

Experimental Investigation of Micrometric Droplets and Artificial Particles

Dissertation for the award of the degree
“Doctor rerum naturalium”
of the Georg-August-Universität Göttingen

within the doctoral program

“Göttingen Graduate Center for Neurosciences, Biophysics, and Molecular
Biosciences (GGNB): Physics of Biological and Complex Systems (PBCS)”

of the Georg-August University School of Science (GAUSS)

submitted by

Johannes Milan Güttler

from Nürnberg

Max Planck Institute for Dynamics and Self-Organization, DE-37077 Göttingen

April 15, 2021

Thesis Committee

Prof. Dr. Dr. hc. Eberhard Bodenschatz / Laboratory for Fluid Physics, Pattern Formation and Biocomplexity / Max Planck Institute for Dynamics and Self-Organization

Prof. Dr. Jens Niemeyer / Institute of Astrophysics / Georg-August Universität Göttingen

Prof. Dr. Dr. Andreas Dillmann / Institute of Aerodynamics and Flow Technology / German Aerospace Center (DLR)

Members of the Examination Board

Prof. Dr. Dr. hc. Eberhard Bodenschatz (first reviewer) / Laboratory for Fluid Physics, Pattern Formation and Biocomplexity / Max Planck Institute for Dynamics and Self-Organization

Apl. Prof. Dr. Martin Rein (second reviewer) / Institute of Aerodynamics and Flow Technology / German Aerospace Center (DLR)

Further members of the Examination Board

Dr. David Zwicker / Theory of Biological Fluids (Max Planck Research Group) / Max Planck Institute for Dynamics and Self-Organization

Apl. Prof. Dr. Ulrich Parlitz / Biomedical Physics (Max Planck Research Group) / Max Planck Institute for Dynamics and Self-Organization

Contents

Contents	III
List Of Figures	V
List Of Tables	VII
Acknowledgements	VIII
1 Introduction	1
1.1 Clouds and particles in Earth’s atmosphere	1
1.2 Aerodynamics of spherical particles	3
1.3 Aerodynamics of non-spherical particles	5
1.4 Objectives of this work	11
2 Design of a droplet generator for the generation of micrometric droplets	12
2.1 Introduction	12
2.2 Generating droplets	12
2.2.1 Common droplet generator types	14
2.2.2 Design considerations	15
2.3 Development of a droplet generator	18
2.4 Conclusion	24
3 Two-Photon-Polymerization (TPP) for the production of nozzles and atmospheric particles	26
3.1 Introduction	26
3.2 2-Photon-Polymerization and Nanoscribe GT	27
3.2.1 Fundamentals	27
3.2.2 Printing parameters	28
3.3 Droplet generator nozzles	31
3.4 Ellipsoidal particles	35
3.4.1 Shape descriptors: Elongation (EL) and flatness (FL)	35
3.4.2 Particle printing results	36
3.5 Conclusion and outlook	37
4 Design of a settling chamber	39
4.1 Introduction	39
4.2 Drop tower	41
4.3 Insertion mechanism	42
4.3.1 Evolution of the insertion mechanism	44
4.4 Shadowgraphy	45
4.4.1 High-speed cameras and LED light sources	45
4.5 Calibration	46
4.5.1 Calibration procedure	47
4.5.2 Processing the calibration images	48
4.6 Measurement procedure	49

4.7	Validation of the experimental setup	50
4.7.1	Flow insertion	50
4.7.2	Temperature measurements	52
4.7.3	Terminal velocity of spheres	52
5	Free-fall dynamics of highly inertial ellipsoids: Single camera tracking	54
5.1	Introduction	54
5.2	Structure of the code, Python and crucial libraries	55
5.3	Python implementation	56
5.4	Matlab implementation	57
5.5	Post processing and plotting	59
5.6	Results	59
5.6.1	Data set	59
5.6.2	Terminal velocity	60
5.6.3	Steady-state orientation and rotation rates	63
5.7	Conclusions	68
6	Free-fall dynamics of highly inertial ellipsoids: Multi-camera tracking	75
6.1	Introduction	75
6.2	Analyzing frames: extraction of ellipsoid parameters	76
6.3	Pre-processing of data: estimating camera temporal offset	77
6.3.1	Line-of-sight projection	77
6.4	3D-tracking: trajectory of the particle flight	77
6.5	Pre-processing of data: extraction of camera frames and thresholding	78
6.5.1	Cropping	79
6.5.2	Calculating particle contours	79
6.5.3	Method 1: Ellipse fit	82
6.5.4	Method 2: Convex hull	82
6.5.5	Method 3: Alpha shape	82
6.5.6	Scaling and final touches	82
6.6	Orientation determination	84
6.6.1	3D model, Euler rotations and projection images	84
6.6.2	Creation of reference libraries	86
6.6.3	Cost calculation	86
6.6.4	Minimizing cost: solvers and optimization approaches	87
6.7	Validation/Verification of the orientation determination	92
6.8	Results	94
6.8.1	Data set	94
6.8.2	Particle trajectories	95
6.8.3	Transient dynamics	99
6.9	Conclusions	103
7	Conclusion	104
7.1	Summary	104
7.2	Outlook	104
7.2.1	Different particles and particle combinations	104
7.2.2	Setup improvements	105
7.2.3	Lattice-Boltzmann simulations	105
7.2.4	Final remarks	109
	Bibliography	110
	Appendix A Data sheets	117
	Appendix B Insertion test 3	120

List of Figures

1.1	Saharan dust spreading across the tropical North Atlantic Ocean	1
1.2	The most common cloud types on Earth with their typical heights and latitudes	2
1.3	Dependency of C_D on Re_p for a sphere	4
1.4	Settling behavior regimes	5
1.5	Parameter space of heavy anisotropic particles	6
1.6	DNS of point particles in turbulent channel flow	7
1.7	Examples of solid and icy particles found within Earth's atmosphere	10
2.1	Satellite droplets formed from the liquid thread after ejection of the primary droplet	16
2.2	Liquid reservoir with constant liquid level	17
2.3	Schematic overview of the initial droplet generator design	19
2.4	The droplet generator setup	20
2.5	The first droplet generator design	20
2.6	DOD and CJ droplet generation with version 1 of the droplet generator	21
2.7	The final droplet generator design	22
2.8	Updated droplet generator setup	23
2.9	Wave forms for both operation modes of the droplet generator	23
2.10	Operation modes of the droplet generator	24
2.11	Orifice test print	24
3.1	Schematic overview of the printing process	27
3.2	Result of the parameter sweep	30
3.3	Nozzle test print	32
3.4	Nozzle prototypes	33
3.5	Nozzle and droplet generator test prints	33
3.6	Final nozzle designs	34
3.7	Nozzles used with the self-built droplet generator	34
3.8	The nine shapes of ellipsoids created using TPP	35
3.9	Ellipsoids created using TPP	37
3.10	Hexagonal ice crystal created as a demonstrator	38
4.1	Drop tower and insertion mechanism	40
4.2	Drop tower components	42
4.3	Insertion mechanism	43
4.4	VEO camera spectrum response	46
4.5	LED spectrum	46
4.6	Calibration approach multi-camera setup	48
4.7	Results of insertion test 3: Sealed drawer, big particle	51
4.8	Effect of LEDs on observation volume temperature	52
5.1	Flow chart of the single-camera tracking code	55
5.2	Example of ellipse fitting results created with the Matlab implementation	59
5.3	Example ellipsoids as recorded 20 mm after insertion	63
5.4	Example ellipsoids as recorded 70 mm after insertion	64

5.5	Particle rotation rate as a function of median change of particle orientation angle Θ_p over time, EL 0.2, data set 1	65
5.6	Particle rotation rate as a function of median change of particle orientation angle Θ_p over time, EL 0.5, data set 1	66
5.7	Particle rotation rate as a function of median change of particle orientation angle Θ_p over time, EL 1.0, data set 1	67
5.8	Particle track of shape EL 0.2, FL 1.0, experiment 2	68
5.9	Recorded angles of experiment 2 for shape EL 0.2, FL 1.0	70
5.10	Particle rotation rate as a function of median change of particle orientation angle Θ_p over time, EL 0.2, data set 2	72
5.11	Particle rotation rate as a function of median change of particle orientation angle Θ_p over time, EL 0.5, data set 2	73
5.12	Particle rotation rate as a function of median change of particle orientation angle Θ_p over time, EL 1.0, data set 2	74
6.1	Flow chart of the multi-camera tracking code	76
6.2	Trajectory extracted from a particle with shape EL 0.5, FL 0.2	78
6.3	Intensity profile of a single line of a camera frame containing an ellipsoid	80
6.4	Effect of different particle contour fitting options	81
6.5	Results of the different contour fitting methods	83
6.6	Orientation determination solver procedure	84
6.7	Projection images generated from a 3D model of an ellipsoid	85
6.8	Visualization of the difference and overlap between two ellipses	87
6.9	Operations performed by the Nelder-Mead method	89
6.10	Effect of cost calculation type on orientation determination	94
6.11	Trajectory extracted from particle with shape EL 0.2, FL 1.0	96
6.12	Trajectory extracted from particle with shape EL 0.5, FL 0.5	97
6.13	Trajectory extracted from particle with shape EL 1.0, FL 0.5	98
6.14	Particle orientation	99
6.15	Particle flight composite image and extracted angles with cost for a prolate spheroid (EL 0.2, FL 1.0)	100
6.16	Particle flight composite image and extracted angles with cost for a bulky ellipsoidal particle (EL 0.5, FL 0.5)	101
6.17	Particle flight composite image and extracted angles with cost for a disc-shaped particle (EL 1.0, FL 0.5)	102
7.1	Lattice-Boltzmann simulations for several ellipsoids	106
B.1	Results of insertion test 3: Sealed drawer, big particle	121

List of Tables

1.1	Particle relaxation times	8
3.1	Slicing distances (SD) that were chosen for the parameter sweep	30
3.2	Maximum, minimum, average and standard deviation surface roughness values . .	31
5.1	Distances d_{95} to reach 95% of the terminal velocities v_t and particle Reynolds number for each shape.	54
5.2	Single-camera tracking: Amount of experiments per shape	60
5.3	Comparison between the recorded terminal velocities and the empirical drag model of Bagheri and Bonadonna [2016]	62
6.1	Different ways of calculating ellipse overlap cost	92
6.2	Multi-camera tracking: Amount of experiments per shape	95
7.1	Comparison of terminal velocities for ellipsoidal particles	108

Acknowledgements

I would like to thank my supervisor, Prof. Bodenschatz, as well as Dr. Molacek and Dr. Bagheri for their many helpful comments and inspiration. A big thank you also goes to my colleagues, many whom I consider good friends. I'd further like to thank my family and especially my wife, Jasmin, who was there for me every step of the way.

Disclosure: The vast majority of this work was done by the author alone. Exceptions are:

1. The Matlab implementation in chapter 5.4 was created by Dr. Jan Molacek
2. The code in chapter 6 was jointly developed with Dr. Gholamhossein Bagheri. The initial implementation were then improved upon. Specifically, Dr. Bagheri provided the initial implementations for
 - (a) Analyzing frames: extraction of ellipsoid parameters, Ch. 6.2
 - (b) Pre-processing of data: estimating camera temporal offset , Ch. 6.3
 - (c) 3D-tracking: trajectory of the particle flight, Ch. 6.5.2
3. The Lattice-Boltzmann simulations shown in chapter 7.2.3 were carried out by Dr. Tara-prasand Bhowmick.

Funding: This project has received funding from the European Union Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie Actions, Grant Agreement No. 675675.

Chapter 1

Introduction

1.1 Clouds and particles in Earth's atmosphere

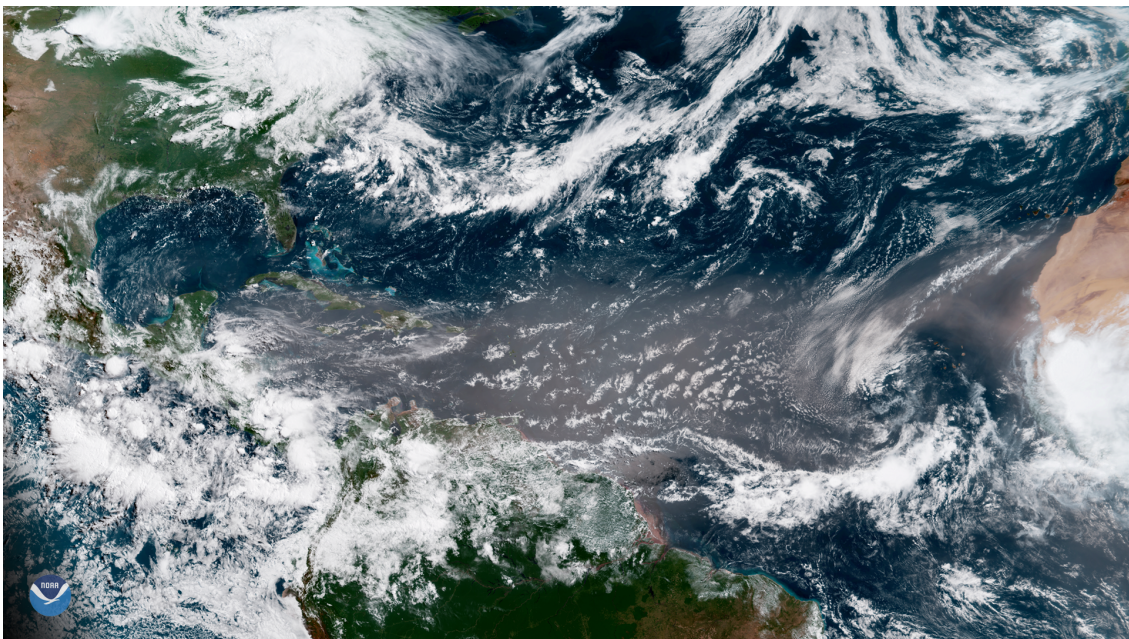


Figure 1.1: Saharan dust across the tropical North Atlantic Ocean; credit: NOAA GOES, June 27, 2018

Clouds are ubiquitous in nature and cover the globe by roughly two thirds at any given time [Boucher et al., 2013]. Particles are similarly common and found in Earth's atmosphere as ice crystals, smoke, dust or pollen. As seen in figure 1.1, both cloud and solid particles can be transported over wide distances, potentially traveling thousands of kilometers away from their center of origin. At the same time, the cloud constituents, such as individual water droplets, ice- or matter particles are typically found in the sub-micrometer to micrometer scale. While cloud droplets with diameters below 500 micrometers mostly have a spherical shape [Houze Jr, 2014], ice and particulate matter can exhibit a wide variety in shapes and (ir-)regularities [Pruppacher and Klett, 1980].

Clouds and particles heavily influence Earth's energy budget and contribute to the largest uncertainties in climate prediction models [Boucher et al., 2013]. Clouds form when the air becomes cool and moist enough to sufficiently super-saturate and activate on available cloud condensation

nuclei (CCN). The droplets first grow by condensation and later through other interactions, possibly due to clustering caused by turbulent air currents [Bodenschatz et al., 2010], [Pruppacher and Klett, 1980]. Many types of clouds exist with differing shapes, heights and compositions. As a rough distinction, clouds can be divided into warm clouds that contain only liquid water in the form of droplets at temperatures above 0°C and into clouds that are composed of ice crystals at about -38°C and lower. Mixed phase clouds combine the two options and are found in the temperatures in-between [Koop et al., 2000], [Boucher et al., 2013], [Pruppacher and Klett, 1980]. An overview of the most common cloud types is shown in figure 1.2.

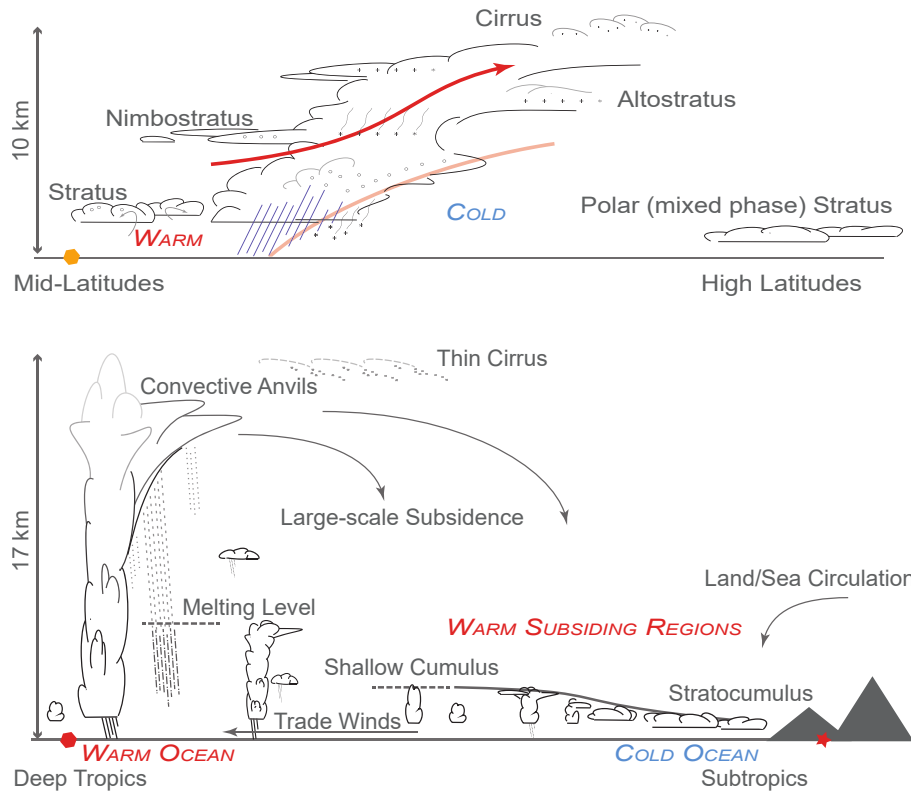


Figure 1.2: The most common cloud types on Earth with their typical heights and latitudes. They can vary wildly in their vertical extend and are typically composed of water at temperatures above 0°C , ice at about -38°C and lower. Mixed phase clouds that exhibit both phases are found the temperatures in-between. From Boucher et al. [2013].

Clouds involve a huge variety of scales, from sub-micrometer particles that act as cloud condensation nuclei (CCN) to turbulent mixing and entrainment processes, up to coherent atmospheric flows that can span tens to thousands of kilometers [Bodenschatz et al., 2010], [Davis et al., 1999]. This causes challenges for both simulations and theory as most climate models cannot resolve scale differences on that order. Several options are available to study clouds numerically but are typically restricted to a certain scale: Direct numerical simulations (DNS) offer the highest resolution and can resolve particle interactions on the scale of a few meters or less. However, they cannot span processes on meteorological scales. For these, several other options are routinely utilized: Large-eddy simulations (LMS), cloud-resolving models (CRM) and, on the largest scales, global circulation models (GCM). The intricate microphysical behaviors of droplets and micron sized particles are then often reduced to sub-grid parametrizations. Consequently, nuanced interactions of droplet/particle-clouds are only weakly represented [Boucher et al., 2013].

Combining *in situ* and laboratory work with the simulations, however, can enhance our understanding significantly. Different approaches exist, ranging from satellite observations to airborne, ship-based and field-site studies (for a recent example, see the EUREC4A campaign [Bony et al., 2017]) to controlled experiments. Studying small-sized droplets and particles in a laboratory setting necessitates high precision in both imaging and manufacturing, as will become evident in this work.

Micrometer-sized droplets and particles are also of major importance in many other scientific and engineering disciplines. Liquid droplets are used for drug delivery [Dames et al., 2007], combustion [Sornek et al., 2000], and melt atomization to produce metal and alloy powder [Ashgriz, 2011]. They further play a prominent role in the transmission of viruses and bacteria as sneeze [Han et al., 2013] or cough [Bourouiba et al., 2014]. Small solid matter clusters together to form planetesimals (precursors of fully grown planets) in protoplanetary discs [Capelo et al., 2019]. Comets such as 67P/Churyumov–Gerasimenko, recently visited by the ROSETTA spacecraft, are thought to have formed similarly and reveal intricate fractal patterns of particle aggregates [Hilchenbach et al., 2016], [Mannel et al., 2016]. In the ocean, marine snow contains organic aggregates and microorganisms like phytoplankton [Alldredge and Silver, 1988], providing a source of nutrition to the sea life food chain. In the form of microplastics, small particles pose a growing threat to the marine environment as they are consumed by sea life and might become a health risk for humans by ingestion [Desforges et al., 2015]. Microplastics further pose a potential respiratory danger for public health when suspended in air [Kim et al., 2015].

In this work, the focus is on Earth’s atmosphere. The dynamics of its droplets and non-spherical particles will be covered in the following.

1.2 Aerodynamics of spherical particles

Spherical objects, such as the typical warm cloud water droplets, are counteracted in their free-fall by the drag force F_D [Bagheri and Bonadonna, 2016], [Loth, 2008]:

$$\mathbf{F}_D = -\frac{1}{2}\rho_f C_D A |\mathbf{v}_p - \mathbf{v}_f| (\mathbf{v}_p - \mathbf{v}_f). \quad (1.1)$$

Here, ρ_f is the fluid density, C_D the drag coefficient of the particle, and A is a reference area that is related to the particle size, in the case of sphere as follows: $A = \pi d^2/4$. \mathbf{v}_p and \mathbf{v}_f are the particle and fluid velocities, respectively.

The terminal velocity v_t marks the fastest falling velocity the particle will assume when gravity is balanced with \mathbf{F}_D . It can be obtained by replacing \mathbf{F}_D with gravity force acting on the particle for a settling case and with a buoyancy force in case the particle has a lighter density than the surrounding fluid. The drag coefficient C_D depends on many parameters and is the most challenging parameter to be determined in equation 1.1. However, only the particle Reynolds number and particle-to-fluid density ratio have a first order influence on C_D [Loth, 2008], [Hölzer and Sommerfeld, 2008], [Brosse and Ern, 2013], [Bagheri and Bonadonna, 2016].

The particle Reynolds number is then defined as:

$$Re_p = \frac{\rho_f d_{eq} |\mathbf{v}_p - \mathbf{v}_f|}{\mu_f}, \quad (1.2)$$

where d_{eq} is the *volume equivalent* diameter and μ_f is the fluid dynamic viscosity. Using d_{eq} , the particle Reynolds number can be calculated also for non-spherical particles. This is done by determining their volume first and then calculating d_{eq} as if the particle in question was a sphere. Re_p characterizes the flow structure around the particle, depending on a characteristic length,

viscosity and velocity. Figure 1.3 shows the typical development of turbulent wake structures behind a sphere as a function of Re_p together with the corresponding drag coefficient.

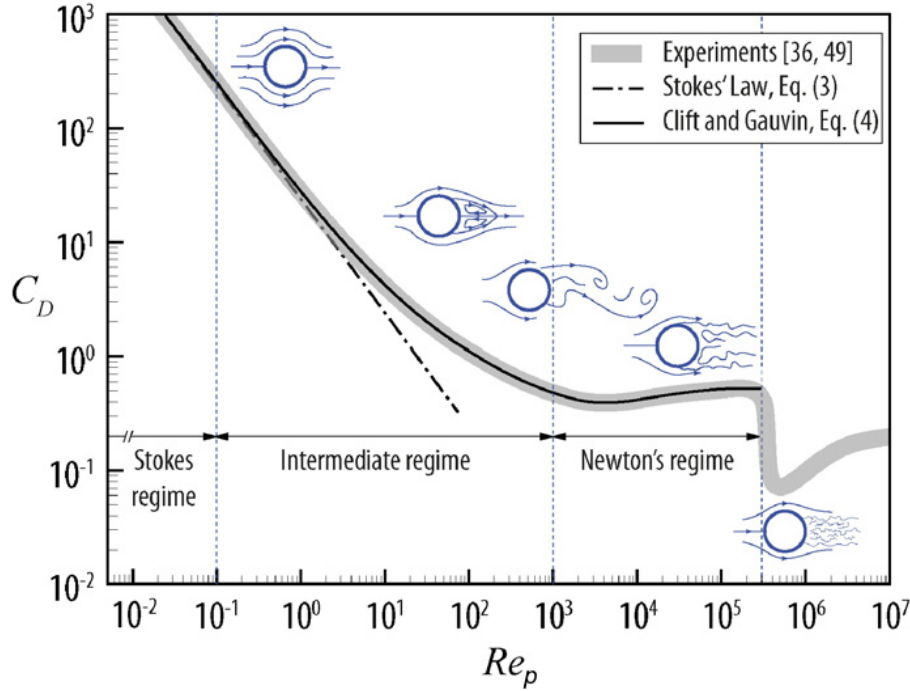


Figure 1.3: Dependency of C_D on Re_p for a sphere. From Bagheri and Bonadonna [2016]

With increasing Re_p different regimes are encountered by the sphere:

- $Re_p \ll 1$: "Stokes' regime". Completely laminar flow around the sphere. The only regime where an analytic solution exists, given by Stokes [1851]:

$$C_D = \frac{24}{Re_p} \quad (1.3)$$

- $0.1 < Re_p < 1000$: "Intermediate regime". Marks the beginning of flow separation at the particle wake that eventually transitions into full vortex shedding.
- $1000 < Re_p < 3 \times 10^5$: "Newton's regime". The wake behind the sphere are now fully turbulent [Clift et al., 2005].
- $Re_p > 3 \times 10^5$: "Supercritical regime". Both the wake and the boundary layer are turbulent. $Re_p < 3 \times 10^5$ is called the *critical Reynolds number*, at which point the *drag crisis* occurs and the drag is suddenly reduced by a large factor¹.

