1;z2;z3;z4];

Ex=ex{1};
ex1=Ex(1,:,1)';
ex2=Ex(1,:,2)';
ex3=Ex(1,:,3)';
ex4=Ex(1,:,4)';
extot = [ex1;ex2;ex3;ex4];

Ey=ey{1};
```

```
ey1=Ey(1,:,1)';
ey2=Ey(1,:,2)';
ey3=Ey(1,:,3)';
ey4=Ey(1,:,4)';
eytot = [ey1;ey2;ey3;ey4];

Ez=ez{1};
ez1=Ez(1,:,1)';
ez2=Ez(1,:,2)';
ez3=Ez(1,:,3)';
ez4=Ez(1,:,4)';
eztot = [ez1;ez2;ez3;ez4];

% placing vectors in a matrix
mat = [xtot ytot ztot extot eytot eztot];

% saving the matrix in an ini-file
save field.ini mat -ascii -double
```

mirror.m

This program mirrors the electric field data created with `makegrid.exe`.

```
% clear workspace
clear all
% load the file made with cornerevaluation.m and makegrid.exe
load grid; % this could have a different name "name"

% rename the matrix : in this way only the name of the loaded file has to be changed if necessary
grid = "name";

% extracting parameters from grid
A = grid;
xtot = grid(:,1);
ytot = grid(:,2);
ztot = grid(:,3);
extot = grid(:,4);
eytot = grid(:,5);
eztot = grid(:,6);

% mirroring
B = [-xtot ytot ztot -extot eytot eztot];
C = [-xtot -ytot ztot -extot -eytot eztot];
D = [xtot -ytot ztot extot -eytot eztot];
E = [A ; B; C; D];

% saving the new matrix E in an ini-file
save totalfield.ini E -ascii -double
```

evaluate_deposition.m

This program analyses the droplet deposition pattern on the target cylinder of a simulation.

```
clear all;
format long;

load deposit;

N = deposit(1,1)

dist_max = 0; % initialise at zero
dist_max_x = 0; % initialise at zero
dist_max_y = 0; % initialise at zero
sum_dist = 0; % initialise at zero
sum_dist_sq = 0; % initialise at zero
sum_dist_x = 0; % initialise at zero
sum_dist_y = 0; % initialise at zero

for i=1:N,

dist_sq(i) = (deposit(i+1,1))^ 2 + (deposit(i+1,2))^ 2;
dist(i) = sqrt(dist_sq(i));
dist_x(i) = abs(deposit(i+1,1));
dist_y(i) = abs(deposit(i+1,2));

% calculate dist_max
if dist(i)>dist_max,
dist_max = dist(i);
i_max = i;
end

if dist_x(i)>dist_max_x,
dist_max_x = dist_x(i);
i_max_x = i;
end

if dist_y(i)>dist_max_y,
dist_max_y = dist_y(i);
i_max_y = i;
end

% calculate sum of distances
sum_dist = sum_dist + dist(i);
sum_dist_x = sum_dist_x + dist_x(i);
sum_dist_y = sum_dist_y + dist_y(i);

% calculate sum of squared distances
sum_dist_sq = sum_dist_sq + dist_sq(i);