According to Bagheri and Bonadonna [2016], for Re_p that lie below the critical Reynolds number, the most accurate model for the prediction of the drag coefficient C_D is given by Clift and Gauvin [1971]. The error compared to measured values is then within 6% and C_D is:

$$C_D = \frac{24}{Re_p} (1 + 0.15 Re_p^{0.687}) + \frac{0.42}{1 + \frac{42500}{Re_p^{1.16}}} \quad \text{for } Re_p < 3 \times 10^5 \quad (1.4)$$

¹[Loth, 2008], [Clift and Gauvin, 1971], [Clift et al., 2005] and [Achenbach, 1972]

1.3 Aerodynamics of non-spherical particles

Considering the drag coefficient for non-spherical particles, the parameters with first order influences are again the particle Reynolds number Re_p and the particle-to-fluid density ratio. Compared to the sphere, however, two more parameters have first order influences: the shape and the orientation. Bagheri and Bonadonna [2016] mentions that the average of the drag coefficient of a falling non-spherical particle is generally higher than that of a sphere. In Stokes' regime, the most likely outcome during the fall of non-spherical particles is that of random orientation [Happel and Brenner, 1983], [Loth, 2008], [Clift and Gauvin, 1971], [Albertson, 1953]. Contrary to that, Cox [1965] showed that a freely falling spheroid with small eccentricity falls with the largest projection area towards the settling direction. From $Re_p \approx 100$, particles tend to fall with the largest projection area towards the settling direction ([Clift et al., 2005], [Marchildon et al., 1964] and [Komar and Reimers, 1978]), with wake instabilities starting to develop in the earliest case at $Re > 50$ for cylinders [Jayaweera and Mason, 1965].

The particle-to-fluid density ratio $\rho' = \rho_p/\rho_f$ (where ρ_p is the particle density) can significantly affect orientation and secondary motions of particles in Newton's regime $Re_p > 1000$. As ρ' increases, the secondary motion of particles increases and this leads to a reduction of the average projected area of the particle during falling. In turn, the drag coefficient is reduced [Bagheri and Bonadonna, 2016]. Most studies of falling particles have been performed in the range $1 < \rho' < 15$, significantly lower than the ρ' for particles falling in air where it is $O(10^3)$. Ern et al. [2012] covered path instabilities of simple, homogeneous, mostly axisymmetric or two-dimensional (2D) bodies such as disks, flat cylinders, spheroids and 2D plates in a review paper on experimental observations and numerical simulations. They found that flat bodies are known to present far more spectacular path instabilities than spheres. Four regimes that describe the settling behavior of particles [Ern et al., 2012]:

- A: mostly chaotic with small amplitude of oscillations
- B: period harmonic, oscillating a mean vertical direction with planar trajectories
- C: period anharmonic, oscillating a mean vertical direction with helical trajectories
- D: highly non-linear, i.e. chaotic and strongly oscillating

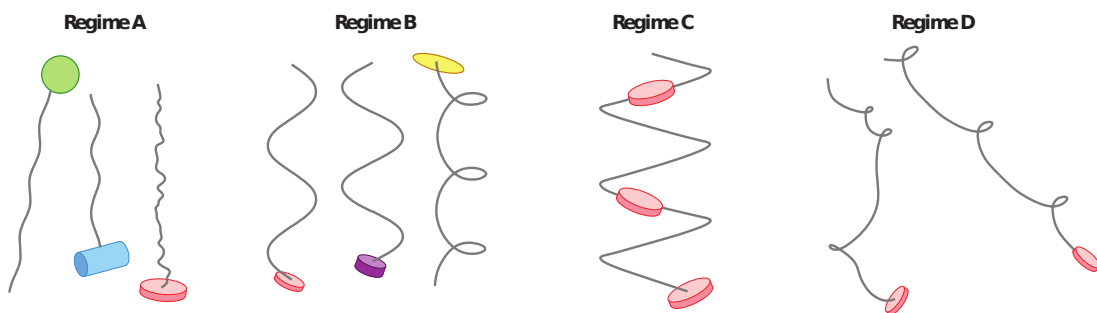


Figure 1.4: Settling behavior regimes. A: mostly chaotic, B: period harmonic, C: period anharmonic, D: highly non-linear. From Ern et al. [2012].

Regimes A-D are also shown in figure 1.4. By performing settling experiments, Bagheri and Bonadonna [2016] measured the drag of falling volcanic ash particles and 3D-printed ellipsoids. They then introduced a new empirical drag coefficient model that is valid for any particle shape:

$$C_D = \frac{24k_S}{Re_p} (1 + 0.125(Re_p k_N/k_S)^{2/3}) + \frac{0.46 k_N}{1 + 5330/(Re_p k_N/k_S)} \quad (1.5)$$

where k_S and k_N are drag correction coefficients for Stokes' and Newton's regimes. Equation 1.5 is associated with an mean error of 9.8% and a maximum error of 73.4% for $Re_p < 3 \times 10^5$. According to the authors, the high maximum error stemmed mostly from cases where the particle orientation differed strongly from an assumed random orientation or when the particle shape was irregular and, hence, could not be represented by an ellipsoidal shape. As another important factor for uncertainties, the particle-to-fluid density ratio ρ' was mentioned.

Figure 1.5, left side, shows a conceptual overview of the parameter space of heavy anisotropic particles in turbulent flows by Voth and Soldati [2016]. Here, the keyword 'heavy' indicates particles with an approximate particle-to-fluid density ratio $\rho' \approx 1000$, which would be on the order of solid particles falling through air, as found in Earth's atmosphere. On the x-axis, the shape of the particles is given as the aspect ratio λ which is the ratio between the semimajor and the semiminor axis of an axisymmetric ellipsoidal particle. A schematic on how the aspect ratio is calculated, together with example shapes of axisymmetric ellipsoidal particles, such as an oblate disc, a sphere and a prolate, is shown on the right side of the figure. The y-axis of the parameter space is the particle size as a ratio of the largest dimension of the particle, 'L', and the *Kolmogorov length scale*, ' η ', the smallest length scale of turbulence, at which point the energy of turbulence is dissipated into heat. It is defined as $\eta = (\frac{\nu^3}{\epsilon})^{\frac{1}{4}}$, where ν is the kinematic viscosity of the surrounding fluid and ϵ the average rate of dissipation of turbulence kinetic energy per unit mass [Kolmogorov, 1941].

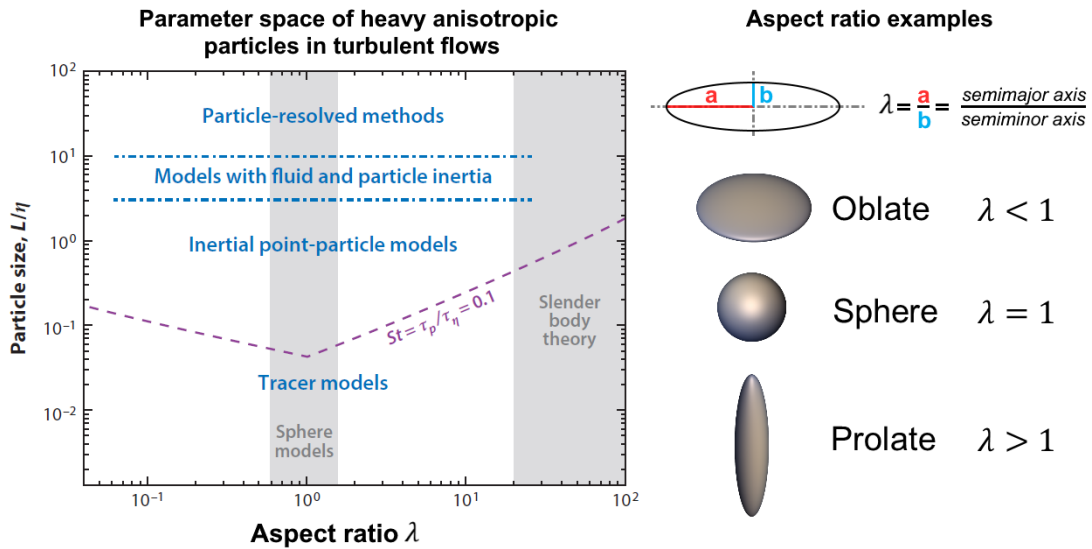


Figure 1.5: Parameter space of heavy anisotropic particles and examples of typical shapes that can be represented by aspect ratio λ . The purple dashed line marks the Stokes number $St = \tau_p/\tau_\eta = 0.1$. Here, τ_p and τ_η are the particle relaxation time and the Kolmogorov time scale, respectively, as introduced later within the chapter. The gray bars represent the aspect ratios for which models are available, namely sphere ($\lambda = 1$) and strongly elongated shapes, so called *slender bodies* ($\lambda \gg 1$). All represented shapes are axisymmetric ellipsoids, hinting at a lack of models for non-axisymmetric or irregular shapes. Adapted from Voth and Soldati [2016]

The blue annotations in figure 1.5 show approximate lengths that are suitable for different simulation models. The authors stress that the quantitative determination of the parameters at which point specific models begin to fail depend on the quantities of interest and the degree of accuracy that is desired. Nonetheless, this graphic provides a suitable conceptual overview of the currently applied models. For small particles sizes, tracer models are suitable. Here, 'tracers' are particles small enough to follow the flow perfectly without influencing the surrounding flow structure ($Re_p \ll 1$). With increasing length, the interaction between the particle and the flow becomes increasingly complex: fluid and particle inertia have to be considered and particle-fluid interac-

tions between the surrounding media and the particle takes place. A concise description of the different models that are available for modelling particles in turbulent flows is given in the paper by Voth and Soldati [2016].

Particularly important here, however, are the two gray bars found in the center and the right side of the parameter space. These indicate the approximate aspect ratios for which analytical solutions have been found and the corresponding models that rose from them. Only a narrow region around the spherical case ($\lambda = 1$) and strongly elongated shapes ($\lambda \gg 1$) are marked. This means that for the majority of shapes, no such solutions are available at present, pointing at a clear gap in our current understanding. Furthermore, the shapes represented in figure 1.5 are all axisymmetric ellipsoids, hinting at the issue at hand that non-axisymmetric or even irregular particles are currently not well described.

Also shown in figure 1.5 is the Stokes number, which is defined as follows:

$$St = \frac{\tau_p}{\tau_\eta} = \frac{\text{Particle relaxation time}}{\text{Kolmogorov time scale}} \quad (1.6)$$

where the Kolmogorov time scale $\tau_\eta = (\frac{v}{\epsilon})^{\frac{1}{2}}$ functions analogous to the Kolmogorov length scale and states the short time scale at which viscosity dominates and turbulent kinetic energy is dissipated into heat [Kolmogorov, 1941]. The particle relaxation time will be introduced momentarily.

The Stokes number is a measure of a particle's inertia and as such indicates how quickly the particle can respond to changes of the surrounding flow structure. A particle with a very small Stokes number ($St \ll 1$) follows the surrounding flow perfectly and is often called a tracer particle. On the other hand, particles with large Stokes numbers ($St \gg 1$), do not follow the flow precisely and might detach completely from the flow's streamlines, especially when the surrounding flow changes direction quickly. In this case, the particle's inertia would transport it forward in the direction of its original trajectory [Toschi and Bodenschatz, 2009, Pumir and Wilkinson, 2016].

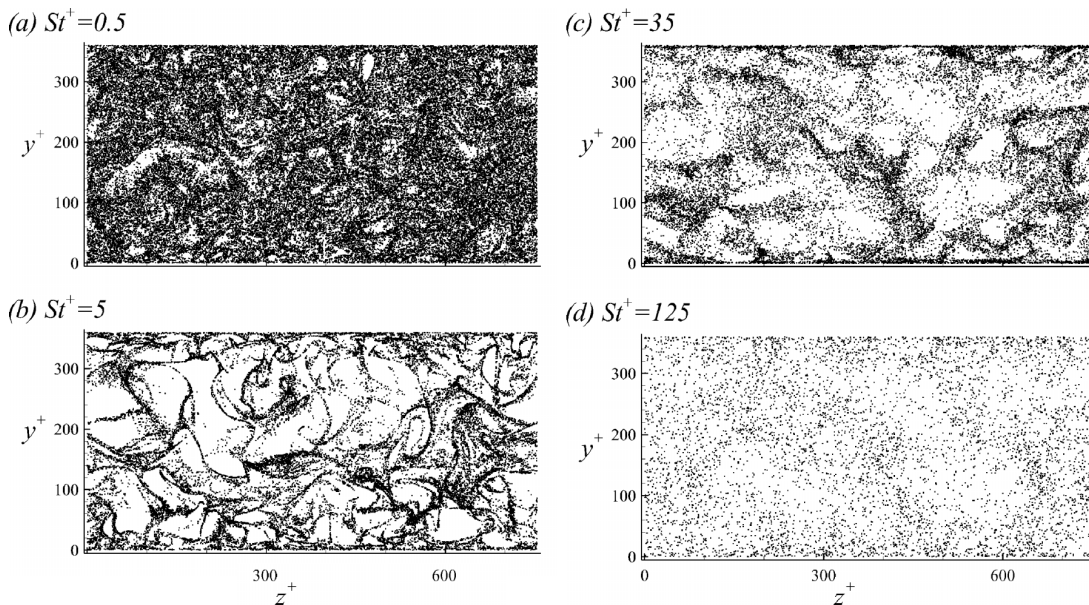


Figure 1.6: Direct numerical simulations (DNS) of point particles in turbulent channel flow with (flow) Reynolds number $Re_f = 180$. Different Stokes numbers St^+ , based on wall-units, lead to widely different clustering of the particles. From Lee and Lee [2015]

Figure 1.6 by Lee and Lee [2015] shows the effect that particle inertia can have on clustering of the particles in a turbulent flow. In the direct numerical simulation (DNS), a turbulent channel flow is established and point particles, i.e. shapeless particles, with varying Stokes numbers St^+ (0.5, 5, 35, 125) are inserted into the flow. Note that Lee and Lee [2015] used a variation of the Stokes number that is based on wall-units, i.e., the friction velocity and the kinematic viscosity. Initially, at $St^+ = 0.5$, the particle distribution is relatively homogeneous with some visible structures. This changes dramatically as the Stokes number is increased to $St^+ = 5$, where clear structures caused by particle clustering are visible. As the Stokes number is increased further, the particles have more and more difficulty following the turbulent flow structure and first display a mixture of homogeneous distribution and flow structures at $St^+ = 35$ to then then be mostly homogeneously distributed at $St^+ = 125$.

Clustering of water droplets due to turbulence is thought to play a major role in rain-formation due to droplet collisions and subsequent coalescence [Bodenschatz et al., 2010]. On the other hand, clustering of non-spherical particles and their interaction with turbulence in general, might influence the time these particles can remain suspended in the atmosphere and be transported to faraway locations. However, at the time of this writing, this interaction remains largely unexplored [Voth and Soldati, 2016].

As is evident, particle clustering depends strongly on the Stokes number. The Stokes number, however, in turn depends on the particle relaxation time. Two particle relaxation times exist: the translational particle relaxation time τ_p and the rotational relaxation time τ_r , where τ_p is the more widely known one. Table 1.1 shows the currently used particle relaxation times for oblates ($\lambda < 1$), spheres ($\lambda = 1$) and prolates ($\lambda > 1$).

Shape:	Oblate	Sphere	Prolate
Aspect ratio:	$\lambda < 1$	$\lambda = 1$	$\lambda > 1$
Translational:	$\tau_p = \frac{2}{9} \frac{\rho_p r^2}{\mu_f} \frac{\lambda \ln(\lambda + \sqrt{\lambda^2 - 1})}{\sqrt{\lambda^2 - 1}}$	$\tau_p = 2\rho_p r^2 / 9\mu_f$	$\tau_p = \frac{2}{9} \frac{\rho_p r^2}{\mu_f} \frac{[\pi - 2 \tan^{-1}(\lambda(1 - \lambda^2)^{-\frac{1}{2}})]}{2(1 - \lambda^2)^{\frac{1}{2}}}$
Rotational:	N/A	$\tau_r = \frac{2}{15} \rho_p r^2 / \mu_f$	N/A

Table 1.1: Translational and rotational particle relaxation times, according to particle shape. Here, λ is the particle aspect ratio, ρ_p is the particle density, r the particle radius and μ_f the fluid dynamic viscosity. Currently, rotational particle relaxation times for shapes other than spheres have not been determined. Source: Voth and Soldati [2016] and references therein.

Both particle relaxation times are measures for how quickly a particle can 'relax' to its natural state and describe translational and rotational slip between the particle and the fluid [Voth and Soldati, 2016]. The lack of rotational particle relaxation times for shapes other than spheres, together with the the lack of models for non-spherical shapes, constitutes a clear gap in knowledge that is important to be closed. The rotational particle relaxation time is mainly of importance when particles are not of spherical shape as it determines how long particles take to assume their preferred orientation towards an incoming flow. Should the time be longer than the shortest time frame of the flow, i.e. the Kolmogorov time scale, the particle might never relax to its preferred orientation. If, on the other hand, the relaxation time is much shorter than the Kolmogorov time, the particle might always assume its preferred orientation. This directly influences the drag the particle experiences and, in turn, the time it can remain suspended in the turbulent flow.

Adding to the complexity of the particle-fluid interaction in turbulence is the so called *coupling*. There is a distinction between 1-way, 2-way and 4-way coupling, depending on the volume fraction of particles in the fluid, $\Phi_p = MV_p/V$, where M is the number of particles, V_p the volume of a single particle and V the volume occupied by both particles and the fluid. The four types of coupling are described by Elghobashi [1994] as follows:

- 1-way: When the volume fraction has a low value, i.e. $\Phi_p < 10^{-6}$, the momentum which is transferred from the particles to the fluid does not significantly affect the turbulent flow. Indeed, the interaction between the flow and the particles only affects the particles, hence the term one-way coupling.
- 2-way: When the volume fraction is in the regime $10^{-6} < \Phi_p < 10^{-3}$, the total momentum that is transferred from the particles to the fluid is large enough so that the turbulent structure is altered. As an example of two-way coupling, consider a particle changing the flow structure in its wake and then another particle interacting with this changed wake structure.
- 4-way: In dense suspensions, where $\Phi_p > 10^{-3}$, particle/particle collisions take place in addition to the already occurring two-way coupling, leading to four-way coupling.

Interestingly, for two-way coupling, particles might either dampen or enhance turbulence, depending on their translational particle relaxation time τ_p . For a fixed Φ_p , as long as the particle material and the surrounding fluid viscosity remain the same, decreasing τ_p effectively means decreasing the particle diameter and, hence, increasing the surface area of the particulate phase. This leads to an increased dissipation rate ϵ of the turbulence energy. Conversely, increasing τ_p for a fixed Φ_p translates to an increased particle diameter and with it an increased particle Reynolds number Re_p . Once $Re_p > 400$, vortex shedding takes place in the wake of the particle, causing an enhanced production of turbulence energy [Elghobashi, 1994].

In conclusion, the highly non-linear nature of particle-fluid interactions such as clustering and coupling, as well as the vast variety in possible particle shapes constitute major challenges when it comes to modelling turbulent flows. In conjunction with other measures (simulations, field studies), a thorough experimental investigation of particle settling behavior and particle relaxation times is needed. To underline the importance and difficulty of the task, figure 1.7 shows just a few shapes of the multitude of particle shapes that are found within Earth's atmosphere.

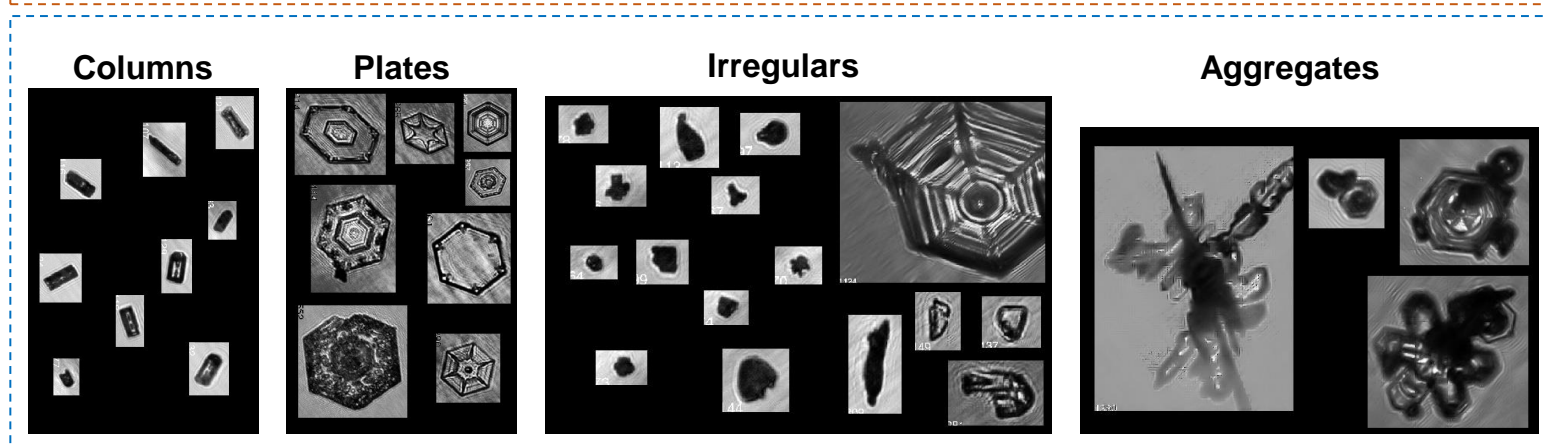
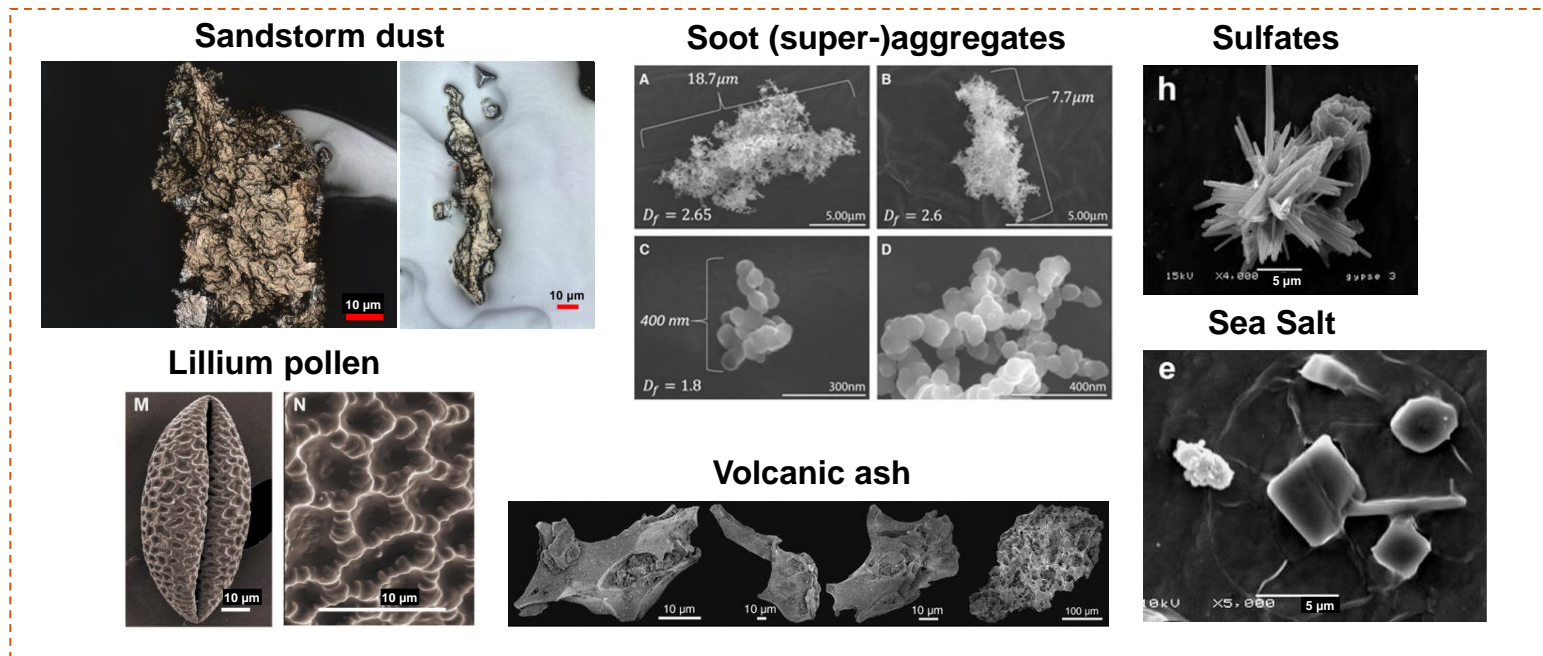


Figure 1.7: Examples of solid and icy particles found within Earth's atmosphere. The particles show a wide array of shapes and sizes, ranging from regular to irregular. Please note that the source for the ice crystals did not include a scale bar. However, their typical size range is found between 10 – 1000 μm . Sources: **Sandstorm dust:** MPI DS, **Soot aggregates:** Chakrabarty et al. [2014], **Sulfates and sea salt:** Bourotte et al. [2006], **Lillium pollen:** Du et al. [2014], **Volcanic ash:** Nurfiani and Bouvet de Maisonneuve [2018], **Ice particles:** University of Manchester - Centre for Atmospheric Science website <http://www.cas.manchester.ac.uk/resactivities/cloudphysics/background/ice/>, accessed 02.04.2021

1.4 Objectives of this work

The main goal of this work was to explore the dynamics of atmospheric microphysics. It started out with a focus on building a droplet generator in order to perform drop-on-drop collision experiments. However, it was soon realized that the recently commercially available Two-Photon-Polymerization (TPP) printing held an opportunity for the study of the settling dynamics of precisely defined 3D-printed particles in so far unreachable regimes in terms of size and density ratio to the settling medium (air). The focus of the work was then switched to the construction of a settling experiment and corresponding analysis software.

This thesis consists of five main chapters and a conclusion. **Chapter 2** describes the design, creation and testing of a droplet generator. The generator was able to produce droplets with diameters $< 100 \mu\text{m}$ in two different operational modes: Drop-on-Demand (DOD) and Continuous Jetting (CJ). It was designed so that it could hold a wide array of different nozzles and be connected to different mounts and reservoirs.

Chapter 3 dives into the intricacies of using two-photon-polymerization (TPP) for the creation of miniature parts with feature sizes of $1 \mu\text{m}$ and less. After performing an extensive parameter search, nozzles for the droplet generator from chapter 2 and ellipsoidal particles are produced.

In **Chapter 4** a settling chamber that was designed and built from scratch is described. The chamber consists of a drop tower and an insertion mechanism for particles that were created using TPP. The insertion mechanism had to be validated to be suitable for the settling particle experiments. Additionally, a shadowgraphic setup was built around the settling chamber.

To analyze the results of the settling particle experiments, a code is written and introduced in **Chapter 5**. It uses a single-camera view to extract and track particles and determine their orientation. Rotation rates are covered and the measured terminal velocities are compared against the empirical drag coefficient model from Bagheri and Bonadonna [2016].

Finally, in **Chapter 6**, the idea of the code from chapter 5 is expanded upon and a multi-camera tracking code is written. It features full 3D-tracking and orientation determination by taking the views of two cameras into account that are positioned 90° apart from each other. A 3D model is used as a base of comparison and projection images of the 3D model are fed into a solver that attempts to minimize the differences between the camera images and the projection images. Tracking the orientation of the particles during their flight is a crucial step in determining their drag and particle relaxation times. First results are presented.

The thesis is then wrapped up with a conclusion and an outlook in **Chapter 7**.

Chapter 2

Design of a droplet generator for the generation of micrometric droplets

2.1 Introduction

A first step to study cloud droplets in a laboratory setting is the use of a droplet generator to dispense droplets with well characterized size. Different designs exist, ranging from spinning discs [Bewley et al., 2013], condensation aerosol generators [Sinclair and La Mer, 1949], to continuous jet (CJ) [Hoath, 2016b] and drop-on-demand (DOD) [Castrejón-Pita et al., 2012] generators. An overview will be given in chapter 2.2; for a more detailed recent review the reader is referred to Ashgriz [2011]. Most droplet generators aim to restrict the droplet diameter to a specified size and create a mono-disperse droplet distribution. The diameter of the droplets is directly proportional to the nozzle orifice diameter and of the same order of magnitude. Typically, nozzles are manufactured from glass or synthetic sapphire [Yang et al., 1997], from metal using traditional methods such as milling, drilling and erosion, or from PDMS (Polydimethylsiloxan), when part of a microfluidic chip [Bransky et al., 2009]. These approaches have been widely successful but are restrictive in terms of nozzle orifice design, as in the majority of cases the orifice consists of a simple round hole.

In order to study atmospheric droplets in an undisturbed environment, e.g. without a surrounding flow, a droplet generator with a screw-in mechanism to use and easily exchange nozzles created by two-photon-polymerization (TPP) was built. TPP provides a new approach and a high degree of freedom for the design of miniature nozzles. Compared to commercially available droplet generators, this allows for fast prototyping and iteration of nozzle designs.

2.2 Generating droplets

Most droplet generators follow a similar general design consisting of a mechanical housing that contains all parts, a liquid reservoir with a filter at the inlet, an actuator, for example a piezoelectric element to either transport the liquid or change the breakup behaviour of the liquid thread, and a nozzle with a small opening - the orifice - through which the liquid exits the generator. Interfacing the macro-scale housing to the micrometer sized orifices of the nozzles can prove difficult and requires a sound choice of sealing. Several different modes of operation exist, influencing the design of such a droplet generator. According to Dong et al. [2006], droplet generation can be roughly divided into three basic types:

1. **Dripping:** Caused by gravitational effect on a liquid capillary column with a low flow rate. It takes relatively long to generate drops this way, with time scales needed for the

generation of a single drop ranging from seconds to minutes.

2. **Continuous jetting (CJ):** Increasing the flow rate through a dripping capillary tube leads to the creation of a continuous jet. The jet will eventually break up into droplets due to the Rayleigh-Plateau instability [Rayleigh, 1878]. Although this instability will occur naturally, more sharply mono-disperse size distribution of the resulting droplets can be achieved by enhancing the instability by an actuator with well-chosen oscillation frequency.
3. **Drop-on-demand (DOD) jetting:** In this case, no flow rate is required. Instead, single drops are generated at a set repetition rate by applying short pressure pulses to the liquid filling the capillary. The dynamics of the meniscus that forms the resulting drop can be manipulated by tuning the electric signal that drives the actuator, such that only a single drop is generated.

Of the three options, dripping is the least useful method for generating monodisperse droplets of the size relevant for the desired purpose. Therefore, only the latter two options are considered from now on.

Continuous jetting is a straightforward way of operating a droplet generator. First, the jet is created, for example by a syringe pump which overcomes the surface tension and continuously provides flow through the orifice. Then a Rayleigh-Plateau instability is enforced on the resulting jet with an actuator. The optimal wavelength for the breakup of the jet to create monodisperse droplets is given by:

$$\lambda_{opt} = \sqrt{2}\pi D_{jet} \approx 4.5 D_{jet} \quad , \quad (2.1)$$

where D_{jet} is the diameter of the jet directly upon exiting the orifice and follows the orifice dimensions closely [Rayleigh, 1878]. In practice, monodisperse droplets can be created in a relative wide range around the optimal wavelength [Schneider and Hendricks, 1964]:

$$3.5 D_{jet} < \lambda < 7 D_{jet} \quad . \quad (2.2)$$

The driving frequency of the actuator is ideally chosen to match the optimum wavelength so that

$$f = v_{jet}/\lambda_{opt} \quad , \quad (2.3)$$

where v_{jet} is the velocity of the jet at the orifice. The droplet volume V_d is then given by [Schneider and Hendricks, 1964]

$$V_d = \pi \cdot (D/2)^2 \cdot \lambda \quad , \quad (2.4)$$

with a resulting droplet diameter D_d of

$$D_d = 2 \cdot (0.75 \cdot V_d/\pi)^{\frac{1}{3}} \quad . \quad (2.5)$$

Since the size of the droplets is directly affected by the diameter of the orifice, reducing the orifice size would appear to be the simplest way to reduce the droplet diameter. However, this can lead to problems in the operation of the droplet generator due to the higher pressures required to force the liquid through the small opening, and due to an increased susceptibility to orifice blockage by any micron-sized dust particles.

Advantages of continuous jet (CJ) droplet generators include a straight-forward design and reduced force needed from the piezoelectric element compared to the drop-on-demand (DOD) mode, as the pressure needed for a continuous jet is provided by the pump. In contrast, DOD generators rely solely on the actuator to force a specified amount of liquid through the orifice. In both cases, the force applied must overcome the surface tension σ_{surf} of the liquid and is aided by the hydrostatic pressure (as long as the nozzle is aligned with gravity and the liquid level is above the nozzle orifice). Without a force applied by the actuator to create a pressure wave, the maximum height h that the water reservoir can have above the orifice to still allow a static equilibrium of the meniscus at the orifice is given by:

$$|\Delta h| \leq 4\sigma_{surf}/\rho g D_o \quad (2.6)$$

where ρ is the density of the liquid, g the acceleration due to gravity and D_o the orifice diameter [Harris et al., 2015].

Castrejón-Pita et al. [2012] states that a combination of surface tension and viscosity effects restrict the formation of small drops. Surface tension causes freely falling drops to take the form of a sphere (the configuration that requires the lowest total energy), as long as the cohesion between the fluid's molecules is not overcome by the fluid's inertia. The ratio of the fluid's inertia to its surface tension is often expressed as the dimensionless Weber number We [Frohn and Roth, 2000]:

$$We = \frac{\rho v^2 d}{\sigma_{surf}} \quad (2.7)$$

where d is a characteristic length scale such as the diameter of the jet or drop and v the drop's velocity. Surface tension is further responsible for the breakup of the stream in CJ printing and for the formation of the main drop and satellite drops during DOD printing. Viscosity μ , on the other hand, is the interaction between molecules of the liquid and typically counter-acts regions of liquid moving relative to each other [Hoath, 2016a]. A way of quantifying the effect that viscosity has on droplet formation is the Ohnesorg number Oh . It is defined as follows:

$$Oh = \frac{\mu}{\sigma_{surf} \rho d} \quad (2.8)$$

The value of the Ohnesorg number is closely related to the achievable droplet generation quality. In cases of a too high Ohnesorg number, such as $Oh > 1$, viscous forces prevent the separation of a drop. On the other hand, if the Ohnesorg number is too low at $Oh < 0.1$, a large number of satellite droplets are created when the liquid jet exits the orifice [Hoath, 2016a]. Satellite drops in terms of droplet generation are unwanted and hard to control secondary drops that are created with the primary drop. They are typically polydisperse, meaning that they have heterogeneous diameters, and exit the orifice with varying velocities. Water droplets with a diameter of 100 μm that are created at laboratory conditions ($T = 20^\circ\text{C}$, $p = 1 \text{ atm}$) have an Ohnesorg number of $Oh = 0.012$ and are therefore already very close to the lower limit. As the aim of this work was to create droplets with diameters smaller 100 μm , greater care had to be taken when designing the droplet generator and choosing its driving parameters.

2.2.1 Common droplet generator types

A wide variety of droplets generators are available, many using DOD or CJ, which represent the most common types of droplet generators [Hoath, 2016a], or combinations thereof. A small overview of common methods follows:

- **Piezoelectric.** In this type of droplet generator the actuator is a piezoelectric element. Examples piezoelectric drop-on-demand generators can be found in the works of Yang et al. [1997], Harris et al. [2015], and Bransky et al. [2009]. Piezoelectric elements are combinations of certain materials such as crystals, metals and ceramics that change their shape when an electric voltage is applied. The voltage creates mechanical stress due to different expansion/shrinking rates of the materials. Piezoelectric elements can be acquired in a wide range of sizes and shapes. Advantages include high repeatability without change in performance, as well as precise and fast response (e.g. kHz up MHz) to an applied electric signal. The droplet generator in this work used a piezoelectric element.
- **Thermal expansion.** A resistive heating element is used to heat the fluid within an ejection chamber. This generates a vapor bubble that expands rapidly and provides the pressure needed to eject a droplet [Hoath, 2016a].
- **Flow focusing.** Reducing the diameter of the jet further without using a smaller orifice, where the nozzle is a gas dynamic virtual nozzle (GDVN). In this case, a liquid jet is thinned by focusing a gaseous stream to a point shortly behind the orifice, hence creating a "virtual"

nozzle. Recently, Nelson et al. [2016] used Two-Photon-Polymerization (TPP, a 3D-printing technique introduced in chapter 3) to create a GDVN droplet generator and were able to routinely create jets of diameter 2-5 μm by reducing the diameter of a water jet by a factor of 10. Using a similar approach, sub-micron jets have been demonstrated by Deponte et al. [2011]. In both cases, flow focusing was combined with a piezoelectric device to control the instability that drives the breakup of the fluid jet. With the clear advantage of reduced droplet sizes while retaining a relatively large orifice, one potential disadvantage of flow focusing is that the droplets are moving within a gaseous jet. This makes the approach potentially unsuitable for some applications where the surrounding gas should be at rest, such as the study of droplet collisions.

- **Electrohydrodynamic (E-jet) printing.** E-jet printing uses electric fields to deliver fluids to a target location. A constant voltage difference between the nozzle carrying the fluid and the print substrate attracts the ions in the fluid towards the desired location. E-jet printing has been used for the printing of micron and sub-micron scale droplets using a wide variety of inks. Disadvantages of using E-jet printing typically include unreliable production of monodisperse droplets and, depending on the application, the charging of the droplets [Mishra et al., 2010].
- **Hybrid jet printing.** Electrohydrodynamic printing combined with piezoelectric excitation at the nozzle tip [Kim et al., 2009].
- **Spinning Disk.** Jetting fluid onto a spinning disk which will then transport the fluid towards the outer diameter where it is thrown out in the form of droplets [Bewley et al., 2013]. A recently developed version of the spinning disc droplet generator is the Vibro-Rotating Spray Generator [Chicheportiche et al., 2011]. It combines a continuous jetting with spinning disc to produce monodisperse aerosols. An artificial vibration, imposed on a jet, amplifies instabilities along the jet and breaks it at a fixed distance before being transported outward by the spinning disc. The difference to traditional spinning disc is that here the disc itself is additionally vibrating with the excitation frequency.
- **Supersonic liquid microjets.** Produced by an instantaneous expansion of a small fraction of the liquid located in a capillary. This is realized by a laser beam that heats up the liquid, which evaporates and expands, producing pressure jetting of the fluid [Steinke, 2015].

2.2.2 Design considerations

Ejection velocity and Satellite drops

Satellite drops, as shown in figure 2.1, are often an undesirable side effect when generating drops-on-demand. A liquid thread is formed at the nozzle orifice as the liquid gets ejected due to the pressure pulse generated by the actuator. If a piezoelectric element is used to generate the pressure pulse, then the dimensions of the liquid thread depend on the driving voltage and the properties of the liquid. If the liquid thread is long enough it can then break up into satellite drops. For a given piezoelectric element, Dong et al. [2006] observed two modes of breakup: end-pinching where the liquid thread pinches off from an almost spherical head, and multiple breakup due to capillary waves. Since the fluid is usually a given requirement for the droplet generator, it is mainly the driving voltage to the piezoelectric transducer, together with the response time of the mechanical structure, that determines whether satellite formation will occur and the size of satellites. Perhaps unexpectedly, the driving voltage has an insignificant effect on the primary drop size which is mainly determined by the orifice diameter. Droplet ejection velocity can be varied by changing the amplitude of the pulse. However, this effect was small for Yang et al. [1997], ranging from 0.5 to 0.7 m/s in their setup.

Dong et al. [2006] suggested two approaches to combat the formation of satellites: 1) using a single pulse with a sufficiently small driving voltage or 2) to use a combination of a larger and a smaller pulse. The smaller pulse is used to promote the separation of liquid from the nozzle orifice by producing a negative pressure, a suction, in the nozzle during the small pulse rise time. The

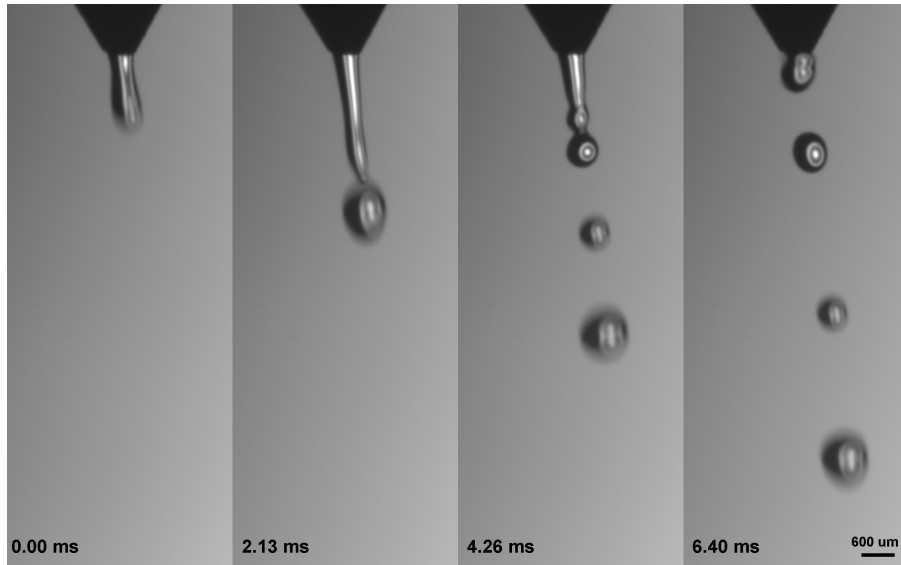


Figure 2.1: Satellite droplets formed from the liquid thread after ejection of the primary droplet.

suction of liquid back to the reservoir combined with the axial motion of the liquid mass away from the orifice causes a strong stretching of the fluid at the orifice. This leads to a necking which ultimately causes the liquid mass to break from the nozzle orifice sooner than would occur for the single-peak waveform.

Reservoir

The fluid pressure must be regulated so that a stable fluid meniscus forms at the nozzle outlet [Harris et al., 2015]. A reservoir that is placed too high causes continuous outflow, while one placed too low causes air to be drawn into the chamber. The droplet generators built by Cas [2008], Yang et al. [1997] and Harris et al. [2015] depend on a height difference between the fluid reservoir and the fluid chamber (where the nozzle is located). The height difference is then adjusted by using a precise translation stage. This concept assumes that the nozzle is orientated downward in the direction of gravity and might not be possible if the orientation of the nozzle is to be adjustable.

Additionally, Harris et al. [2015] used two concentric reservoirs together with a peristaltic pump to achieve a constant fluid level in the inner cylinder. The water coming from the pump would simply overflow into the outer cylinder. Using a reservoir without some sort of fluid level control like the peristaltic pump will cause the droplet diameter to decrease over time, directly depending on the decreasing reservoir height. However, the authors note that a pump is unnecessary for applications requiring only a small number of droplets. All of the mentioned reservoirs had in common that they were open to the surrounding atmosphere and no low pressure could be built up inside the reservoir that would counteract pushing out fluid at the nozzle. On the other hand, Terwagne et al. [2013] used an air pump for pressure regulation and control of the meniscus. Figure 2.2 shows the schematic of a reservoir as used by Harris et al. [2015].

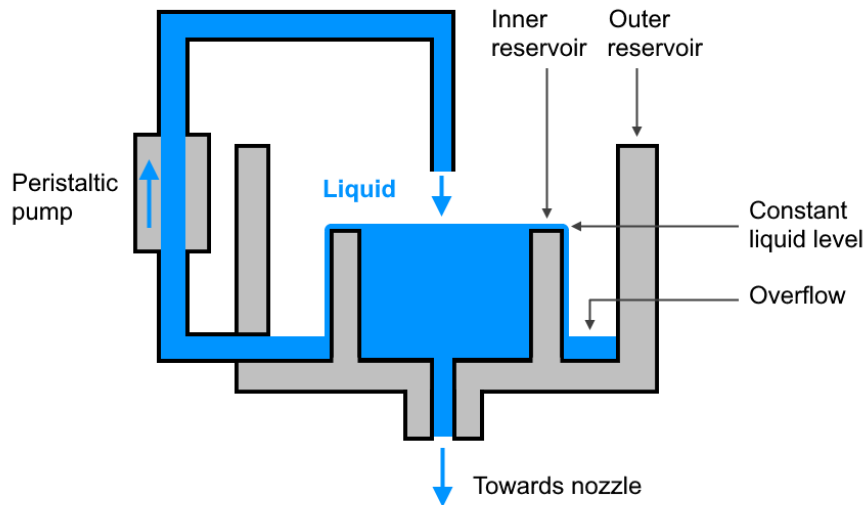


Figure 2.2: Liquid reservoir with constant liquid level where the liquid level is kept constant by means of overflowing an inner reservoir. The overflow is then collected in an outer reservoir and transported back towards the inner reservoir by means of a peristaltic pump. Schematic modeled after the reservoir seen in Harris et al. [2015].

Number of nozzles

Most DOD droplet generators are operated with a single nozzle, for example the generators used by Yang et al. [1997] and Harris et al. [2015], due to the difficulty of controlling the distribution of the pressure pulse in space between several nozzles precisely. If more droplets are required at a given time, it would be possible to use a group of tightly spaced droplet generators each with their own piezoelectric element and nozzle. Another approach is to create a nozzle that holds multiple orifices and use it with continuous jetting. Such a nozzle and nozzles in general are discussed in chapter 3.3.

Operation preparation

Air bubbles have to be removed from the tubes, the nozzle and the liquid volume in front of the actuator prior to usage of the generator. If this step is skipped, the force of the pressure pulse is no longer transporting the fluid towards the nozzle orifice but instead the gas inside the air bubbles is compressed, leading to no droplets being produced. If air bubbles are inside the nozzle, the air might be pushed out instead of the liquid. Harris et al. [2015] inverted their generator and held it at lower height than the fluid reservoir until fluid bled from the nozzle. They then sealed the nozzle with an elastic band and turned the generator back into the original position. They noted, however, that in normal operation, this bleeding procedure need only be performed once at the beginning of the experiment. Terwagne [2011] used a pressure pump together with a tank and a drainage pipe. The drainage pipe was used to evacuate any remaining air bubbles from the tank prior to operation. Then, using the air pump the pressure in the dispenser was finely adjusted so that the liquid surface into the hole was found just at the level of the exit hole.

It is further sensible to dry the nozzle with lint-free cloth before operation to discourage wetting of its outer surface. Another, albeit more difficult, option would be to design a hydrophobic nozzle, such as briefly shown in chapter 3.3.

Driving of the piezoelectric element

Harris et al. [2015] applied a constant negative voltage to the piezoelectric disc which caused it

to expand. They then used a positive square wave to contract the disc. Chen and Basaran [2002] and Dong et al. [2006] tested specific wave shapes.

Droplet charge

Changing the charge of the drops was not an option as it could alter the coalescence efficiency. Therefore, E-jet was not a viable option. Mechanical, aerodynamic or acoustic solutions would be suitable.

2.3 Development of a droplet generator

The design goals for the droplet generator were the following: Design, prototyping, calibration and testing of a new droplet generator, with drop size and drop velocity control. The drop generator could then be used to calibrate other equipment (e.g. particle-tracking systems, interferometers) and to speed up and refine other experiments. Using two such devices, experimental results on the drop collision and coalescence rate can be obtained, leading to a better understanding of their dependence on the relative velocity, size and angle of impact of the drops. The instrument had the following objectives:

1. Rapid creation of liquid droplets of diameter $5\ \mu\text{m}$ to $50\ \mu\text{m}$, similar to those typically found in warm atmospheric clouds.
2. Control of the initial velocity of the created drops, both in magnitude and direction.

The initial design of the droplet generator consisted of three major parts. The electronics assembly was planned to house a micro-controller unit (MCU) to control the experiment and the sensors, together with a corresponding power supply unit (PSU) and data storage. The particle imaging system (shadowgraphy) would monitor the relative velocity, size and angle of the drops. Finally, the droplet assembly would house the drop creation mechanism, consisting of a pump, a liquid reservoir, an actuator (a piezoelectric element), a nozzle with a micrometer sized orifice, and a system to control the angle with which the drop exits the experiment. Further ideas included a heater/cooler to control the temperature of the fluid. The droplet assembly would then be complemented by a set of environmental sensors (pressure, relative humidity, temperature). The block diagram of the instrument can be seen in figure 2.3.

The initial plan was then quickly adjusted by simplifying the setup through the use of commercially available hardware. A shadowgraphy setup was chosen for the imaging of the droplets, consisting of a Phantom VEO4k 990S high-speed cameras from Vision Research and a high-powered LED of type LMH2 from company Cree. The data in the form of video files would then be stored on a connected computer. Considering the electronics assembly, tests were performed with a Keysight 33512B signal generator, instead of a micro-controller, together with a Matsusada AMJ-2B20 $\pm 2\text{kV}$ high-voltage amplifier that was available from earlier experiments. The piezoelectric elements chosen were simple discs of type CEB-20D64 CUI Inc. with an outer diameter of $20\ \text{mm}$ ¹. Data storage was not necessary in the electronics assembly as the cameras would take care of this job. The setup as it is used can be seen schematically in Fig. 2.4.

The drop creation mechanism included the first design of a droplet generator in this work, shown in figure 2.5. Its main body contained a cylindrical shaft upon which a piezoelectric element was mounted with a lid and an o-ring as a seal. The cylindrical volume would later be filled with fluid and a pressure pulse transferred from the piezo towards the conical nozzle mount. The nozzle mount was exchangeable with three available options:

1. A mount to install a titan nozzle that was a leftover from older experiments and offered an orifice with a diameter of $600\ \mu\text{m}$ and therefore an easy test case compared to the later nozzles with much smaller orifices.

¹Data sheet in Appendix A

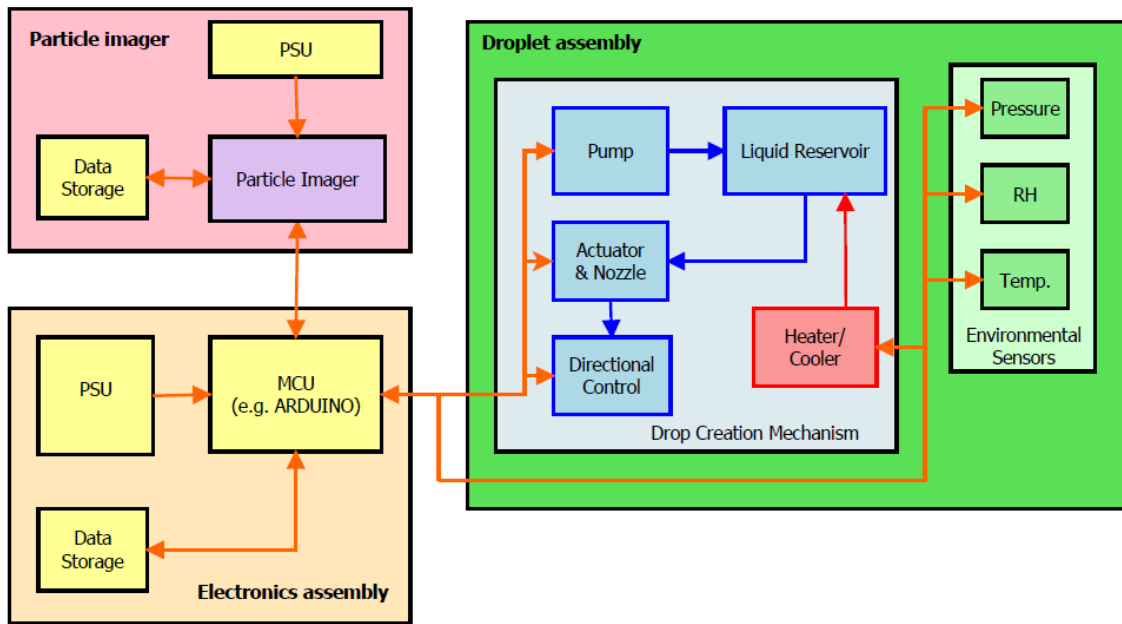


Figure 2.3: Schematic overview of the initial droplet generator design, composed of three major parts: 1) the particle imager would be used to record the generated droplets. It was to consist of the actual imager (a camera), a power supply unit (PSU) and a device to store the data. 2) The electronics assembly that would control the driving voltage for the piezoelectric element, consisting of a PSU, a micro-controller unit (MCU) and some data storage. 3) The droplet assembly, consisting of a set of sensors to measure the environmental variables, and the drop creation mechanism.