end

% maximum distance
dist_max

% average distance
dist_av_1 = sum_dist/N
```

```
% average distance 2
dist_av_2 = sum_dist_sq/N

% maximum distance in x-direction
dist_max_x

% average distance in x-direction
dist_av_x = sum_dist_x/N

% maximum distance in y-direction
dist_max_y

% average distance in y-direction
dist_av_y = sum_dist_y/N

% plot
k = 2:N+1;
plot(deposit(k,1),deposit(k,2),'r.')
grid
xlabel('x (m)')
ylabel('y (m)')
hold on
plot(deposit(i_max+1,1),deposit(i_max+1,2),'b*')
plot(deposit(i_max_x+1,1),deposit(i_max_x+1,2),'g*')
plot(deposit(i_max_y+1,1),deposit(i_max_y+1,2),'c*')
```

Interpolation algorithm F

In this appendix the interpolation algorithm used to calculate the external electric field at the position of a droplet is explained. The interpolation algorithm discusses the averaging strategy for tensorial quantities from discrete data-sets. For details is referred to S. Luding.

Taylor series average

The points where the electric field components $\vec{E}$ are known are the grid points $\vec{g}$. The point where the droplet (momentarily) stays is $\vec{r}$. The interpolation algorithm uses a crude midpoint averaging and the first term of a Taylor expansion around that midpoint. This results in:

$$\vec{E}(\vec{r}) = \vec{E}_0 + grad(\vec{E}|_0) \cdot (\vec{r} - \vec{g}_0), \tag{F-1}$$

with the midpoint $\vec{g}_0 = (1/m)\sum_{i=1}^{m} \vec{g}_i$, and the midpoint electric field $\vec{E}_0 = (1/m)\sum_{i=1}^{m} \vec{E}_i$. In this work m is set to 4, meaning the electric field is interpolated between the 4 closest grid points to the droplet. The gradient of the electric field around the midpoint $grad(\vec{E}|_0)$ is discussed in the rest of the appendix.

The gradient is obtained from any two grid points $\vec{g}_i$ and $\vec{g}_j$ with their corresponding field values $\vec{E}_i$ and $\vec{E}_j$ is

$$grad_{ij}(\vec{E}) = \frac{\vec{E}_i - \vec{E}_j}{|\vec{g}_i - \vec{g}_j|}\hat{\vec{n}}_{ij} = (grad(\vec{E}) \cdot \hat{\vec{n}}_{ij})\hat{\vec{n}}_{ij} \ ,$$

where the subscript ij indicates that the gradient is the projection of the gradient into the direction along the connection of the points i and j,

$$\hat{\vec{n}}_{ij} = \frac{\vec{g}_i - \vec{g}_j}{|\vec{g}_i - \vec{g}_j|}$$

The gradient $grad_{ij}(\vec{E})$ is thus a degenerate tensor with one non-zero eigenvalue in the eigen-direction $\hat{\vec{n}}_{ij}$. In order to obtain the full tensor, the directed tensor is projected into the coordinate directions $\alpha = x, y, z$ and sampled over all $n = m(m-1)/2$ pairs i and j. This leads to:

$$grad_\alpha(\vec{E}) \propto \left(\sum_{i \neq j} grad_{ij}(\vec{E}) \cdot \hat{\vec{n}}_\alpha\right) \hat{\vec{n}}_\alpha, \tag{F-2}$$

with the unit vector $\hat{\vec{n}}_\alpha$ along the α-direction, where $grad_\alpha$ indicates the α-component of the gradient. The projection term corresponds to a weighting proportional to $w_{ij}^\alpha =$

$\cos\phi_{ij}^{\alpha}|\cos\phi_{ij}^{\alpha}|^{k}$, with a power k that is set to 4, and the angle ϕ_{ij}^{α} between the vectors $\hat{\vec{n}}_{ij}$ and $\hat{\vec{n}}_{\alpha}$. Thus after rewriting Eq. (F-2) and normalising with the sum of the weighting factors $w^{\alpha}=\sum_{i\neq j}w_{ij}^{\alpha}$, so that:

$$grad_{\alpha}(\vec{E})=\frac{1}{w^{\alpha}}\left(\sum_{i\neq j}(grad(\vec{E})\cdot\hat{\vec{n}}_{ij})\hat{\vec{n}}_{ij}\cdot\hat{\vec{n}}_{\alpha}\right)\hat{\vec{n}}_{\alpha}. \tag{F-3}$$

Finally, the three components together lead to

$$grad(\vec{E})=\sum_{\alpha}grad_{\alpha}(\vec{E})=\sum_{\alpha}\sum_{i\neq j}\frac{w_{ij}^{\alpha}}{w^{\alpha}}\left(\frac{\vec{E}_i-\vec{E}_j}{|\vec{g}_i-\vec{g}_j|}\right)_{\alpha}\hat{\vec{n}}_{\alpha} \tag{F-4}$$

The interpolation Equation (F-1) results in

$$\vec{E}(\vec{r})=\vec{E}_0+\sum_{\alpha}\sum_{i\neq j}\frac{w_{ij}^{\alpha}}{w^{\alpha}}\left(\frac{\vec{E}_i-\vec{E}_j}{|\vec{g}_i-\vec{g}_j|}(\vec{r}-\vec{g}_0)\right)_{\alpha} \tag{F-5}$$