2. A mount that could hold a nozzle created using two-photon-polymerization (TPP) with a simple nozzle design consisting of a cylindrical orifice of various diameters of size $50\ \mu\text{m}$ and below. The printing parameters for the creation of these nozzles are as stated in chapter 3.2.2. Print times for these nozzles were typically around 20 hours per nozzle. The bulk of that time is required for printing the supporting structure used for mounting the nozzle and only a fraction of the time is used for the creation of the actual orifice and the funnel guiding the liquid to the orifice. In the case of an orifice blockage the nozzles can be taken out of the droplet generator and placed in a isopropylalcohol (IPA) bath inside an ultrasonic cleaner which increases their usable life time. TPP will be extensively covered in chapter 3.

3. A TPP nozzle with a flow-focusing design.

The material chosen for the body of the droplet generator was stainless steel to minimize corrosion. Corrosion can be an issue when liquids are being handles and can by minimized e.g. by avoiding electrical current in contact with the fluids, by choosing materials with low potential differences (galvanic corrosion), by choosing corrosion-resistant materials, and/or by controlling the pH-value of the fluid. Directional control was achieved by mounting the droplet generator onto a rotational stage from supplier Thorlabs. Both a simple liquid reservoir made from Plexiglas as well as a syringe pump of type PhD Ultra from Harvard Apparatus were tested.

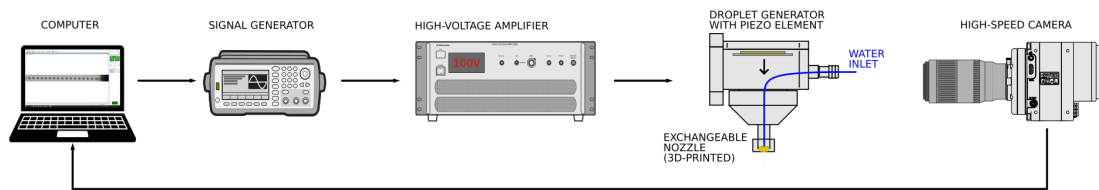


Figure 2.4: The droplet generator setup. When generating droplets with the setup the following steps are processed (from left to right): 1) a waveform is designed and passed to the signal generator. 2) A waveform is generated, ranging between ± 5 V and passed to an amplifier. 3) The signal is amplified by a factor of 200. 4) The amplified signal reaches the piezoelectric element and contracts/expands, generating a pressure wave in the process. 5) The generation of the droplets is recorded with a simple shadowgraphy setup consisting of a high-speed camera and a LED (not shown). 6) Data is stored on the computer in the form of videos.

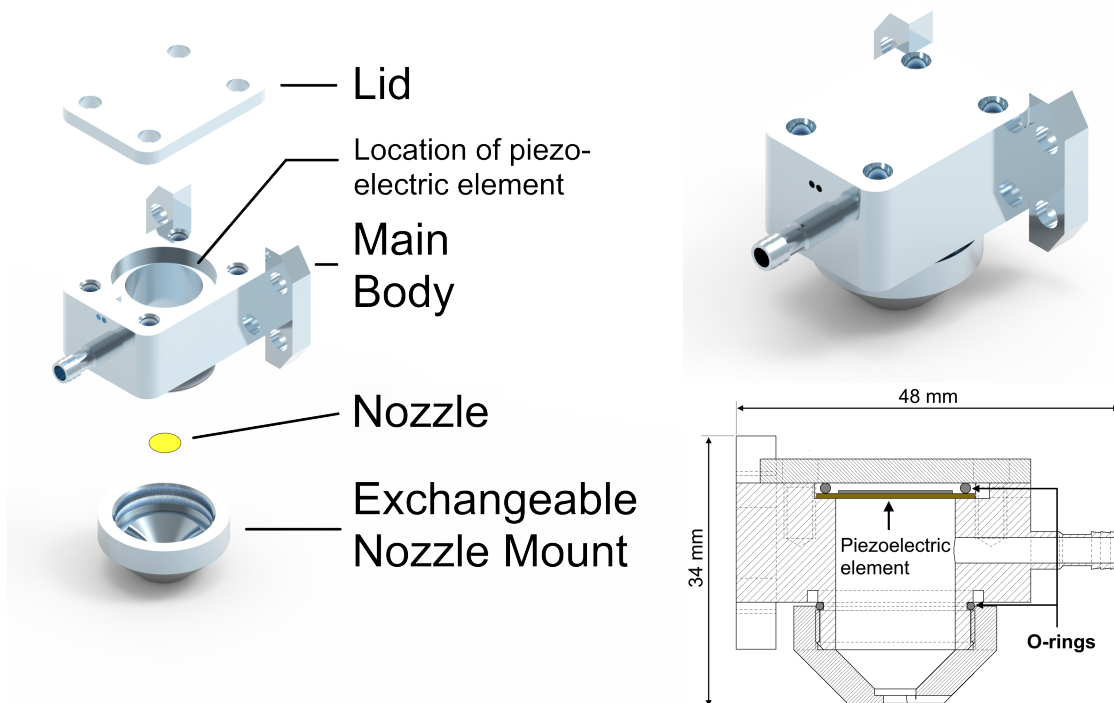


Figure 2.5: The first droplet generator design. **Left:** Exploded view of the droplet generator showing the different components. It features an exchangeable nozzle mount that could be used with either of three available options: Titan nozzle, Two-Photon-Polymerization (TPP) simple nozzle, TPP flow focusing nozzle. A piezoelectric element was mounted with the lid and an o-ring, attempting to provide a water-tight housing. Another o-ring was used to seal the connection between the main body and the nozzle mount. The main body featured two through-holes that lead the electrical connections outside the housing. The front of the droplet generator features a connector piece onto which a tube could be mounted that lead to the fluid reservoir. The fixation points can be used to mount the droplet generator to different x-/y-/z-stages if desired. **Right:** Closed view of the droplet generator and cross-section view of the droplet generator's interior. The cross-section view indicates the positions of the two o-rings and the piezoelectric element. The nozzle mount in shown in the cross-section was designed for TPP flow focusing nozzles.

The first design was moderately successful in generating drops-on-demand. An example of drops created with the titan nozzle was already shown in figure 2.1. It was generally possible to force out drops with a 100 μm TPP nozzle but they were heterogeneous in size and the majority of them were generated with many satellite droplets, even when varying the driving voltage amplitude and pulse width. An example sequence of polydisperse droplets generated in that manner is shown on the left in figure 2.6. Continuous jetting, on the other hand, was working well and successfully generated monodisperse or bidisperse droplets, as shown on the right side of figure 2.6. Generally, running either sine waves or repeating pulses worked well for the continuous jetting mode, with the frequency being set to $f = 4232$ Hz, and the driving voltage varied between 20 Vpp (peak-to-peak) to 80 Vpp.

As it turned out, the reason for the difficulties with generating droplets in the DOD mode were mainly due to the type of amplifier used and partly due to continuing issues with leakage, typically centered around the piezo element. While the amplifier was working well for the high voltage scenarios that it was built for, it had trouble providing a clear signal in the comparatively low voltage use case of 20 Vpp to 80 Vpp compared to its usual application range of up to 2kV Vpp. It was therefore difficult to feed the piezo with a clear driving voltage, which had a larger negative effect on the droplet generation in the DOD mode than it had in the CJ mode. Even when the input voltage from the signal generator into the amplifier was set to zero, the amplifier would still output a noisy signal to the piezo of $O(1)$ volts. This meant that the piezo, which was rated for well below 100 Vpp, was constantly in motion. Feeding a signal of 0.1 Vpp would then be amplified by a factor of 200x, resulting in an amplifier output $V = 20$ Vpp. In the case of the CJ mode, the amplified signal, even if noisy, was still forcing the Rayleigh-Plateau instability at the correct frequency to enable monodisperse droplet generation but for the DOD mode having such a noisy input signal was not good enough for the reliable generation of monodisperse droplets.

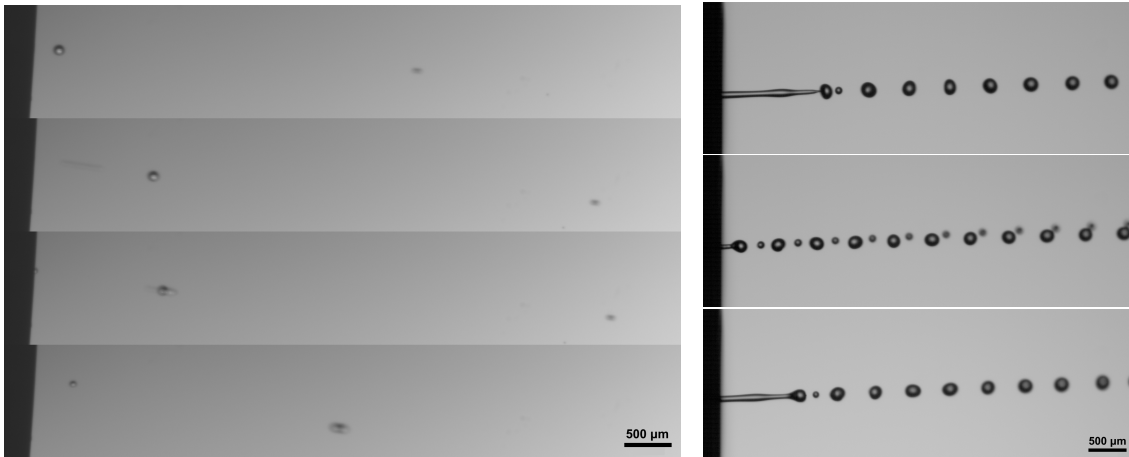


Figure 2.6: DOD and CJ droplet generation with version 1 of the droplet generator. Direction of gravity is pointing towards the right. **Left:** Drop-on-demand generation sequence with a 100 μm TPP nozzle. The generation is polydisperse and droplets are ejected with different velocities. These issues were caused by a noisy driving voltage and issues with sealing the droplet generator. **Right:** The same nozzle but used with a continuous stream and with different parameters, either providing the driving voltage as a sine wave or as a repeating pulse with specified pulse width and frequency. *Top:* Sine wave, $V = 20$ Vpp, $f = 4232$ Hz, monodisperse droplets, *Middle:* Sine wave, $V = 80$ Vpp, $f = 4232$ Hz, bidisperse droplets, *Bottom:* Pulse with pulse width of 20 μs , $V = 40$ Vpp, $f = 4232$ Hz, monodisperse droplets.

The second design of the droplet generator, shown in Fig. 2.7, was improved with the lessons learned from the first design in mind. It consists of a stainless steel main body, a stainless steel

screw cap with a concentric hole in the middle for mounting the nozzle and a stainless steel connector for the liquid supply. An inexpensive piezoelectric element CPT-1495C-300 from CUI Devices with a watertight housing² was used to generate a pressure wave, and a polyoxymethylene (POM) plastic top cover held the piezoelectric element in place. Additionally, two rubber o-rings and several screws make the generator leak-tight. A self-built amplifier³ and the same Keysight 33512B signal generator provided the necessary driving wave form to actuate the piezoelectric element. The altered setup is shown in figure 2.8.

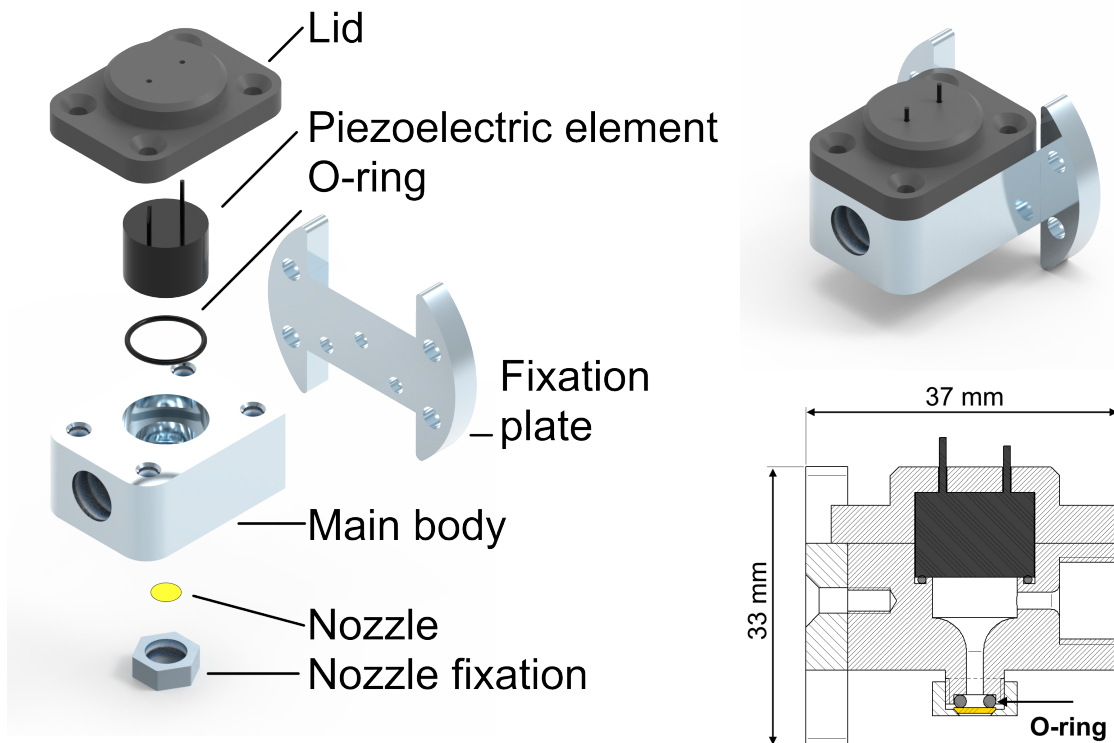


Figure 2.7: The final droplet generator design. **Left:** Exploded view of the droplet generator showing the different components. It features an exchangeable nozzle with corresponding fixation, as well as a mount for a piezoelectric element with water-tight housing. O-rings are used to seal the components and are held in place by four screws and the lid in case of the piezo or by the nozzle fixation. The lid features two through-holes that lead the electrical connections outside the housing. The front of the droplet generator features a threaded hole into which the connector for the fluid reservoir can be inserted. The fixation plate can be used to mount the droplet generator to different x-/y-/z-stages if desired. **Right:** Closed view of the droplet generator and cross-section view of the droplet generator's interior.

²Data sheet in Appendix A

³Built by the electrical workshop of the MPI DS (the employing institute).

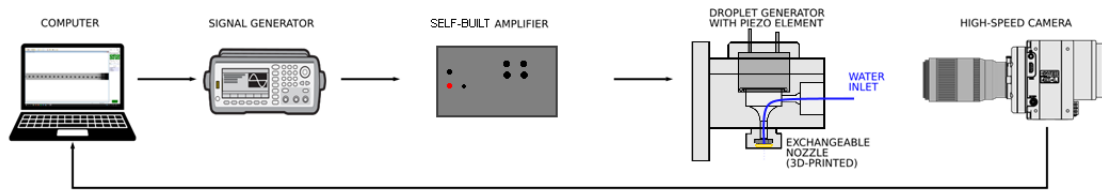


Figure 2.8: Updated droplet generator setup. Compared to the original setup the amplifier has been replaced by a self-built model. Further, the second iteration of the droplet generator is used which features a streamlined design with fewer points of failure compared to the original design.

In continuous jet mode, the droplet generator is driven with a repeating square wave at an amplitude of $U = \pm 30$ V and a frequency $f = O(10)$ kHz. A sine wave can be also used as an alternative. In DOD-mode, the droplet generator is driven by a $U = \pm 30$ V pulse with a pulse width of 200 μ s and selectable droplet generation frequency of up to 70 drops/second. Both wave forms are shown in Fig. 2.9.

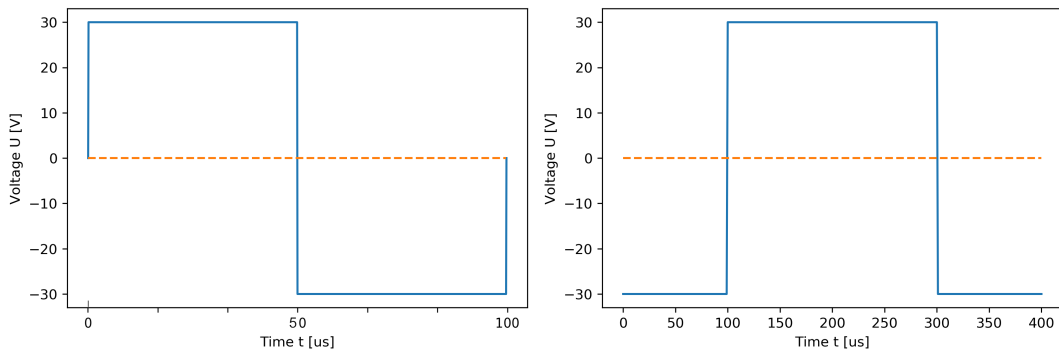


Figure 2.9: Wave forms for both operation modes of the droplet generator. Left: Continuous jet mode where a repeating square wave is applied with an example frequency of $f = 10$ kHz; Right: DOD mode which is operated with a single pulse, creating drops at a selectable droplet generation frequency of $f < 1 - 70$ drops/second.

When used in continuous jet mode, the droplet generator is able to create monodisperse droplets with diameters of approximately 115 – 125 μ m with an average distance of 180 μ m between individual drops when using a 50 μ m diameter orifice. As shown in Fig. 2.10, the drops have similar diameters and equal spacing between each other and continue to travel in this configuration until they leave the field of view.

In drop-on-demand mode, the generator achieves droplets with a diameter of down to 55 μ m while using a 40 μ m orifice. The spacing between the droplets is much larger in DOD mode compared to continuous stream mode, given by the slower generation frequency. This generation frequency could be reduced to less than 1 Hz if desired. Both modes of operation are shown in Fig. 2.10.

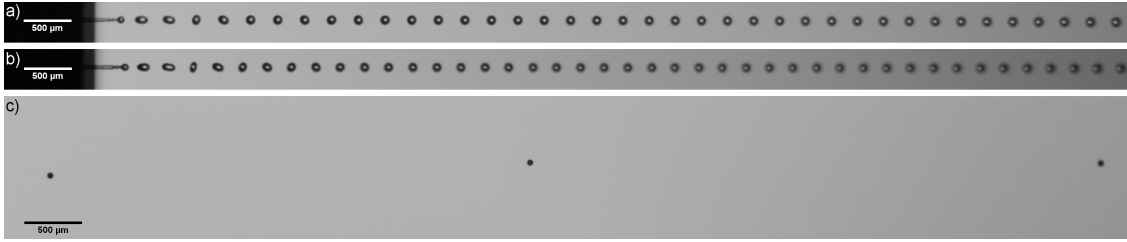


Figure 2.10: Both operation modes of the droplet generator using a nozzle with a $40\ \mu\text{m}$ orifice. **a)** CIJ mode driven by a square wave with $f = 11\ 210\ \text{Hz}$ and flow rate $Q = 0.25\ \text{ml/hr}$. The droplets have an average diameter of $D_d = 121.5\ \mu\text{m}$ with a standard deviation $\sigma = 2.4\ \mu\text{m}$. **b)** CIJ mode driven by a sine wave with $f = 12\ 470\ \text{Hz}$ and $Q = 0.25\ \text{ml/hr}$. The droplets have an average diameter of $D_d = 118.0\ \mu\text{m}$ with a standard deviation $\sigma = 3.8\ \mu\text{m}$. **c)** DOD mode, recognizable from the large inter-droplet distances compared to the continuous jet mode. The droplet diameter in DOD mode was approximately $55\ \mu\text{m}$. The direction of gravity in all images is from left to right.

2.4 Conclusion

For DOD mode, a simple square pulse as a driving wave form was used. However, recent work from Castrejón-Pita et al. [2012] shows that more complex wave forms can manipulate the meniscus on a large orifice in such a way that it retracts before detaching and a smaller drop is detached, effectively decreasing the size of the droplet without the need for smaller orifices. On the other hand, it was found that TPP is easily capable of producing much smaller orifices than the $D_o = 40\ \mu\text{m}$ and $D_o = 50\ \mu\text{m}$ nozzles with our droplet generator. In a test print, shown in Fig. 2.11, orifices of down to $D_o \approx 3\ \mu\text{m}$ were created successfully.

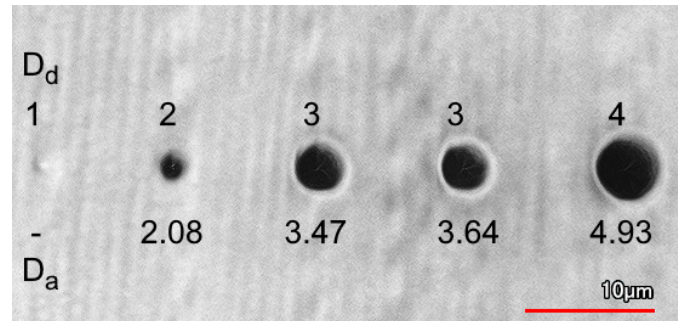


Figure 2.11: Orifice test print. The resulting diameters (in μm) are indicated as desired diameter D_d and true (achieved) diameter D_a . Orifices were blocked when designing them with $D_d = 1\ \mu\text{m}$, open at $D_d \geq 2\ \mu\text{m}$, and achieved good circularity from $D_d \geq 3\ \mu\text{m}$. The prints were created with the $25\times$ objective. Using the $63\times$ objective instead of the $25\times$ is likely to allow for even smaller orifices (the available objectives are covered in chapter 3).

However, using such small orifices comes with a heightened importance of pre-filtering the liquid to prevent any eventual orifice blockage. It would also necessitate a different mechanical design for the droplet generator. The liquid volume directly between the nozzle and the piezo element is relatively large at about $1\ \text{cm}^3 = 1 \cdot 10^{-6}\ \text{m}^3$ compared to the small droplet volume of $\leq 6.5 \cdot 10^{-14}\ \text{m}^3$ for droplets with $D \leq 50\ \mu\text{m}$. The piezo actuator could not reliably force water through orifices smaller than $D_o = 40\ \mu\text{m}$, presumably due to the relatively large liquid volume in between the nozzle and the piezo. Using CIJ mode, the problem that the pressure needed to drive the necessary pipe flow through the orifice follows $\Delta p \propto 1/D^4$, as stated in Hagen–Poiseuille’s equation, is encountered. Going below $D_o = 50\ \mu\text{m}$ created leakages within the droplet generator at the o-ring that connected the piezoelectric element to the DG housing, so a more sturdy design would be

necessary in this case.

Ultimately, the droplet generator was able to fulfill its main function of creating droplets with either drop-on-demand or continuous stream modes. In doing so it achieved droplets diameters of down to 118 μm in CJ mode and down to approximately 55 μm droplets in drop-on-demand mode. Operating the generator at these limits demanded fine tuning of the parameters driving the piezoelectric element (voltage, frequency for CJ, pulse width for DOD) and care in avoiding both particles entering the interior of the generator, as well as avoiding the formation of air bubbles during the filling of the generator with fluid.

Chapter 3

Two-Photon-Polymerization (TPP) for the production of nozzles and atmospheric particles

3.1 Introduction

Solid particles found in Earth’s atmosphere include ice crystals [Pruppacher and Klett, 1980], dust particles [Lamb, 1970, Prospero et al., 1970], pollen [Gregory, 1978], and any other material that is light enough to be swept up by surface winds [Gieré and Querol, 2010]. These particles are mostly non-spherical in shape and come in various combinations of aspect ratios and levels of regularity. Given their ubiquity, finding non-spherical particles in nature is easy: for example, already a small amount of sand shows large differences in grain sizes and shapes [Daphalapurkar et al., 2011]. Similarly, cirrus ice clouds contain a multitude of ice crystal shapes such as compact, irregular or columnar particles, as well as plates and rosettes [Wolf et al., 2018]. However, performing repeated experiments with objects of the exact same shape and material properties can prove difficult: Many of these particles are smaller than 1 mm in diameter and seldom occur in the same shape more than once. Typically, experimentalists have to either find naturally occurring shapes or work with easily manufactured goods, such as glass and metal capillaries/cylinders, prisms and discs [Roy et al., 2019b, Madhav and Chhabra, 1995, Chhabra et al., 1999]. An alternative approach is to change parts of the experiment and use larger-sized particles like cm-sized 3D-prints and perform experiments in a different surrounding fluid.

Using Two-Photon Polymerization(TPP), it is not only possible to manufacture micrometric non-spherical and irregular particles well below 1 mm in diameter and to reproduce them as often as desired, but also to create precisely defined parts with complex designs such as miniature nozzles as will be shown in this chapter. TPP is the process of selectively curing parts of a viscous, photosensitive fluid called a photoresist with a laser at a fitting wavelength. In the chapter, the potential of a Nanoscribe GT TPP printer for the production of nozzles and artificial particles is explored. It is structured as follows: In section 3.2.1, the fundamentals of TPP are introduced. As the default production parameters specified by the manufacturer do not always create optimal results, a parameter search with a specific geometry was performed and is covered in section 3.2.2.

Next, several nozzles are created in section 3.3 and used with the self-built droplet generator from chapter 2. This allows for the creation of water droplets of down to $D = 50 \mu\text{m}$ in diameter in two modes of operation, 1) continuous jet (CJ) and 2) drop-on-demand (DOD). The droplet generator has a screw-in mechanism to quickly exchange printed nozzles and enable fast prototyping of different nozzle designs. Different nozzle geometries are discussed.

In section 3.4, ellipsoidal particles with equivalent diameter $D_{eq} = 140\mu\text{m}$ are created to explore the terminal velocity of non-spherical particles with a high particle-to-fluid density ratio of $\rho_p/\rho_f \approx 1000$. This density ratio is appropriate to study atmospheric particles such as ice crystals, sand storm dust or pollen flight, and constitutes a step closer to reality from the usual experiments with $\rho_p/\rho_f \approx 1$ where particles are immersed in a fluid such as water.

Finally, an outlook is given on possible future uses of TPP in an atmospheric science context in section 3.5 and the reconstruction of complex particles on the example of an ice crystals is demonstrated.

3.2 2-Photon-Polymerization and Nanoscribe GT

3.2.1 Fundamentals

Used in this work was a Nanoscribe GT from Nanoscribe GmbH & Co. KG, a 3D-printer that allows creation of features with sizes of down to 200 nanometer with a $63\times$ objective or down to 1 micrometer with a $25\times$ objective. Typically, the workflow includes the preparation of an indium tin oxide (ITO)-covered glass slide which is first cleaned with acetone and isopropyl alcohol (IPA). After the glass slide has been secured on a sample holder, a ca. 5 mm diameter drop of photoresist, a light-sensitive viscous fluid, is placed onto the glass slide. The sample holder is turned upside down and inserted into the printer. Then, the objective is immersed into the photoresist. The printing process writes a structure into the photoresist line by line while slowly traveling away from the glass slide to build up layers of lines. A microscope stage is used for large scale movements (a large scale in this case is about 200-300 micrometers) and a piezoelectric element controls the fine movements. A schematic overview of the process is shown in Fig. 3.1 a).

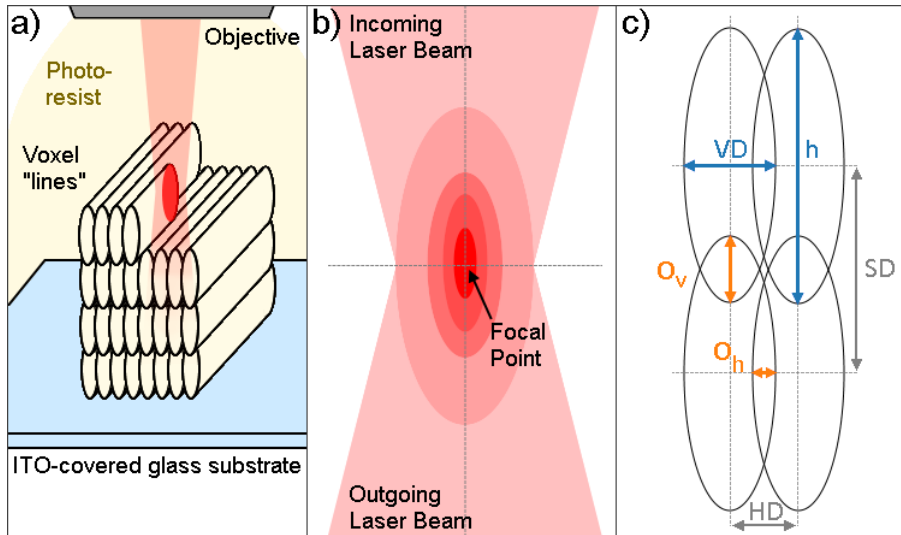


Figure 3.1: **a)** Schematic overview of the printing process: The structure is made up from individual voxel lines that are written into the photoresist by focused application of a laser. Note that the different parts are not to scale and in reality the voxel lines would be much smaller compared to the glass substrate and objective, with a typical diameter of $0.4\mu\text{m}$. **b)** The laser creates an ellipsoidal spot of polymerized photoresist, the *voxel*. The voxel diameter is directly proportional to the laser power LP up to a limit at which bubbles are created within the resist. **c)** Overlap between neighbouring voxels. Here, VD is the voxel diameter, h the voxel height, o_v the vertical voxel overlap, o_h the horizontal voxel overlap, SD the slicing distance and HD the hatching distance.

The Two-Photon-Polymerization itself is carried out by a laser with a center wavelength of $\lambda = 780$ nm. The simultaneous absorption of two photons hitting a spot in the photoresist locally raises the energy level of the photoresist at that spot to a virtual state followed by an excited state which leads to polymerization. The underlying principle is explained in detail by Zhou et al. [2015]. The laser of the Nanoscribe GT has a mean power of 50 mW and a pulse duration of about 100 femtoseconds with a repetition rate of 80 MHz. The main parameter of interest when using TPP is often the *laser power* (LP) which is a percentage of the maximum intensity provided by the laser. Controlling LP is done through an acousto-optic modulator, a small optical element that, depending on a voltage set in the range of 0V to 5V, passes a percentage of the full laser intensity. As such, LP determines the intensity within the focal spot and can range from zero to one hundred percent. As indicated in Fig. 3.1 b), the resulting polymerized spot, also called a *voxel*, is ellipsoidal in shape and its size will be directly proportional to the laser power. Leaving the photoresist exposed to the laser for a longer duration will eventually lead to locally creating bubbles inside the photoresist which are visible as dark spherical spots. At the same time, a laser power that is too low will not reach the polymerization threshold. The speed with which the laser beam is moved over the photoresist is called the *scan speed*. In some cases, laser power and scan speed are combined to a single parameter, the *dose*. The dose applied to a spot of photoresist is inversely proportional to the square root of the scan speed. The scan speed was not varied in this work, so we will only refer to the laser power (LP) from now on.

Once the printing process has finished, the printed object (henceforth referred to as the print) is taken out of the printer and placed into a beaker filled with Propylene Glycol Methyl Ether Acetate (PGMEA), a chemical that removes the non-polymerized photoresist and leaves only the print. After a minimum of 30 minutes in the PGMEA, the print is placed into a beaker filled with IPA for another 2 minutes and finally blow-dried with a nitrogen gun. This work has to be performed inside of a clean room due to the sensitivity of the printing process to pollution through dust particles and the small scales involved. Optically, the printed objects are transparent with a light yellow tint and a smooth surface. The smoothness of the surface and the inherent optical properties of the photoresist, IP-S from company *Nanoscribe*, are of high enough quality to be used in miniature optics as demonstrated by Thiele et al. [2017]. According to Liu et al. [2017], the density of the finished print is 1.2 g/cm^3 . The polymerized photoresist material is hard enough to permit careful handling with plastic tweezers without creating dents on the surface.

3.2.2 Printing parameters

When using an apparatus for TPP, such as the Nanoscribe GT, several printing parameters need to be carefully chosen: 1) the laser power (*LP*), which determines the diameter of the resulting voxel, 2) the hatching (*HD*) and slicing (*SD*) distances, which determine the horizontal and vertical distances respectively between individual voxel line centers, 3) the scan speed with which the lines are created and 4) the stitching parameters for objects that exceed the printable area of the chosen objective (a volume of up to $400 \times 400 \times 400 \mu\text{m}^3$). Stitching is the process of splitting a large print into several smaller blocks, each fitting within the maximum printable area, and later reattaching the blocks by precisely overlapping their outer walls. Two parameters control the stitching: overlap angle and amount of voxel lines that overlap. Together, these parameters have a strong influence on the resulting quality of the print and affect the surface roughness, overall structural stability and how closely the print follows the design.

When using their IP-S photoresist, together with the corresponding 25x objective, the standard parameters provided by Nanoscribe GmbH ($LP = 100\%$, $HD = 0.5 \mu\text{m}$, $SD = 1.0 \mu\text{m}$, scan speed = $100\,000 \mu\text{m/s}$) weren't ideal for the majority of the applications and resulted in poor print quality. Finding the correct parameters for the apparatus at hand was done once initially and then every several months as properties of the apparatus can deteriorate over time. These properties include the maximum output of the laser, the objective surface and the precision of the piezo-element. Another contributing factor is the used photoresist which, in the case of the

photoresists provided by company Nanoscribe, has a typical expiration time of 6 months after purchase and the optimum print parameters can vary between batches. Optimum parameters, in this case, describe the best achievable combination of surface roughness, structural stability and print accuracy (how closely the print follows the model).

One approach for selecting the right parameters is to perform a parameter sweep. For this, a simple square block of $100 \times 100 \times 30 \mu\text{m}^3$ was designed in the CAD-program Solid Edge (though any CAD-program would be suitable) and then processed with the program *DeScribe* that comes with the *Nanoscribe GT*. The advanced STL processing dialogue allows for the specification of a parameter range, such as varying the laser power, HD , or SD . Since the actual resulting voxel diameter (VD) and aspect ratio VAR , with

$$VAR = \frac{\text{voxel height}}{\text{voxel diameter}} = \frac{h}{VD} \quad (3.1)$$

are unknown, a set of slicing and hatching parameters was chosen with a specific overlap between individual voxel lines in mind, based on the approach described in the following.

As seen in Fig. 3.1 c), the overlaps between voxel lines horizontally (o_h) and vertically (o_v) are directly affected by HD and SD . The overlaps o_h and o_v can be set between 0 (no overlap) and 1 (fully overlapped). A desired overlap of $o_h = o_v = 0.25$ was chosen. The voxel diameter VD was assumed to be between $VD = 0.3 \mu\text{m}$ and $VD = 1.0 \mu\text{m}$, depending on the laser power LP . The voxel aspect ratio VAR , on the other hand, was assumed to be found somewhere between $VAR = 1.0$ (sphere) to $VAR = 6.0$ (elongated ellipsoid). This informed the chosen values for HD and SD . For example, if the actual $VD = 0.6 \mu\text{m}$ and $VAR = 3.0$, then a 25% overlap between voxels both horizontally and vertically would be achieved by setting $HD = 0.45 \mu\text{m}$ and $SD = 1.35 \mu\text{m}$. The calculation goes as follows: If $VD = 0.6 \mu\text{m}$, then an overlap o_h of 25% VD leads to

$$HD = (1 - o_h) \cdot VD = 0.75 \cdot 0.6 \mu\text{m} = 0.45 \mu\text{m}. \quad (3.2)$$

For SD , the voxel height h is calculated first:

$$h = VD \cdot VAR = 0.6 \mu\text{m} \cdot 3.0 = 1.8 \mu\text{m} \quad (3.3)$$

and then

$$SD = (1 - o_v) \cdot h = 0.75 \cdot 1.8 \mu\text{m} = 1.35 \mu\text{m}. \quad (3.4)$$

Nine sets of blocks were printed with the laser power constant for each set and increasing between sets in increments of 5%, starting at $LP = 60\%$ and ending at $LP = 100\%$. Each set consisted of 31 individual blocks with 30 blocks being used for testing the parameters and a single extra block for determining the orientation of the print. Every block was artificially designed to have two stitching lines intersecting in the center, even though the small block size would allow printing the block in one go. Instead, stitching the block was done to find the optimal parameters for HD and SD while using the default stitching parameters: a 15° angle of overlap and an overlap thickness of 2 voxel lines. The knowledge gained could then be applied to larger prints. The resulting parameters can be seen in table 3.1.

To evaluate the parameters, the prints were examined under a Keyence VK-x260 3D laser scanning confocal microscope that can achieve resolutions of down to 0.5 nm. Fig. 3.2 shows the resulting prints. A dark area is visible in most sets of blocks with both high VAR and high VD , indicating misprinted material. The amount of blocks misprinted due to high VAR and high VD decreases with increasing laser power, going from 27 fully blackened and 1 partially blackened block at $LP = 60\%$ to 1 fully blackened and 7 partially blackened blocks at $LP = 100\%$. Starting at $LP = 85\%$, a similar dark area develops for combinations of small VAR and small VD , leaving undisturbed blocks on a diagonal line of moderate VAR and VD .

VD, HD	VAR					
	1.0	2.0	3.0	4.0	5.0	6.0
0.3, 0.23	0.23	0.45	0.68	0.90	1.13	1.35
0.4, 0.30	0.30	0.60	0.90	1.20	1.50	1.80
0.6, 0.45	0.45	0.90	1.35	1.80	2.25	2.70
0.8, 0.60	0.60	1.20	1.80	2.40	3.00	3.60
1.0, 0.75	0.75	1.50	2.25	3.00	3.75	4.50

Table 3.1: Slicing distances (SD) in μm that were chosen for the parameter sweep. VAR is the voxel aspect ratio and VD is the voxel diameter in μm that was assumed for the given parameters. HD is the hatching distance, also given in μm .

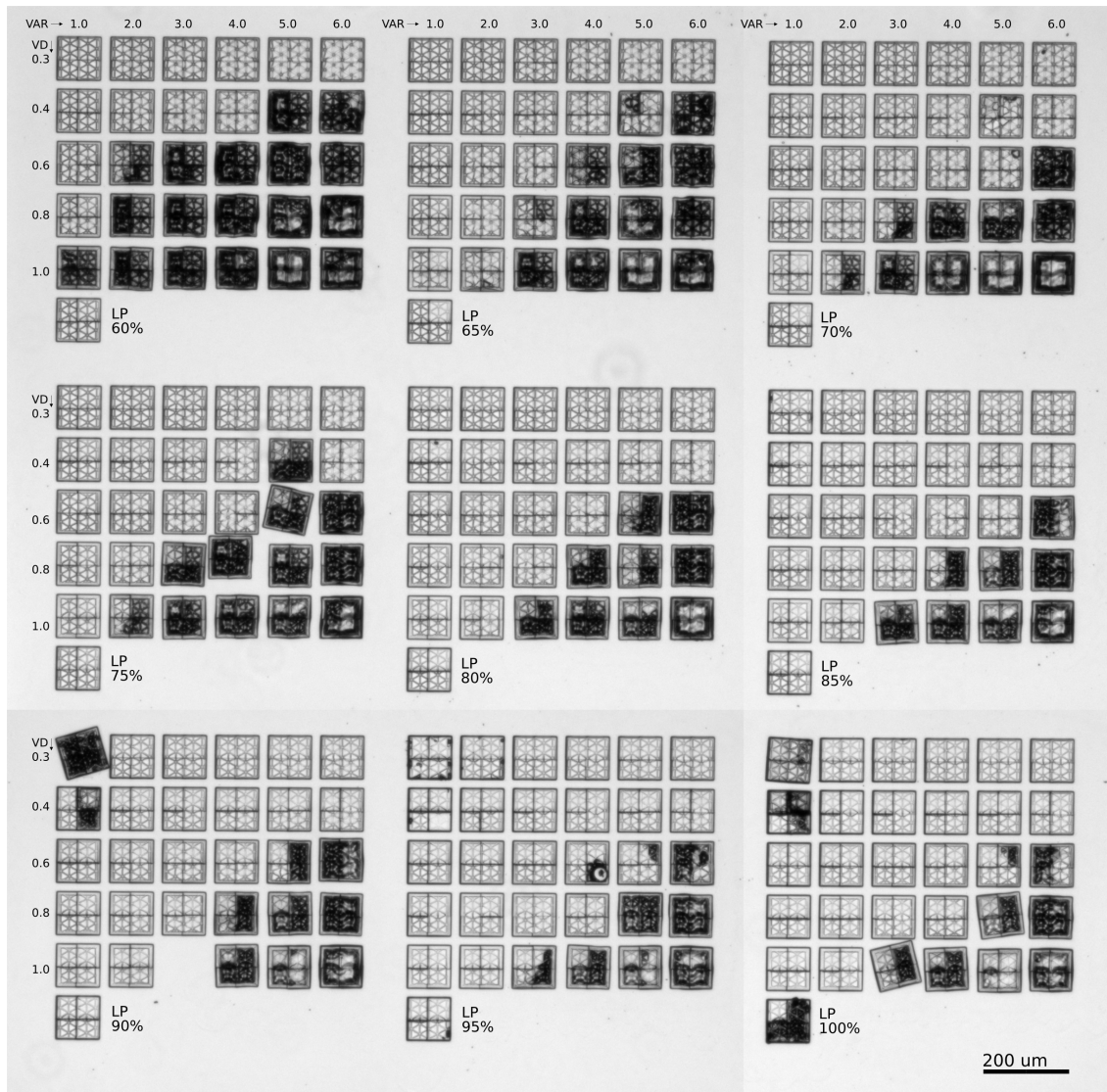


Figure 3.2: Result of the parameter sweep for the IP-S photoresist together with the 25x objective on the Nanoscribe GT. A dark area in the lower right is visible in most sets of blocks which recedes with increasing laser power, indicating misprinted material due to high VAR and VD. Starting at $LP = 85\%$, a similar dark area develops in the upper left due to small VAR and VD, leaving undisturbed blocks on a diagonal line going from bottom left to top right where VAR and VD have moderate values.

	Sa	Sq	Sa _s	Sq _s
Max.	2.37	2.62	3.24	4.02
Min.	0.13	0.16	0.22	0.25
Avg.	0.95	1.08	1.37	1.88
Std. Dev.	0.73	0.79	0.93	1.18

Table 3.2: Maximum, minimum, average and standard deviation surface roughness values for $LP = 70\%$ in μm . Measured with a Keyence VK-X260 3D laser scanning confocal microscope. Sa is the arithmetical mean height and Sq the root mean square height. Seven prints of visibly poor quality have been manually filtered out. This results in better surface roughness and is closer to practical applications. Sa_s and Sq_s include the stitching lines in the surface roughness measurement. Note that the average Sq_s is about $0.5\mu\text{m}$ more than Sq. This increase is consistent with the typical height of a bump created by the stitching lines.

There are two mechanisms giving rise to the misprinted areas:

First mechanism is overexposure of the photoresist when slicing and hatching distances are small and laser power is high, corresponding to small values of VAR and VD ($LP = 85\%$ and upwards). This leads to the formation of bubbles inside the photoresist, creating voids and structural instability inside the print.

Second mechanism results from material that is too loosely connected due to slicing and hatching distances too large so that the voxel overlap increasingly becomes too small. This can be seen in areas where VAR and VD are large. These misprints disappear with increasing laser power as the voxel grows in size with the applied laser power, thus providing enough overlap to form a strong connection between the voxel lines. Considering both the stitching and the overall print quality, the optimal parameters for general applications were found to be $LP = 70\%$, slicing distance $SD = 0.6\mu\text{m}$, hatching distance $HD = 0.3\mu\text{m}$ while using a scan speed of $100\,000\mu\text{m/s}$. The slicing distance can also be increased to $SD = 0.8\mu\text{m}$ for large prints which saves a bit of printing time but comes at the cost a reduced print quality.

The surface roughness of the prints with $LP = 70\%$ was found to be $Sa = 0.95\mu\text{m}$ with a standard deviation $\sigma = 0.73\mu\text{m}$ (see also Tab. 3.2). Stitching lines typically create a bump on the surface between neighbouring parts that is about $0.5\mu\text{m}$ in height. It is important to note that in terms of surface roughness, the optimal parameters can vary depending on the desired application of the print. As the print is created layer by layer, smaller slicing and hatching distances can reduce the height between layer steps and, thus, create less aliased surfaces. However, reducing the slicing and hatching distances by too large a factor will introduce other issues. For one, the laser power needs to be tuned down to not risk creating bubbles inside the print. On the other hand, very small distances can introduce deformations on the bottom of prints. For prints that have a flat surface, this generally is not an issue but it can be for prints with overhanging structures, such as the axisymmetric particles in section 3.4. For those particles, the values $LP = 70\%$, a reduced slicing distance $SD = 0.6\mu\text{m}$, hatching distance $HD = 0.3\mu\text{m}$ while using a scan speed of $100\,000\mu\text{m/s}$ provide good results.

3.3 Droplet generator nozzles

In order to get used to the machine and to learn the capabilities of the Nanoscribe, a complex nozzle design was printed that was taken from Nelson et al. [2016]. This design is used for a flow focusing droplet generator and often called a "gas dynamic virtual nozzle" (GDVN). In a GDVN a gas flow is used to thin an outgoing stream before it is broken up by Rayleigh-Plateau instability. The design was chosen as a test print as it showed promise in creating very small drops and further

provided a complex geometry that was only achievable by use of a precise 3D-printer such as the Nanoscribe due to it having difficult to manufacture internal features. Specifically, the geometry consists of two nozzles, one for transporting a liquid and one for focusing the gas flow. At the same time, the outer diameter of the nozzle is only 1000 μm wide. The design and the test print can be seen in figure 3.3. Printing the design was carried out with a $63\times$ objective and the IP-Dip photoresist from supplier Nanoscribe GmbH. The print was useful in shedding light on a few issues. For one, the printing times for a structure with dimensions of about 1.0 mm x 1.0 mm x 1.0 mm resulted in fairly long printing times that exceeded 24 hours. Further, the printing itself came out in bad quality, showing disconnects between stitching lines as apparent in the zoomed-in portion of figure 3.3. Following from this, it was decided to use a $25\times$ objective with less magnification to speed up printing times. It also turned out that the printer itself hadn't been well calibrated by the selling company and thus needed to be re-calibrated. After this recalibration, the printer performed as expected.

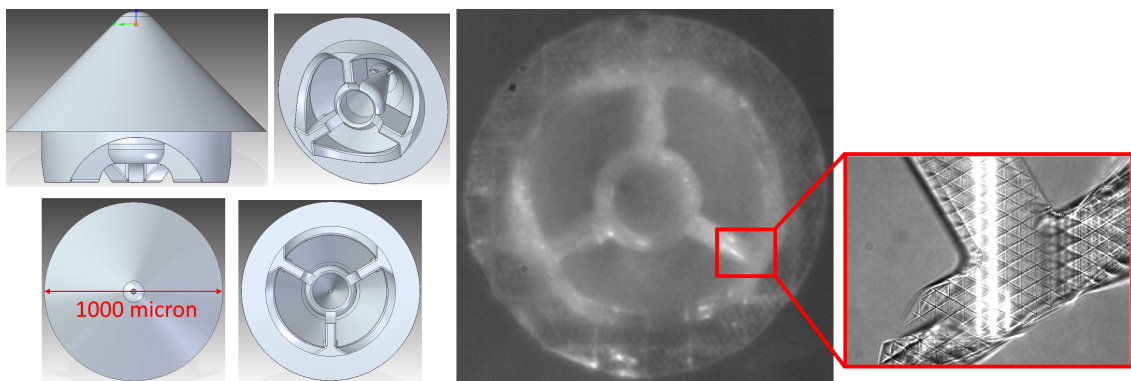


Figure 3.3: Nozzle test print following the design from Nelson et al. [2016].

Next, the first unique design was created, inspired by the GDVN but with an extra inset that moved the fluid nozzle back. As one of the design goals was to freely control the initial velocity of the droplets, the thought process behind the inset was to first generate a drop-on-demand and then accelerate the droplet by moving it through a flow-focusing section. While a prototype design was created, it was deemed impractical as experiments with the first GDVN quickly revealed the difficulty in operating micro- to millimeter sized nozzles. For example, the area surrounding the orifice of the nozzle would frequently wet and then was in need of drying, either by use of a piece of clean cloth or some pressured air. In general, the huge freedom in designing the nozzles with TPP can easily lead the designer astray in creating overly complex nozzles that do not stand the test of practical implementation. A few such nozzles were created in this work. The drop-on-demand GDVN and two prototype nozzles are shown in figure 3.4.

Further nozzles tests include a "multi-nozzle" that was created with 20 individual orifices, visible in Fig. 3.5 a) and b). In yet another test print, the photoresist was applied directly onto a piezoelectric element and printed a structure onto the element. This approach worked reasonably well and a bond between the ceramic of the piezoelectric element and the printed material was established. However, this approach requires extra caution, as the surface of the immersion objective can easily be scratched if it comes into contact with the piezoelectric element. Further, the laser power should be applied as a profile with very small intensity at the surface which then ramps up to the normal printing laser power over the first hundred lines. This is necessary as the laser normally exits the transparent glass substrate on the opposite site while the opaque piezoelectric element reflects the laser back into the photoresist. The print onto a piezoelectric element can be seen in Fig. 3.5 c) and d).

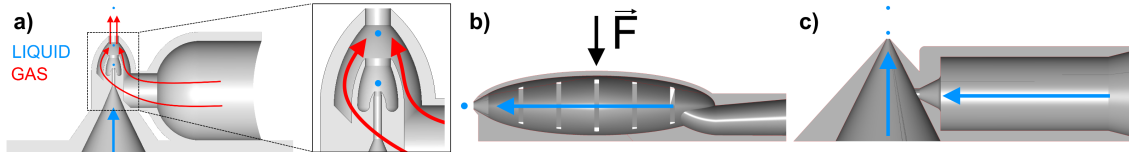


Figure 3.4: Nozzle prototypes. **a)** GDVN/DOD-crossover design: an inset is used to separate the DOD and the flow-focusing nozzles with the aim to generate drops first and then accelerate them with the gas flow later. **b)** Squeezed design: the droplet is squeezed through the frontal orifice by applying force onto the top of the design. The flexible elements indicated in the volume and the inherent flexibility of the material are used to move the top of the design back into position after the force is no longer applied. **c)** Conical design: a simple design that would be bounded to an piezoelectric element with fluid being supplied by the connector from the right side.

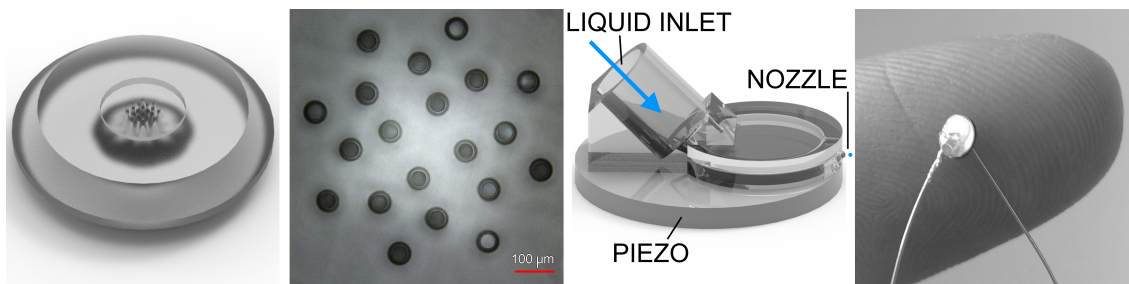


Figure 3.5: Nozzle and droplet generator test prints. **a)** Render of a nozzle with 20 individual orifices. **b)** Microscope image of the printed nozzle from a), focused on the orifices. **c)** Render of a nozzle/droplet generator design to be printed directly onto a piezoelectric element. **d)** Test print of the design shown in c).

The final designs for the nozzles ended up being a lot more simplistic than the nozzles covered until now. Aiming to create an easy to use nozzle with few chances for failure or leakage, a simple cylindrical orifice was combined with a surrounding circular disc and conical water inlet. The design would work with both drop-on-demand and the easier to realize continuous stream mode. The circular disc had a diameter of 5 mm and a thickness that was gradually brought down from 1500 μm to 700 μm over a couple of iterations. The conical section would lead the fluid through the main body and had rounded-off edges both at the entry point and at the transition to the orifice for a smooth transport of the fluid. The orifice itself would then protrude out of the body by 100 μm and was created with several different diameters and shapes. An initial size for the orifice diameter was chosen to be 100 μm and was then subsequently reduced to 50 μm , 40 μm , 25 μm and 10 μm diameters. Most successful of the nozzles turned out to be one with a smooth 40 μm diameter cylindrical orifice. This created the smallest achieved droplets. Diameters of 25 μm and 10 μm turned out to be too small to enable the piezoelectric element to squeeze water through them. The inclusion of ragged edges, on the other hand, did not lead to a notable improvement during the creation of drops but posed a potential source for misprints which could then block the orifice or create an uneven detachment point for the water drops. Conical designs were also created instead of the cylindrical one to provide a more sturdy design as the cylindrical orifices had thin walls of 5 μm thickness and would not survive any mechanical stress. This meant that handling the cylindrical orifice nozzles needed to be done with higher care than the conical ones. Figure 3.6 shows some of the mentioned designs.

A selection of created nozzles can be seen in Fig. 3.7. Tested nozzle types include varying diameters and the inclusion of ridges on the orifice for manipulation of the meniscus. Figure 3.7a) shows the top view of a printed nozzle with an outer diameter of 5 mm using the design of figure 3.6a).

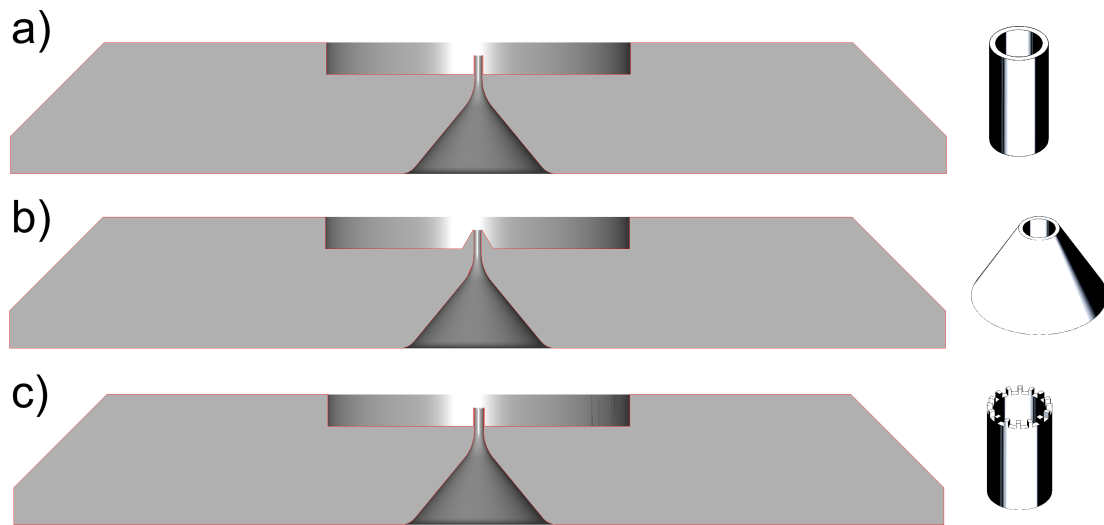


Figure 3.6: Final nozzle designs. **a)** Cylindrical orifice with smooth edges and 40 μm diameter. **b)** Conical orifice with smooth edges and 40 μm diameter. **c)** Cylindrical orifice with ragged edges and 50 μm diameter.

The nozzle orifice is visible in the center of the nozzle. The checkerboard pattern is caused by the stitching procedure that allows the printer to realize designs that exceed the maximum printing area by overlapping blocks of printed material. Figures 3.7b)-e) are zoomed in on the nozzle orifice with b) and d) showing the 3D design of orifices with smooth and ragged ridges while c) and e) show microscopic images of the finished print. Figure 3.7f) is a high-resolution 3D scan of the orifice shown in d) and e), showing how closely the finished print follows the ragged ridge design.

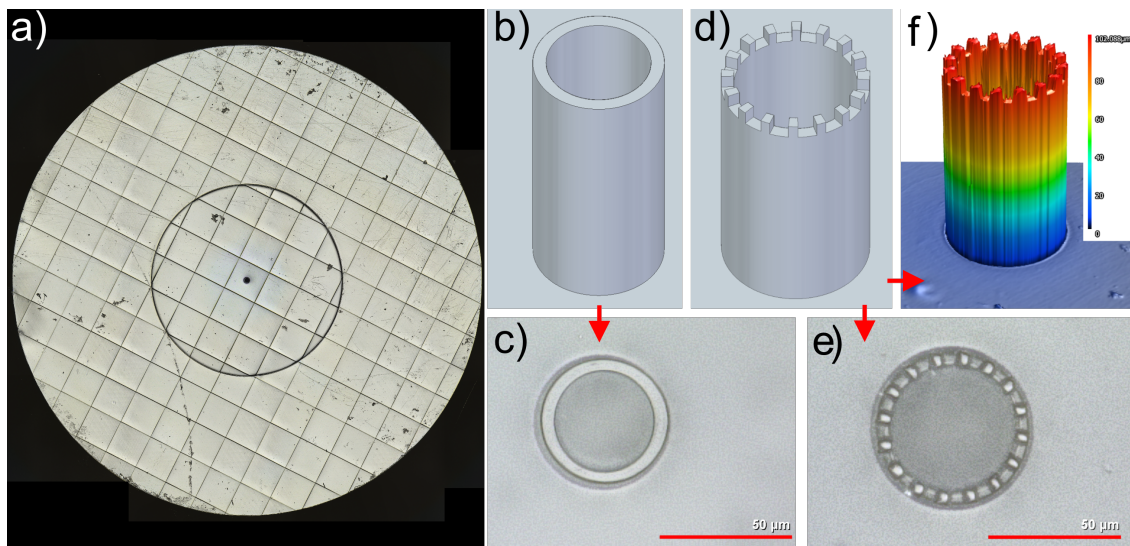


Figure 3.7: Nozzles used with the self-built droplet generator. **a)** Top view of a nozzle with a smooth orifice design. **b)-c)** Orifice design with smooth ridges as a 3D model and the resulting print as seen under a microscope. **d)-f)** Orifice design with ragged ridges as a 3D model, the resulting print as seen under a microscope and as a 3D scan.

3.4 Ellipsoidal particles

The drag force that particles experience in free-fall depends on their shape, the particle-to-air density ratio and the turbulence that develops in the wake of the particles [Bagheri and Bonadonna, 2016]. Different regimes are used to describe drag the particle experiences. These are the Stokes' regime, the intermediate regime and Newton's regime. The regimes are defined by their the particle Reynolds number that describes the flow structure of the wake behind the particle (e.g. laminar or turbulent):

$$Re_p = \frac{\rho_f d_{eq} |\mathbf{v}_p - \mathbf{v}_f|}{\mu_f} \quad (3.5)$$

where ρ_f is the surrounding fluid density, d_{eq} is equivalent particle diameter, \mathbf{v}_p is the particle velocity, \mathbf{v}_f is the extrapolated fluid velocity at the particle centroid and μ_f is fluid dynamic viscosity. Stokes' regime is then found at $Re_p < 0.1$, the intermediate regime between $0.1 \leq Re_p < 1000$ and Newton's regime at $1000 \leq Re_p < 3 \cdot 10^5$. Analytical solutions for the drag force exist only for spheres in Stokes' regime and certain exceptions, such as slender bodies [Roy et al., 2019b]. Therefore, empirical models are used to describe the drag on non-spherical and irregular particles, necessitating experiments and simulations [Bagheri and Bonadonna, 2016].

3.4.1 Shape descriptors: Elongation (EL) and flatness (FL)

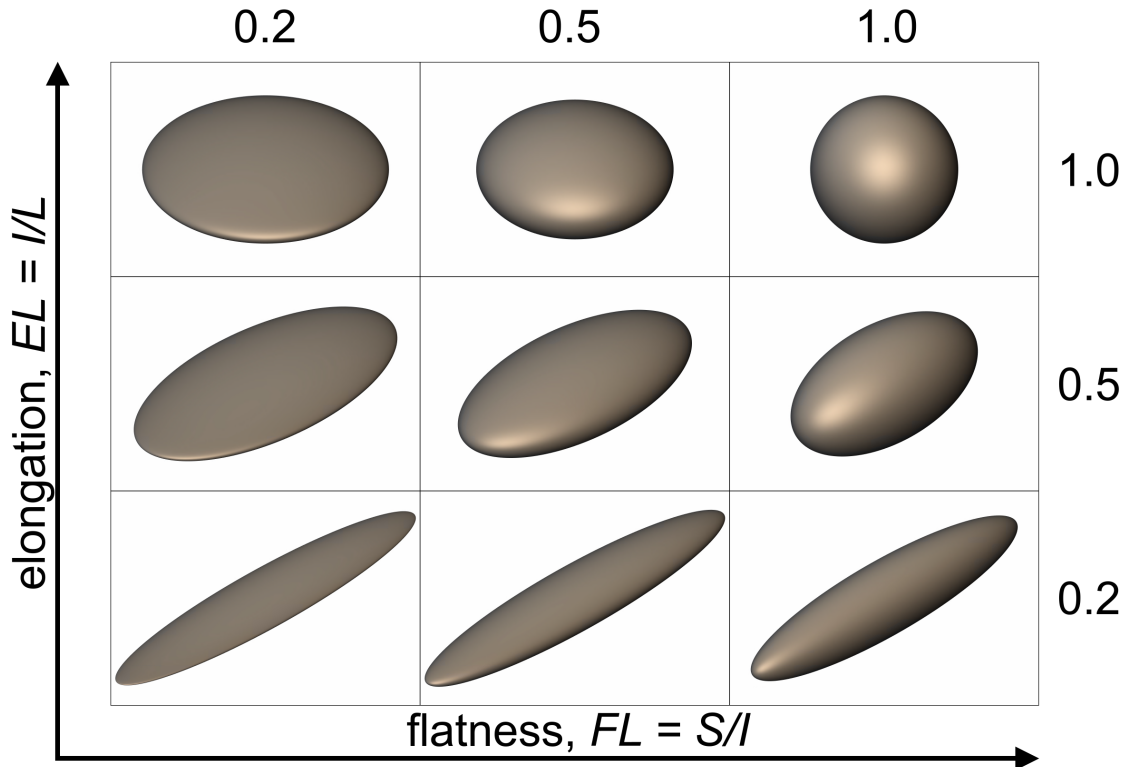


Figure 3.8: The nine shapes of ellipsoids created using TPP, ranging from flat discs (EL 1.0, FL 0.2) to axisymmetric prolates (e.g. EL 0.2, FL 1.0), spheres (EL 1.0, FL 1.0) and proper ellipsoids with three distinct axes (e.g. EL 0.2 FL 0.2).

To study particle terminal velocities, nine ellipsoidal shapes were created with different values of elongation and flatness, based on Bagheri and Bonadonna [2016]. An overview of different shape descriptors is given in the same paper. Elongation and flatness are ratios between the largest

(L), intermediate (I) and smallest (S) dimensions of the particle. In this work, these dimensions correspond to the principal axes of the ellipsoids. Elongation is then the ratio $EL = I/L$ and flatness the ratio $FL = S/I$. Both ratios can range from zero to one with $FL = 1.0$ describing axisymmetric spheroids, $EL = 1.0$ discs, the combination $EL = 1.0$ & $FL = 1.0$ spheres, and other combinations of EL and FL describing ellipsoids with three distinct dimensions. An overview of the created shapes can be seen in figure 3.8. For practical reasons, the smallest values used for elongation and flatness were $EL = 0.2$ & $FL = 0.2$ as smaller EL would result in very long particles that break easily when removing them from the glass substrate. Smaller FL would lead to difficulties when removing these very flat particles from the glass substrate after the print due to their increased adhesiveness. At the same time, the particles cannot be printed along the sharp edge as there is a high chance that they would detach during the printing process or during the development. All printed ellipsoids have the same volume of $1.44 \cdot 10^{-12} \text{ m}^3$ which corresponds to a sphere with equivalent diameter of $D_{eq} = 140 \mu\text{m}$. With a particle Reynolds number of $Re_p = 1 - 10$, the created particles are designed to be in the intermediate regime.

3.4.2 Particle printing results

The particles were produced with the following parameters: IP-S photoresist, 25x objective, $LP = 70 \%$, $SD = 0.6 \mu\text{m}$, $HD = 0.3 \mu\text{m}$ and the standard scan speed of $100\,000 \mu\text{m/s}$. Creating 56 particles takes about 2 hours of printing time. The prints show visible steps between layers, typically $0.64 \mu\text{m}$ in height ($\sigma = 0.04 \mu\text{m}$). Initial prints with smaller $SD = 0.1 \mu\text{m}$, $HD = 0.2 \mu\text{m}$ exhibited much smoother surfaces with smaller steps but resulted in deformations on the bottom of the particle. These deformations cannot be picked up by the 3D-scans and only became visible in the actual experiments when the particles were visible from all sides.

According to the manufacturer Nanoscribe¹, the shrinkage depends on the degree of polymerization and is influenced by the combination of laser power and scan speed. Decreasing the shrinkage is achieved with a higher dose, though care must be taken to not overexpose the photoresist so that micro-bubbles are formed. Another cause for the shrinkage can occur during the development of the particles: Soluble components inside the polymerized structure such as initiators that did not react and unpolymerized monomers, dissolve inside the PGMEA developer [Denning et al., 2011]. Minimizing the development time would therefore help to decrease the shrinkage. The largest contributor to shrinkage, however, was determined to be a slight swelling of the structure caused by the developer with subsequent shrinkage due to capillary effects during the drying process [Meisel et al., 2006, Sun, 2012]. In a test designed with small structural features that display high shrinkage, Purto et al. [2018] managed to reduce the shrinkage from drying from 30% to 10% by UV flood curing the print while it was still within the developer. Using the printing parameters as indicated in the above paragraph ($LP = 70 \%$, $SD = 0.6 \mu\text{m}$, $HD = 0.3 \mu\text{m}$ and the standard scan speed of $100\,000 \mu\text{m/s}$), however, equally lead to a reduction in shrinkage and no visible defects in the printed ellipsoids. A selection of the printed particles can be seen in Fig. 3.9. All particles were printed with the 25x objective and the corresponding IP-S photoresist from supplier Nanoscribe.

¹<https://support.nanoscribe.com/hc/en-gb/articles/360002381814-Structural-Deformation>, accessed 12.04.2021

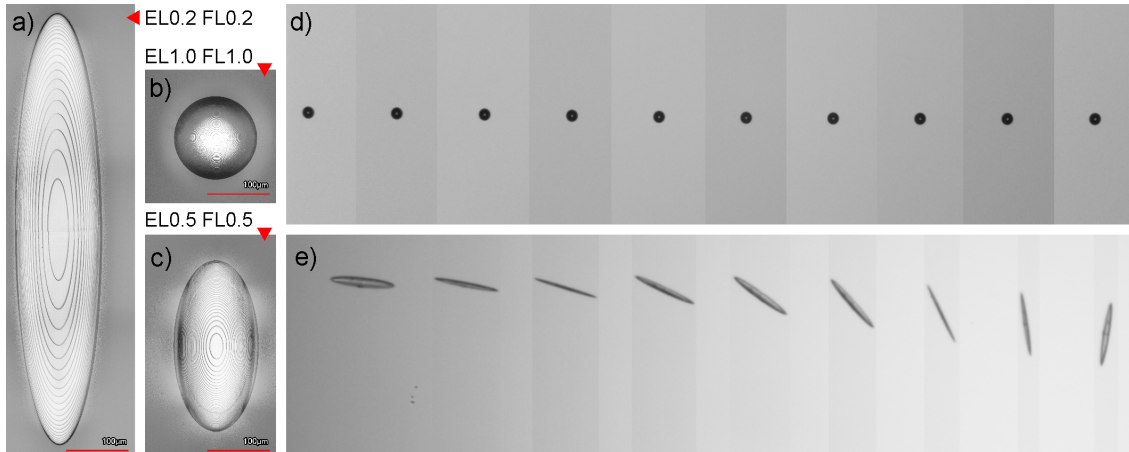


Figure 3.9: Ellipsoids created using TPP (25x objective, IP-S photoresist). **a)** Ellipsoid with elongation $EL = 0.2$ and flatness $FL = 0.2$. **b)** Ellipsoid with elongation $EL = 0.5$ and flatness $FL = 0.5$. **c)** Sphere with elongation $EL = 1.0$ and flatness $FL = 1.0$. **d)** Composite image of sphere from **c)** in free-fall in air. **e)** Composite image of ellipsoid from **a)** in free-fall in air. Both **d)** and **e)**: The slight difference in background lighting is due to heterogeneous illumination by the LED. The direction of gravity is from left to right.

3.5 Conclusion and outlook

Using two-photon polymerization (TPP), nozzles were produced that could be used in a self-made droplet generator to create monodisperse droplets in both continuous jet and drop-on-demand mode. In terms of nozzle design, TPP can create many different geometries. Different nozzles were successfully created and used with different diameters and smooth and ragged orifice edges. In general, TPP enables large freedom in design for nozzles.

It was also possible to create ellipsoidal particles with a realistic density ratio to air which, due to their small dimensions and accurate shape, can be used to explore the settling dynamics in the intermediate regime of particle Reynolds numbers $Re_p = 1 - 10$ with non-spherical particles. The particles behaved as expected in terms of terminal velocity as verified with a spherical shape in a settling column. The corresponding experiments and data will be shown in chapter 5.6.2. Another exciting possibility for the research of atmospheric particles using TPP is the reconstruction of naturally occurring shapes, such as ice crystals. As a demonstrator, a hexagonal ice crystal that was observed in a stratospheric cloud was re-created. The source image was from a holographic recording of the instrument HALOHolo [Schlenczek, 2018]. This was merely meant as a demonstrator, so some freedom was applied when designing the particle from the source image. For example, the 3D model and the final print exhibit small ridge lines along the edge of the particle. These were included to test if small details could be successfully recreated with TPP and do not necessarily represent structures found on real ice crystals. Nonetheless, it clearly shows that particles found in icy and mixed phase clouds can be accurately and repeatably reconstructed using TPP. The ice particle and its print can be seen in Fig. 3.10.

In conclusion, TPP was found to be a versatile tool for the creation of micrometric particles and nozzles with micrometric features. It allows for freedom in design and is well suited for prototyping fine structures up to several mm^3 in volume with high resolution and repeatability. Limitations of TPP include: extensive support structure needed as the machine is ideally installed inside a clean room to avoid particles from entering the printing area and to provide a temperature and moisture controlled environment. Several pre-/post-processing steps are necessary such as cleaning the glass slides, developing the print and curing larger prints under UV light. The technology also does

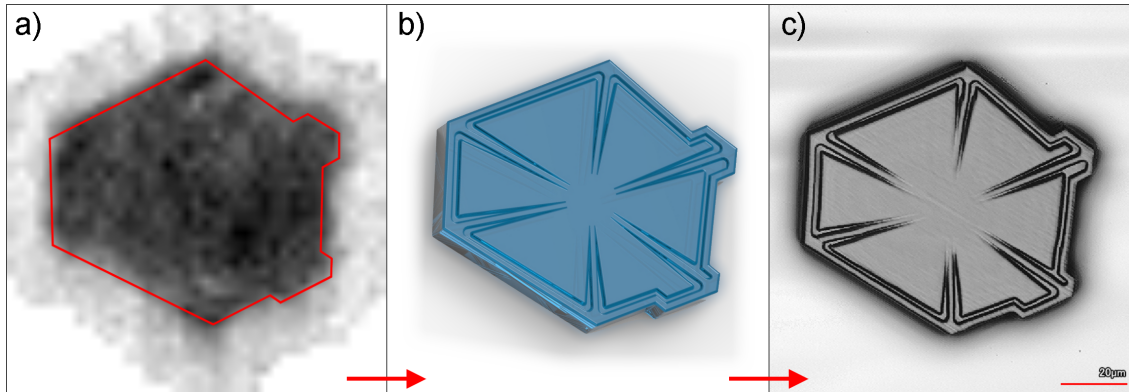


Figure 3.10: Hexagonal ice crystal created as a demonstrator. **a)** Image of an ice crystal recorded by HALOHolo and approximation of the ice crystal's outline. Image courtesy: Oliver Schlenczek, private communication **b)** 3D reconstruction including ridge lines to test fine printing parameters. The shape is an approximation and does not fully represent an actual ice crystal as the source image had a resolution too low for proper reconstruction. **c)** TPP print of the ice crystal.

not upscale to large amounts (10 000+) of particles that would be interesting for some turbulence experiments due to long printing times. For experiments that require smaller numbers or specific geometries, however, TPP brings great opportunities. It enables nozzle geometries that would have been difficult to achieve otherwise and provides a method of reliably creating particles with small dimensions and complex geometries. While it was focused on ellipsoidal shapes, tests with demonstrator ice particles showed the potential of using TPP in a plethora of ways in the context of atmospheric research.

Chapter 4

Design of a settling chamber

4.1 Introduction

Non-spherical particles are a common occurrence within Earth's atmosphere: Sandstorm dust, the flight of pollen in spring or the distribution of soot particles all directly affect people's lives and well-being. Ice crystals, found in icy and mixed-phased clouds, alter the atmosphere's radiation budget and are, hence, an important factor when studying climate change. Many models assume a spherical shape as a simplification as it massively reduces computational costs and enables analytical solutions. However, orientation does have a profound effect on the behavior of non-spherical sedimenting particles. As stated by Roy et al. [2019a], the particle orientation strongly influences the rate of sedimentation and the presence or absence of horizontal drift. In consequence, these parameters directly impact how long a particle of given shape can remain suspended in the air.

Much of the undiscovered realm lies in the dynamics of inertial non-spheroid ellipsoids as inertial particles in turbulence lag behind the flow to adjust their orientation [Voth and Soldati, 2016]. In particular, it is not quantitatively understood how particles of different shapes orient themselves in turbulent flows and how fast their orientation responds to flow fluctuations. Experiments on single particles in a quiescent medium are the first step to characterize this.

In order to measure the free flight characteristics of particles, such as the ones introduced in chapter 3, a settling chamber was designed. The requirements for the chamber included the following: the chamber was to allow enough space around the particle to allow for the wake of the particle to fully develop. It also needed to accommodate enough vertical flight time for the particle to reach its terminal velocity. Another requirement was that the insertion mechanism of the particle would not introduce airflow into the observation chamber, apart from the airflow caused by the particle itself. The insertion was to be repeatable, and offer a choice of several different initial velocities. In terms of calibration, the insertion mechanism should at the very least not get in the way of inserting calibration masks, or, ideally, ease the calibration by providing a clear point of reference.

On the measurement side, the settling chamber had to offer a clear view of the particle during its flight and provide a camera system with sufficient magnification to observe the particle while maintaining focus, with a field of view large enough to observe the particle over a period long enough to allow for several periods of oscillation or tumbling. Designing the chamber to be placed on x-, y-, and z-stages would allow for quick adjustments and help with moving the entire setup. This potentially helps during the calibration step or when another region of interest needs to be defined and the setup must be relocated as a means to monitor the region. An overview of the final structure, which consists of a drop tower holding the settling chamber and an insertion mechanism, is shown in Figure 4.1.

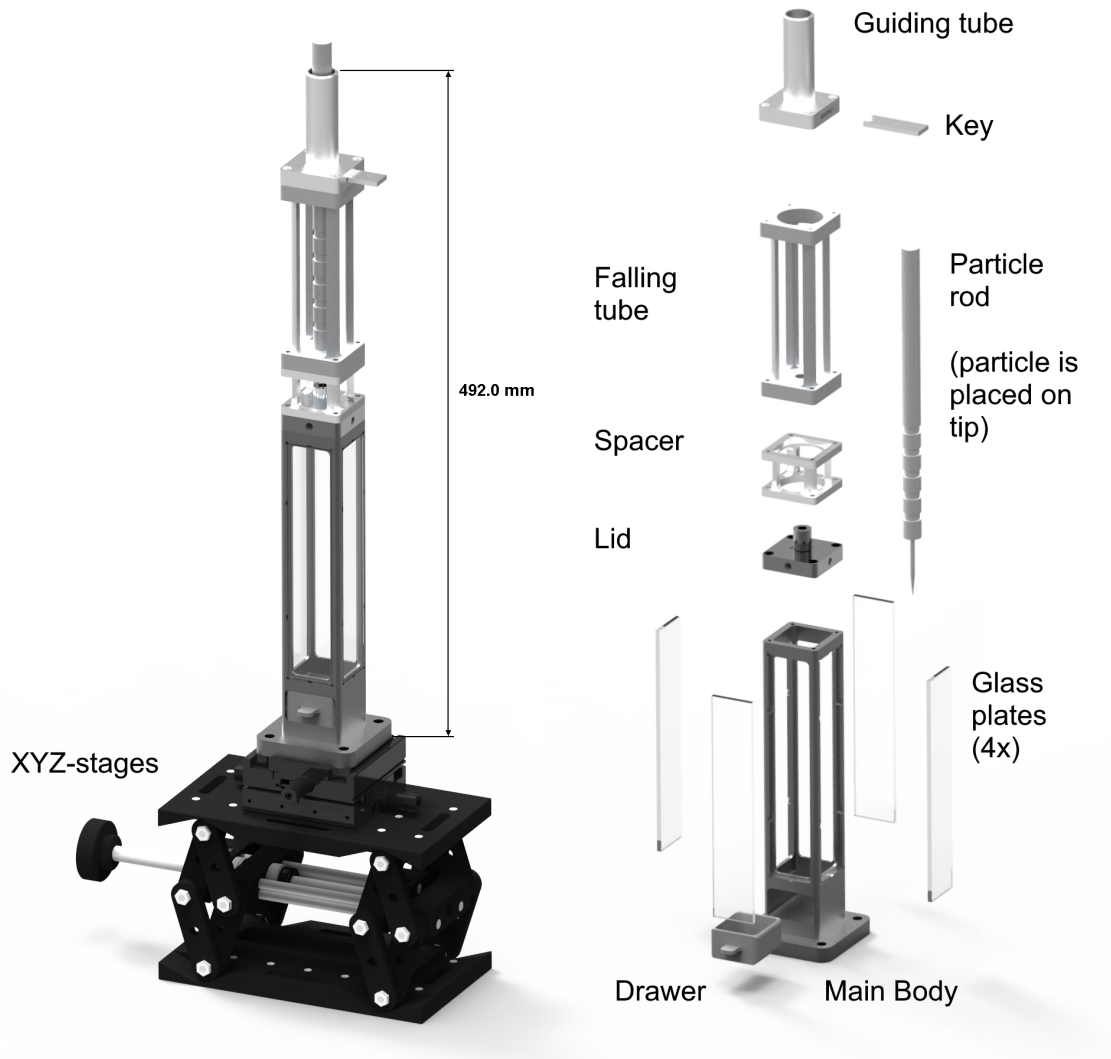


Figure 4.1: Drop tower and insertion mechanism

For observing the particles, a measurement scheme had to be chosen capable of providing both high spatial and temporal resolution. A shadowgraphy setup was adopted for this purpose, leading to the choice of cameras and the selection of a suitable light source, as covered in chapter 4.4.

The following sections specify the approach that was chosen. All parts of the drop tower and the insertion mechanism were designed by the author and built by the Max Planck Institute for Dynamics and Self-Organization mechanical workshop.

4.2 Drop tower

Design criteria: Clear view of particle flight; restrict influence of outside airflow

The drop tower consists of the main body, one drawer and the lid. All parts of the drop tower were made from aluminium and later anodized. To accurately determine the settling velocity, steady-state orientation and transient dynamics, the observation volume had to be closed off from the air flow within the laboratory. Another consideration was the size ratio between the smallest observation volume width and the largest dimension of the ellipsoids, as observation volumes that are too small can affect the wake of the particle and thus the dynamics of the free-fall [Awbi and Tan, 1981, Chhabra, 1995, Arsenijević et al., 2010]. The distance between the walls was chosen to be 30 mm, translating to a minimum aspect ratio of 42.8 given the maximum dimension of 700 μm of the longest particle shape (EL0.2 FL0.2), and a typical aspect ratio of 214 considering the equivalent diameter of 140 μm of all particle shapes.

The drop tower prominently features four cutouts that allow the placement of glass plates and an observable area of $26 \times 150 \text{ mm}^2$ when seen from the normal of the glass plate. The cutouts were then sealed with glass plates of optical quality. For this, four 3 mm thick windows were cut from a large glass slab by the glass workshop of the Max Planck Institute for Biophysical Chemistry. The glass was procured from Edmund optics and came with a broadband anti-reflective (BBAR) coating, passing light in the wavelength range from 425 nm to 700 nm with an average transmission rate of 95.6%, and reflecting other wavelengths. The glass windows were then assembled onto the main body via silicon adhesive.

Inside the main body is a $30 \times 30 \times 170 \text{ mm}^3$ shaft that leads into a drawer which is used to collect particles after operation. Typically, the particles ended up being used one-time only as even a small amount of dirt sticking to the particles could easily change their flight behaviour.

On the top of the main body are four M3 screw holes that can be used to connect other modules to the body. Another four threaded through-holes on the bottom of the main body are utilized to link the main body to an x-Stage. The wall thickness was opted to be 4 mm, creating a very stiff structure, able to withstand vibrations caused by the insertion mechanism.

The drop tower was mounted on three translation stages, two of which were x-stages combined to allow lateral travel, and one stage for z-travel to adjust the height of the drop tower. This facilitated the positioning of the experiment and made it easy to align it with the cameras. The tower was designed to be modular, and additional parts, such as spacers and the slide-in mechanism, could be easily retrofitted.

Finally, the lid consisted of two components: an actual lid, closing off the drop tower main body, and a guiding tube that is held in position by a screw and can be exchanged if needed. The guiding tube marks the point of entry for the falling particle after being released by the insertion mechanism. Figure 4.2 shows a schematic overview of the main body, the drawer and the two parts of the lid.

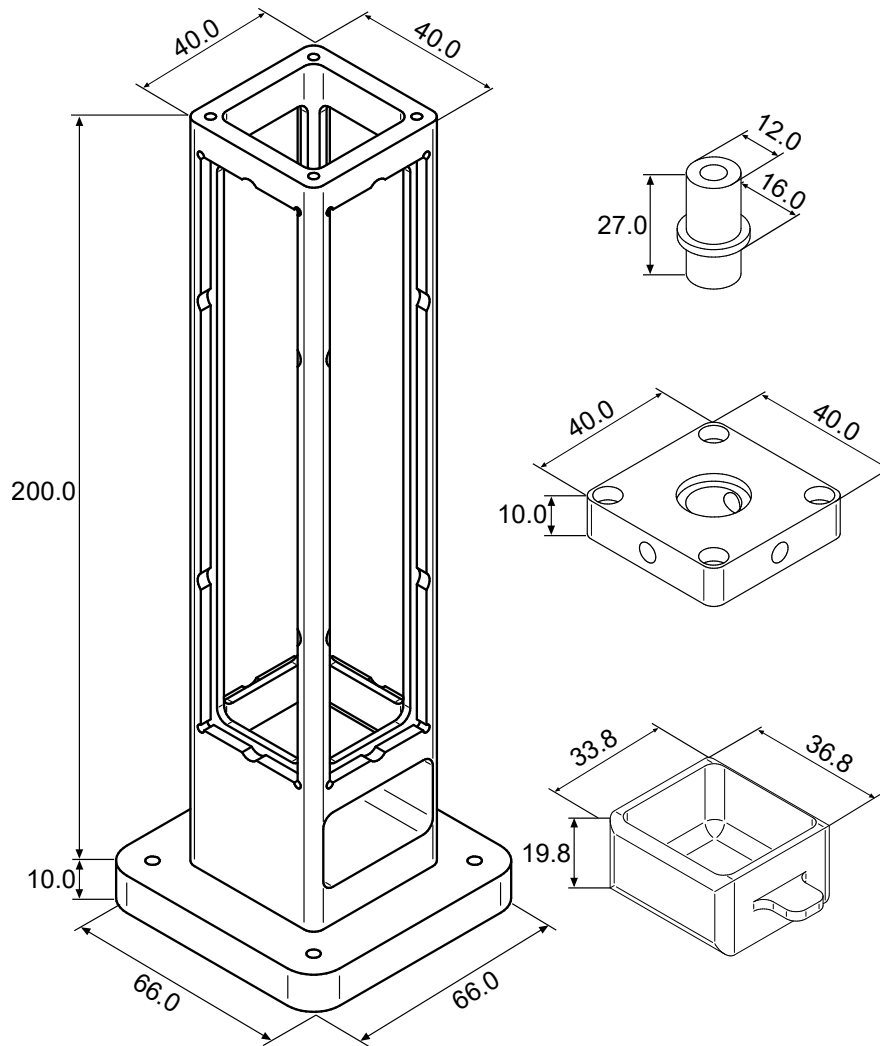


Figure 4.2: Drop tower components. All measurements are in mm.

4.3 Insertion mechanism

Design criteria: Repeatable release of particle; avoid inserting jet into drop tower

Designing a suitable insertion mechanism proved to be challenging as the particle size was small enough for electrostatic and Van der Waals forces to be significant. Typically, the particles would adhere to any surface they may come in contact with. Hence, simply releasing the particles from a pair of tweezers was not viable. An approach in which the particles were placed on a long metal rod with a conical tip was chosen. A concentric guiding tube made from Teflon and enclosed in an aluminum holder was fixed onto the observation volume and the rod inserted. Along the upper part of the rod were several cutouts that allowed the rod to be held at a specific height within the guide tube using a metallic key. By pulling the key, the rod fell freely downward until it collided with the bottom of the insertion mechanism. Importantly, the tip of the rod extended beyond the collision area and passed through a concentric hole on the bottom surface. The sudden deceleration of the rod due to the collision would then detach the particle as it retained its downward momentum. A major construction feature here was to run the sharp tip through an open section to allow the air pushed in front of the falling rod to dissipate. Without this open section, the accelerated air from the falling rod was creating a jet that would travel directly into

the observation volume. After several iterations, a suitable combination of open space and rod design was found and experimentally verified (see chapter 4.7). Importantly, a strong reduction of inserted air flow was aimed for and ultimately achieved. While the main goal of a reliable insertion mechanism with minimal disturbance of the experiment was achieved, the mechanism did not allow the initial particle orientation to be controlled. Instead, the particles would tumble after detaching from the rod tip. This could then be used to study the torque and damping of the tumbling of the particles, but required a careful choice of a minimum falling distance after insertion for the final velocity measurements.

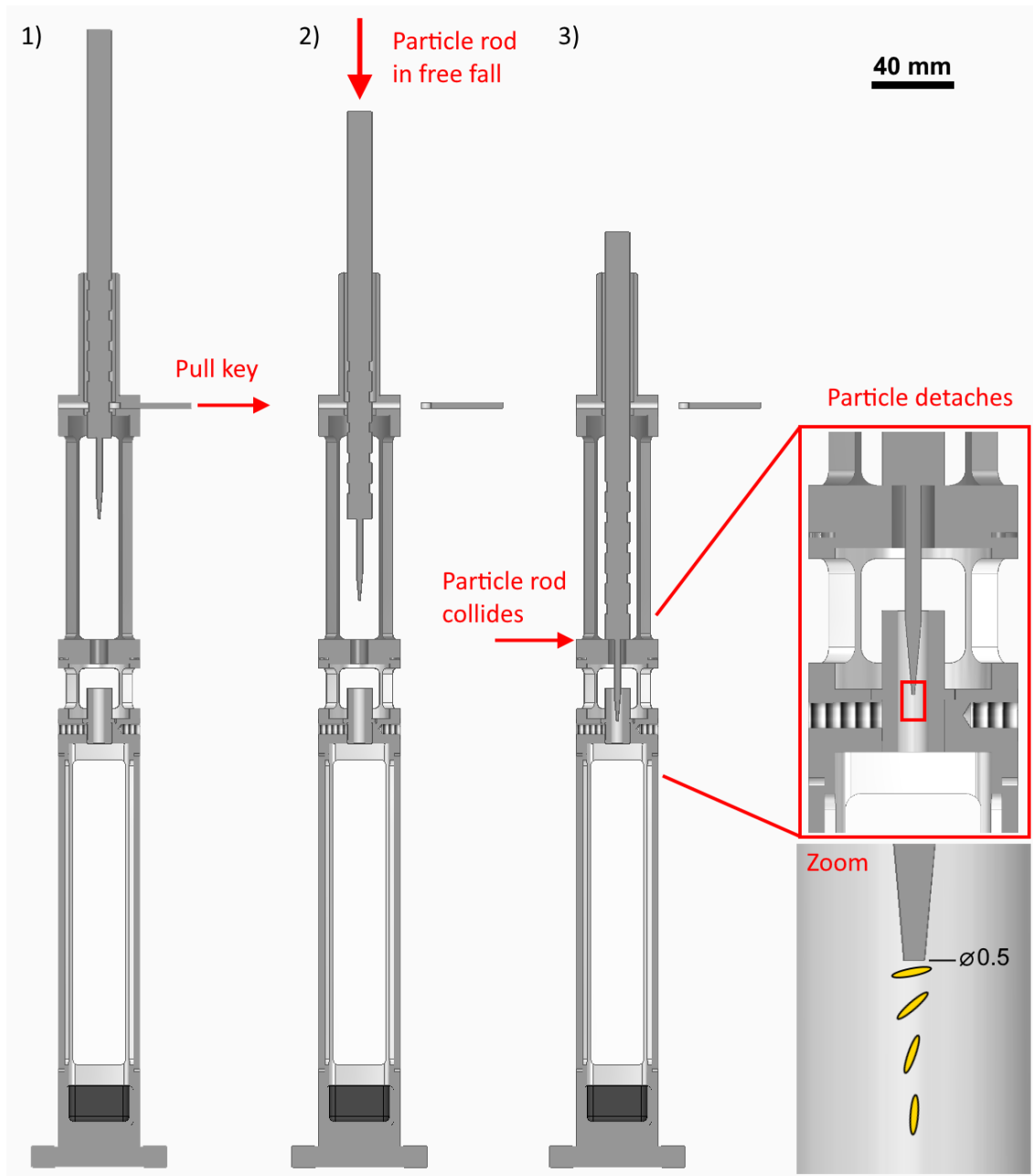


Figure 4.3: Insertion mechanism. **1)** A particle is placed onto the tip of the particle and the rod is secured at the desired height with a key. Then the key is pulled. **2)** The particle rod is now in free-fall, being guided by the Teflon coated guiding tube. **3)** The particle rod collides at the designated collision area, detaching the particle through sudden deceleration. Zoom-ins: close-ups of the particle detachment areas.

The insertion mechanism consists of the following components:

1. A 240 mm long stainless steel particle rod with a polished circular tip with a diameter of 0.5 mm. This circular tip will later be the area on which particles are placed. Above the particle holding area, the rod is expanding to a diameter of 3 mm over a length of 24.08 mm. It then expands in shape of a cylinder at 3 mm diameter for another 25.92 mm. At this point, the particle rod assumes its main diameter of 12 mm. This area forms a collision area that collides with another part of the insertion mechanism to quickly decelerate the particle rod and detach the particle off the rod. Further up, are 5 cut-ins that are used to secure the particle rod in position with a holding key. The cut-ins are spaced 10 mm apart to allow the user to reliably choose a falling height and collision velocity.
2. The particle rod is held by a guiding tube. The main feature of the guiding tube is a hollow Teflon tube with an outer diameter of 16 mm, an inner diameter of 12 mm and a length of 60 mm. The Teflon tube is press-fitted into an aluminium body that forms the second part of the guiding tube. It consists of a cylinder with the same length of 60 mm as the Teflon tube, an inner diameter of 16 mm (corresponding to the Teflon tube's outer diameter) and an outer diameter of 20 mm. Below the tube is a 40 mm × 40 mm × 10 mm plate that holds four through-holes for attaching the guiding tube to the rest of the insertion mechanism, a through-hole into which the holding key is placed to secure the particle rod and a circular cutout, concentric to the hole of the Teflon tube, allowing the particle rod to pass through the guiding tube unobstructed.
3. A holding key to secure the particle rod made of aluminium with a circular cutout concentric to the cutouts in the particle rod. It is used to secure the particle rod.
4. Below the guiding tube is the falling tube. This is a 120 mm long and 38 mm wide (inner diameter 30 mm) cut-open tube through which the particle rod is falling freely once the holding key is pulled. At the end of the falling tube lies small circular opening with a diameter of 8 mm through which only the tip of the particle rod can pass through. As the particle rod has a diameter of 12 mm at the designated collision area, it crashes into the bottom area of the falling tube at this point. The 40 mm long tip of the particle rod, however, was allowed to pass through. Of the 40 mm long tip of the particle rod, 30 mm protrude at the bottom of the falling tube when the particle rod finds its resting position.
5. A cut-open spacer with a length of 30 mm is placed under the drop tube. This is done to adjust the particle detachment position to be outside of the observation volume of the drop tower. The tip of the particle rod is, however, reaching into the top of the particle guiding tube of the lid of the drop tower. Therefore, the particle detaches in an area that is protected from lateral air flow.

4.3.1 Evolution of the insertion mechanism

The insertion mechanism went through several iterations before reaching its final design. Initially, the particle guiding tube of the drop tower's lid¹ was 70 mm long, instead of 27 mm as in the final version, and was used as a holder for a smaller diameter guiding tube. In the first few tests, these smaller guiding tubes were either made from metal or a glass pipette. The plan was then to release particles from tweezers and drop them into the small guiding tube. Metallic guiding tubes were quickly discarded for glass tubes, as glass held the promise of being smooth enough to keep the particles from sticking to the walls. Dropping the particles from tweezers, however, turned out to be impractical as the particles were small enough to stick to the tweezers and getting them off proved to be difficult. Therefore, a more reliable insertion mechanism was needed.

A first iteration of the collision detachment process as described above was inspired by another experiment that was aimed at dropping water drops onto a still water surface. There, a syringe

¹Note: the particle guiding tube is distinct from the guiding tube in figure 4.1 and is part of the lid.

was attached with a fixation piece to a smooth rod. After squeezing out a small drop that remained on the tip of the syringe needle, loosening a screw on the fixation caused the syringe to perform a guided fall that ended when part of the fixation piece collided with a metal plate. This would disconnect the drop from the syringe needle.

In an initial proof-of-concept, a particle was then placed on a dry needle tip and the crude drop mechanism tested. The first results were promising and therefore a more stable and repeatable design was desired. This led to the first version of the particle insertion mechanism. This version resembled the final version as detailed above, but lacked some necessary features. In particular, the falling tube was not cut open yet, the particle rod did not have an extended tip as it instead had an attachment point for placing a needle, and the spacer was missing. While this iteration performed reasonably well in providing a more reliable setup than the crude version, the needle tip was often not perfectly centred in the middle of the setup and, most importantly, the setup with its closed falling tube would induce a jet into the centre of the observation volume.

Strongly reducing the induced airflow was achieved by cutting open the sides of the falling tube and placing an open spacer between the falling tube and the lid. The spacer placed the detachment point of the particle high enough so that the particle rod would not enter the observation volume. The particle rod itself also went through two iterations, starting with the rod that had a mount for a syringe needle to a conical shaped tip to the slender tip that was introduced earlier. Finally, the insertion mechanism performed as expected. The validation itself of the insertion mechanism is covered in chapter 4.7.

4.4 Shadowgraphy

The observation of particles in free fall was implemented using shadowgraphy, where a light source shines on the object being observed and a camera opposite the light source captures the shadow of the object. This is a reasonably straightforward method to observe particles accurately, and it works decently well even for partially transparent objects such as the ellipsoids in this study.

4.4.1 High-speed cameras and LED light sources

Three Phantom VEO4k 990S high-speed cameras from Vision Research were deployed. These cameras are capable of recording up to 938 frames per seconds (fps) at full resolution (4096 x 2304 pixels), but can run at a higher frame rate when the resolution is reduced. With the minimum possible exposure time of 5 μ s, the cameras provided sufficiently sharp images. For the majority of experiments, a frame rate was chosen to be 2100 fps and the resolution was set at 4096 x 800 pixels. The trigger point was configured so that the camera recorded both for some time before and after the trigger. The cameras were used together in combination with 200 mm Nikon objectives and macro rings, which provided an additional $2.25 \times$ magnification. However, the macro rings were later omitted to allow for a larger field of view (FOV). Placement of the cameras in the desired positions was achieved by using macro stages and aluminium profiles. Different configurations were used for the terminal velocity (chapter 5) and the torque experiments (chapter 6). In the first case, one camera on top observed the insertion of the particle into the observation volume and two cameras observed the steady-state orientation and terminal velocity further down. The two cameras were aligned so that their relative angles was close to 90° . In the second case, the order was reversed and two cameras were used on top to observe the transient dynamics, also aligned at 90° , while one camera was placed below to record the terminal velocity as a control. All three cameras had their exposure times synchronised to ensure that images between the two 90° -apart cameras could be correctly matched later.

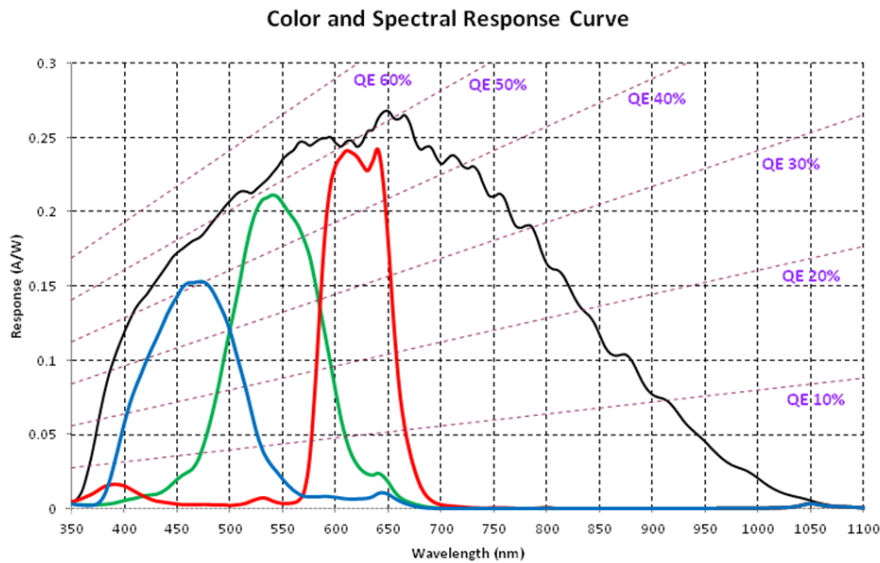


Figure 4.4: VEO camera spectral response, as specified by the manufacturer²

For the light sources, two LEDs of type LMH2 from company Cree were used together with Cree LMD800 LED module drivers and corresponding passive coolers. These LEDs can output 8000 lumens of white light, but require large passive coolers to do so. Comparing the LED relative spectral power shown in Figure 4.5 to the spectral response of the VEO4K 990S cameras shown in Figure 4.4, we see that the main peak of the LED spectrum lies reasonably close the peak of the camera spectral response.

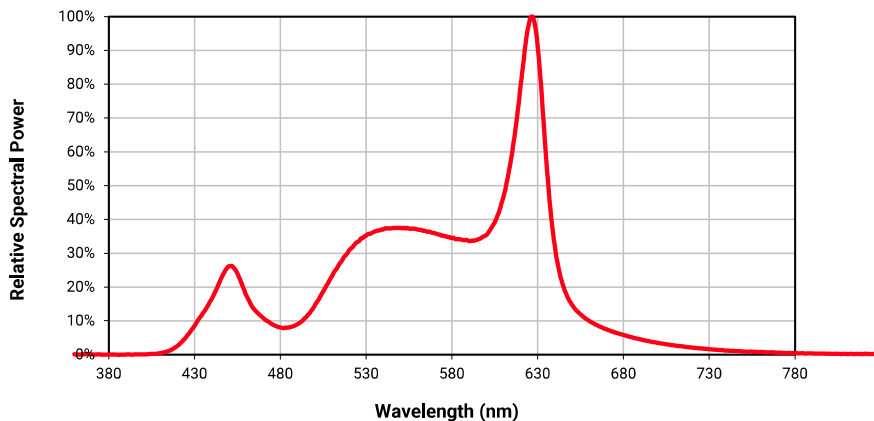


Figure 4.5: LED spectrum, as specified by the manufacturer³

4.5 Calibration

Calibration of the cameras: Two strategies were used for the calibration of the cameras. Both used a common Thorlabs calibration mask that displayed a grid pattern with a grid period of 500 μm apart and the use of a metal rod with a sharp end. This metal rod was modelled after the particle rod of the insertion mechanism, introduced in chapter 4.3. Prior to using the

²<https://www.phantomhighspeed.com/-/media/project/ameteksxa/visionresearch/documents/speccurve/all14kspec.pdf>, accessed 14.04.2021

³https://www.cree.com/led-components/media/documents/LEDModules_LMH2.pdf, accessed 14.04.2021

calibration mask, the sharp rod end would be used to approximate the central position inside the observation volume and the focus point of the cameras was centred on the tip of the rod. Then, for the first series of experiments to determine the terminal velocity and steady state of the free-falling particles, the calibration mask was placed in the centre of the observation volume using a fixture. The camera focus was then adjusted until the grid lines were at maximum sharpness.

For the second set of experiments focusing on the torque and change of angle during free-fall, a more advanced approach seemed necessary to accurately determine orientation changes. Here, the calibration mask was placed at a 45° angle to make it visible by both top cameras simultaneously. The calibration rod provided a point of reference on which both of the 90° apart cameras were focused so that their focal planes would overlap in the centre of the observation volume. The resulting images were then used in conjunction with a software solution created by Dr. Bagheri, MPI DS, to 3D-fit the calibration points and gain accurate information about the position of the particles in 3D-space.

4.5.1 Calibration procedure

Two calibration masks were used for the calibration of the cameras used in this work. One of them is a grid from the manufacturer Thorlabs of type R1L3S3P which is a grid distortion target with 10, 50, 100, and 500 μm grid spacings. The target was used for the droplet generator tests, covered in chapter 2, for the single-camera tracking videos and code covered in chapter 5, and initially for the multi-camera tracking as seen in chapter 6. Independently of the target used, the first step in the calibration procedure was to insert the calibration rod into the insertion mechanism, as detailed above. To use the grid target, the insertion mechanism was then removed from the top of the drop tower, as well as the drop tower lid. A metal fixation for optics was used as guidance to place the grid at the centre of the observation volume of the drop tower. In this configuration, the grid was orientated facing the first camera, an image was recorded and then the target was orientated to face the second camera where the procedure was repeated.

For the multi-camera tracking, a 3D-calibration scheme was sought after. In order to realize this, a 3D-printed holder for the grid target was created with a commercially available 3D-printer. This holder followed the modular design of the insertion mechanism and drop tower. After centering the drop tower with help of the calibration rod, the holder could be mounted on top of the drop tower instead of the lid. After recording images of the grid in this central position, the whole drop tower was then moved 1.80 mm away, normal to the plane of the grid, as indicated in figure 4.6. Images were recorded and the tower was repositioned back into condition where a set of reference images was created. Finally, the drop tower was moved in the opposite direction of the first movement and a set of images was recorded. Keeping track of the distances was achieved by using the x-stages in well-defined steps using their wheel-knobs that were rated 20 TPI (threads per inch), which corresponds to 1.270 mm per full rotation of the knob.

In later experiments, the grid target was replaced by another calibration mask which was a dotted target from the manufacturer Thorlabs of type R2L2S3P2 with a 500 μm grid spacing. Changing to dots simplifies the code that will be covered below.

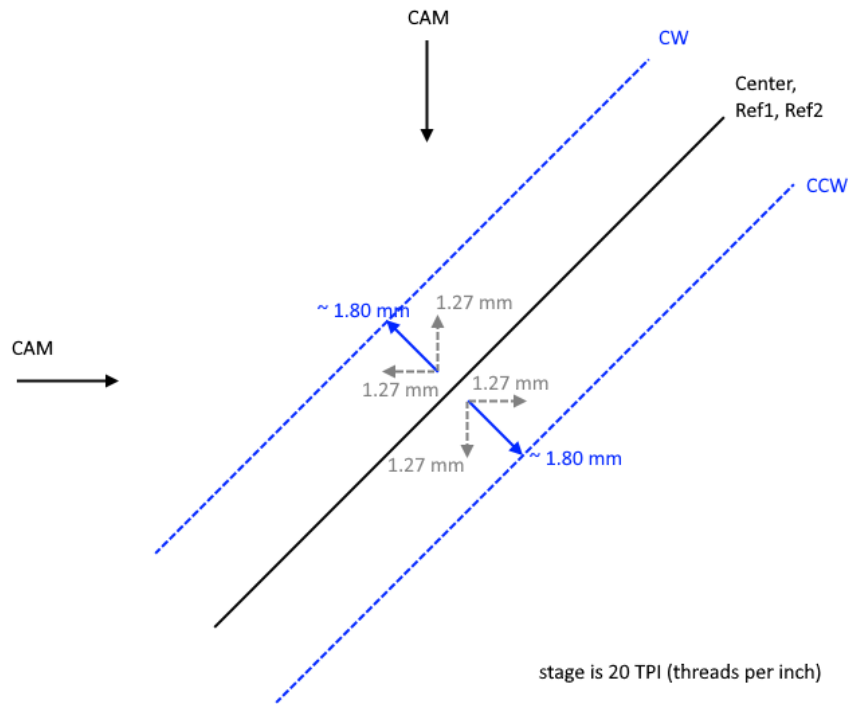


Figure 4.6: Calibration approach for a multi-camera setup. Five images would be taken beginning at the center position (Center), then moving the mask to the CW position. Next, the mask is moved back to the center and a reference image is taken (Ref1). After this, the CCW position is captured and, finally, another reference image is taken at the center position (Ref2).

4.5.2 Processing the calibration images

Depending on the target that was chosen (grid/dots), the calibration code performed the following steps:

- Grid target:** As the target consisted of a grid pattern but the code would expect dot coordinates, the intersection points of the grids had to be extracted from the images. For this step the software imageJ was used together with the plugin *Ridge Detection* (https://imagej.net/Ridge_Detection). Based on work from Steger [1998], *Ridge Detection* is used to extract lines found in an image and can automatically detect intersection. The plugin then provides the user with a list of lines and points. A problem was that in many cases the actual intersection between lines was not precisely defined by a provided intersection point, but was actually composed of a number of points, including duplicate points in several instances with the exactly the same coordinates. To sort out these duplicate and quasi-duplicate points (points that were close-by each other but not actual duplicates), a filter was created in the Python calibration code. The filter searched the list for points with the same coordinates and reduce the number of points at those coordinates to one. The next step was to define a radius and a search for points that were within the radius in respective to other points found. The point that was found first was kept and the others discarded. In principle, this solution would be sub-optimal as there was a chance that the first found point could be the least representative of the quasi-duplicate points. Yet this didn't appear to be an issue as 1) the amount of intersections was high (up to several hundred) and 2) the majority of quasi-duplicate points were very close to each other. The code then reported the angle between the two cameras to be 89.1° . Finally, the code always assigned the right-most and bottom-most intersection point as the reference for all other points.
- Dotted target:** While the grid target approach worked well enough, it ultimately proved to be unnecessarily complicated. A more straight-forward design was chosen with the dotted

calibration target. Here, the line and intersection point detection was replaced with a point-detection solution that was directly implemented in the same Python code that would handle later steps. *TrackPy*⁴ was chosen as an easy-to-use package for tracking features such as particles or, in this case, dots on a calibration mask. The package itself is based on work by Crocker and Grier [1996] and uses the following procedure: First, features in the provided image are located. For the dotted calibration mask this means that the algorithm is looking for dark circular features of a roughly specified size in pixels on a bright background. Second, an object containing all found features' positions and various parameters describing the features is returned. Among them is the so called *mass* of the features. This mass gives the inverse of the total brightness of the feature and is a measure for its size. Defining a minimum mass and filtering out features whose values lie below it, is helpful in preventing small dust particles from being counted as calibration mask points. The algorithm then offers many more options to track features across multiple images but this part was not essential for the calibration code. The reference point assignment was handled by asking the user to click a point on the image in question. The code would then compare the pixel coordinates of all features found by TrackPy and find the closest feature.

- **Both targets:** Once the relevant points (intersection points/dots) were identified, the remaining step was to match the now available sub-pixel coordinates to real-world coordinates. For example, the dots on the dotted target were spaced 500 μm apart from each other and this information had to be available for the sub-pixel coordinates. To do so, the coordinate points had to be sorted so that a definite order could be established. For example, given a reference point (0,0), the next dot to the right of the reference point would have the real-world coordinates (500,0) and the dot below the reference point would have the coordinates (0,-500). However, the targets were not always perfectly aligned with the camera and due to the 45 degree tilt. To assist the code in sorting and later matching both types of coordinates, options for removing shear and rotation of the calibration images were introduced. These options were exclusively used to sort the order of the points, but had no affect on the actual sub-pixel coordinates. The shear and rotation values were typically very low. In some cases, for example, the calibration image was rotated by $\pm 1\text{-}2\%$. At the end of the matching, the results were stored as a json file that could be read by other codes.

4.6 Measurement procedure

The following steps represent a typical day of measurements:

1. Prior to the experiments, the particles had to be printed. After the printing step was completed, the particles were stored in small plastic boxes and brought out of the clean room. To avoid contaminating the particles with dust, they were kept inside the plastic boxes until they were needed for the actual experiment. When picking the particles, rubber gloves were worn as to avoid fat or dust from the skin getting onto the particles.
2. Calibration of the cameras was performed as described in chapter 4.5. The cameras were then operated further, but the LEDs were turned off to prevent heating up of the experimental setup.
3. The particles were then removed from the glass substrates on which they had been printed using plastic tweezers that had been cleaned in an ultrasonic bath prior to the experiment.
4. Dropping the particle rod through the insertion mechanism a few times lifted most the dust from it and any remaining dust was removed by wipes soaked in isopropyl alcohol (IPA).
5. A single particle was then placed onto the polished tip of the particle rod with the plastic tweezers. A magnifying glass was helpful here.
6. The particle rod was secured at the desired falling height via the holding key.

⁴<https://soft-matter.github.io/trackpy>

7. A current session reference (CSR) was run on all the cameras to remove background intensity patterns and recording enabled.
8. LEDs were turned on, the holding key pulled and cameras triggered, all within a few seconds of each other.
9. LEDs were turned off immediately afterwards to avoid heating up the experiment.
10. Lastly, the empty frames of the recorded videos were cut and the rest saved to disc. Hereafter, the next experiment could be prepared, starting from step 5.

4.7 Validation of the experimental setup

4.7.1 Flow insertion

After realizing the issue of inserting a jet into the observation volume of the drop tower, as detailed in chapter 4.3.1, the final design needed to be validated. The observation volume was filled with fog by use of a nebulizer. Light from a laser was led through an optical element to create a laser sheet that was normal to the camera recording direction. Reducing the zoom by replacing the 200 mm objective with a 54 mm objective and adjusting the frame rate to slower values allowed sufficient field of view and light to monitor the flow field of the fog, when inserting a particle. After insertion of the fog, some time had to pass for the fog to stop moving, though a complete stop of movement was not achievable as the fog would eventually evaporate. The leftover movement was deemed acceptable when movement occurred on the order of few mm/s as the order of the velocity of the particles would be closer to $O(1)$ m/s. No development of heat plumes was observed inside the fog, displaying visible differences between highly seeded areas and regions that were relatively devoid of aerosols. Four tests were executed subsequently:

1. **Unsealed drawer, no particle:** The drawer was placed into the slot at the bottom of the drop tower and the insertion mechanism was used several times without a particle attached to the particle rod.
2. **Sealed drawer, no particle:** Similar to test one, but this time the small gaps on the edges of the drawer were sealed with tape to eliminate any airflow that could pass through there.
3. **Sealed drawer, big particle:** The drawer was sealed as in test two and a large particle was released with the insertion mechanism. The particle had the shape of a six-legged cross with an outer diameter of 1.22 mm.
4. **Sealed drawer, small particle:** As before, but the particle inserted was a $EL=0.125$, $FL= 1.0$ ellipsoid with a longest dimension of 0.25 mm.

The experiments revealed three major findings: First, the structure of the smoke was not affected by the insertion mechanism. The collision caused by letting the particle rod drop from the maximum height did not insert detectable flow into the observation chamber, as long as the drawer was sealed with tape. In cases where the smoke still had some leftover movement after the initial wait period, the insertion mechanism did not accelerate the smoke's movement. Second, when the drawer was not sealed, a plume developed on the side of the observed area, hinting at the importance of the seal. Third, the smoke structure was strongly affected by inserting the big particle. Results of test three, that exhibited a particularly strong interaction of the inserted particle with the flow structure of the smoke, is shown in Fig. 4.7.

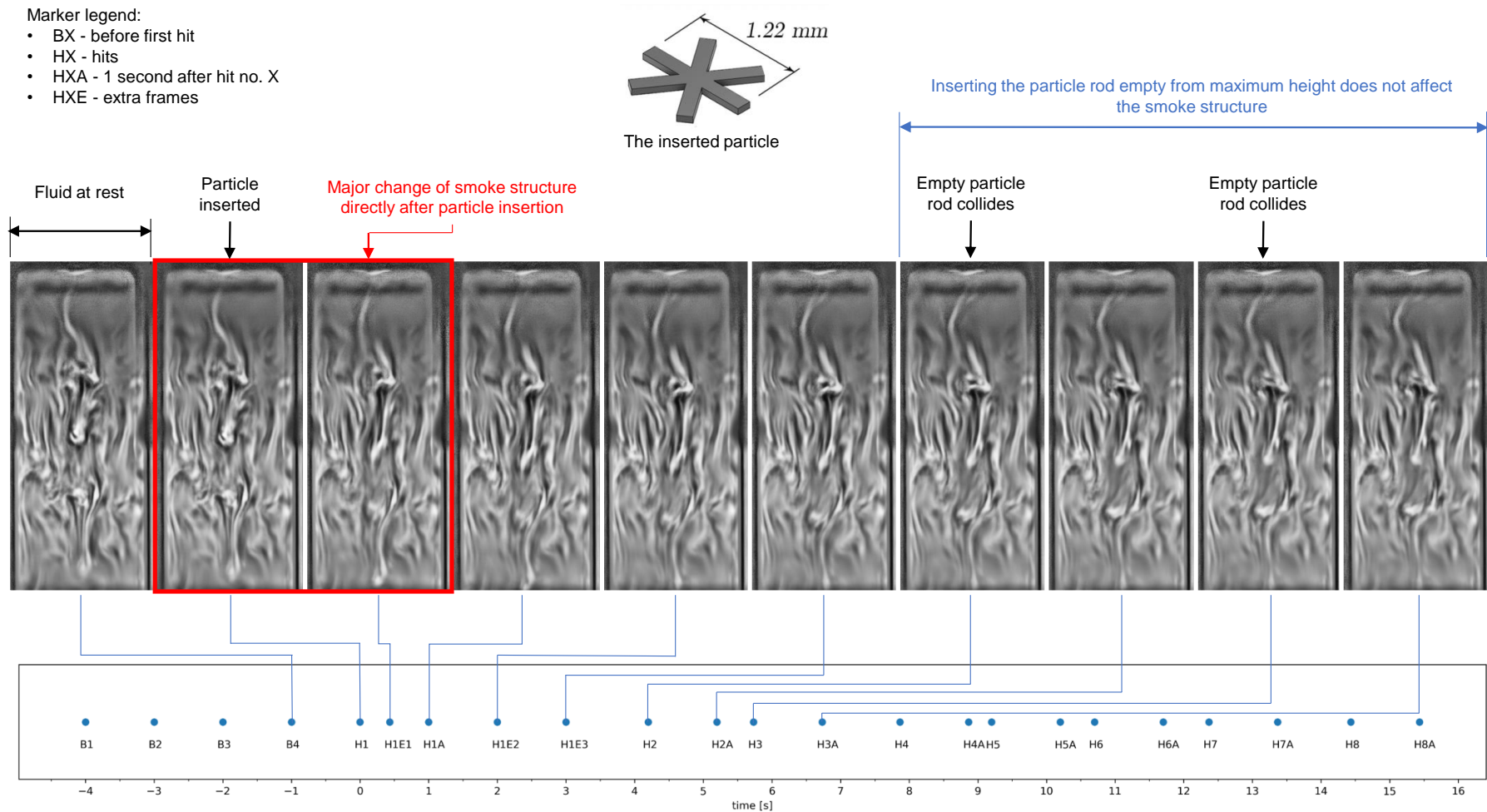


Figure 4.7: Results of insertion test 3: Sealed drawer, big particle. Here, a large particle was released with the insertion mechanism. The particle had the shape of a six-legged cross with an outer diameter of 1.22 mm. Markers B1-B4 are time stamps 4, 3, 2 and 1 seconds before the first time the particle rod collides inside the insertion mechanism after falling freely from maximum height, which is marked with H1. The markers "H" (hit) mark collisions. The flow structure of the smoke is at rest during markers B1-B4. A large change of the flow structure is visible at markers H1E1 (433 ms after H1) and H1A (1 s after H1) compared to marker H1. This was caused by the particle passing through the smoke and affecting it. Subsequent collisions of the empty particle rod falling from maximum height (markers H2-H8 and markers H2A-H8A), do not display any change in the flow structure. This means that the insertion mechanism is working as intended and no relevant amount of flow is inserted into the observation volume during its operation, even when dropping the particle rod from the maximum height possible. Images for all markers can be seen in appendix B, figure B.1.

4.7.2 Temperature measurements

To estimate how much the setup heats up due to the LEDs, several temperature measurements were carried out. The measurements involved two different profiles of switching the LEDs on and off:

- **Profile 1:** The LEDs were switched on at the start of the measurement and then remained lit for 50 minutes. The temperature drop was then monitored for another 64 minutes.
- **Profile 2:** The LEDs were repeatably turned on and off according to the following pattern: 60 seconds off, 15 seconds on, 60 seconds off, 15 seconds on, and so forth. The last time the LEDs were turned on was at the 60 minute mark. The temperature drop was then recorded for another 18 minutes.

Recording the temperatures was done with a sensor of type AM2315 - Encased I2C Temperature/Humidity Sensor, that was connected to an Arduino Mega. The sensor was placed into the centre of the observation volume. Figure 4.8 shows both recordings. The maximum temperature reached was 30.4°C for profile 1 and 24.0°C for profile 2. Profile 2 is showing a notably slower climb in temperatures. During the actual experiments, the times between dropping particles was often well beyond 60 seconds and the 15 seconds estimated time of running the LEDs was held consistent. Therefore, the heating of the experimental setup was indeed minimal and should not have influenced the experiments significantly.

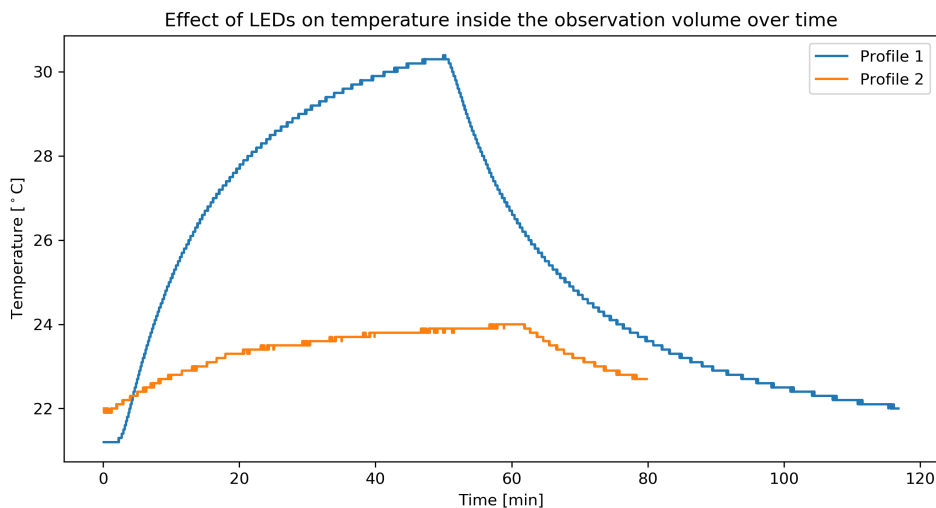


Figure 4.8: Effect of LEDs on observation volume temperature

4.7.3 Terminal velocity of spheres

To verify the validity of using particles created with TPP for settling experiments, ten spheres of diameter 140 μm were dropped individually into the settling chamber and their terminal velocities v_t recorded 70.4 mm after the insertion point. This was well below the distance of 51.4 mm that the particles would require to reach 95% of their terminal velocity v_t , as estimated by an empirical model for the prediction of terminal velocities and drag coefficients from Bagheri and Bonadonna [2016]. Fig. 3.9 d) showed a typical trajectory of a spherical particle in free-fall.

Comparing the terminal velocities v_t of the spheres with the expected values from the empirical model of Bagheri and Bonadonna [2016], the experimental data (camera 1 median of all 10 experiments $v_{t1} = 0.528 \text{ m/s}$, std. dev. $\sigma_1 = 0.011 \text{ m/s}$, camera 2 median $v_{t2} = 0.532 \text{ m/s}$,

std. dev. $\sigma_2 = 0.010$ m/s) was found to be in good agreement with the model (value from model: $v_t = 0.523$ m/s) with a difference of 2.1–2.3% (see also table 5.3 in Ch. 5.6.2). The good agreement between the prediction and the measured terminal velocities for the spheres was a strong indicator for the suitability of the setup for settling experiments. As the settling chamber was deemed to provide a well controlled environment, the study of non-spherical was continued. Results of these experiments will be covered in the next two chapters.

Chapter 5

Free-fall dynamics of highly inertial ellipsoids: Single camera tracking

5.1 Introduction

One of the initial goals of this work was to verify the empirical drag coefficient model that was created by Bagheri and Bonadonna [2016]. To this end, all nine shapes of particles (chapter 3.4) were dropped into the settling chamber and observed several centimeters below the insertion point. The initial assumption for the correct observation point was informed by the drag coefficient model. Table 5.1 shows estimates for the correct distance to achieve terminal velocity, depending on the particle shape. The model takes into account several parameters, such as particle equivalent diameter, particle-to-fluid density ratio, particle shape, fluid dynamic viscosity and fluid temperature. The model that was used for the comparison with the recorded data was 'bb_full'.

Shape	d_{95} [mm]	v_t^{BB16F} [m/s]	Re_p
EL 0.2, FL 0.2	23.1	0.236	2.19
EL 0.2, FL 0.5	34.9	0.321	2.98
EL 0.2, FL 1.0	44.5	0.396	3.67
EL 0.5, FL 0.2	31.8	0.308	2.85
EL 0.5, FL 0.5	43.9	0.405	3.75
EL 0.5, FL 1.0	50.4	0.479	4.44
EL 1.0, FL 0.2	38.2	0.365	3.38
EL 1.0, FL 0.5	48.3	0.459	4.26
EL 1.0, FL 1.0	51.4	0.519	4.81

Table 5.1: Distances d_{95} to reach 95% of the terminal velocities v_t and particle Reynolds number for each shape. Calculated using the full empirical model "bb_full" from Bagheri and Bonadonna [2016], abbreviated "BB16F".

At this point a full 3-D tracking model (Ch. 6) was not developed yet, so a simpler code was created for the terminal velocity measurements. This code considered only the 2D movement through a camera focal plane into account and estimated the mean velocity of the particles by using the camera frame rate (i.e. 2000 frames per second), the number of frames (i.e. 100 frames) and the travel distance (in mm). There was no 3D fitting of the two camera views as of yet. However, this code base became a first implementation for several of the methods that were later used in the full 3D tracking code. For example, adaptive thresholding was applied to the camera frames and ellipsoid contours were found by taking into account the intensity gradient between neighboring pixels.

Figure 5.1 gives an overview of the single-camera tracking code in the form of a flow chart .

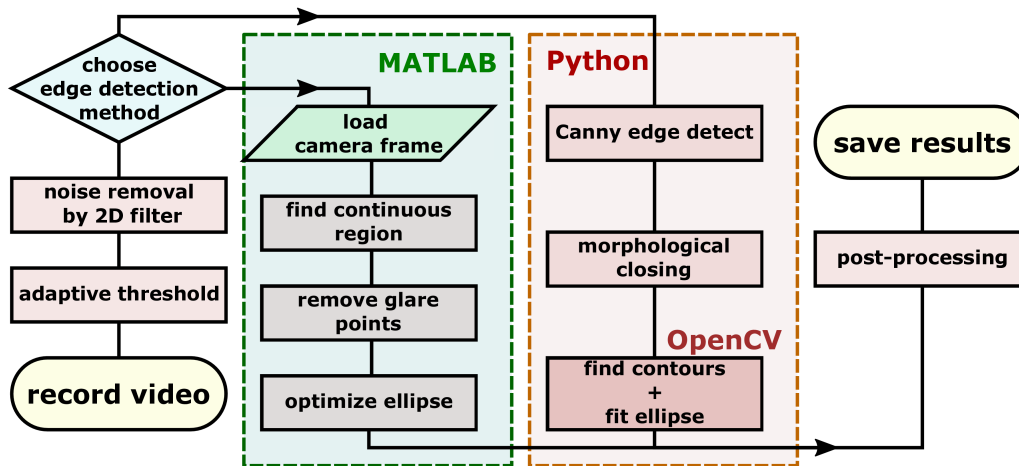


Figure 5.1: Flow chart of the single-camera tracking code. Several steps are performed in a sequential order, beginning with reading in the recorded video, applying an adaptive threshold to them and removing noise with a 2D filter. At this point, a choice is made whether to use the Python or the Matlab implementation for edge detection of the particle contours. The Matlab implementation will attempt to find a continuous region of pixels that make up the particle and remove any glare points that are caused by the partial transparency of the particles. It then calls a function to find and optimize the fit of an ellipse around the found region. The Python implementation utilizes *Canny edge detect* to find the contours of the particle and performs a morphological closing operation that has the same effect as the glare point removal in the Matlab implementation. It passes the contours found with *Canny edge detect* to the OpenCV library to fit an ellipse. After each of the two implementations, a set of parameters of the found ellipses is submitted to the main loop of the code. The data is then filtered for jumps in the angles of the recorded ellipses in a post-processing step. Finally, the results are stored as plots and in a json file.

5.2 Structure of the code, Python and crucial libraries

Most of the code written during this thesis was realized in Python 3, with the exception of the code covered in chapter 5.4 that was realized in Matlab. Python is an object-orientated general purpose programming language that offers multiple levels of organizational structure such as functions, classes, modules and packages [Kuhlman, 2009]. Python offers very easy access to thousands of third-party libraries via its *import* statement. Libraries can alter and extend the scope of Python tremendously, with many of them being both open-source and free of charge to use. In the following, the most utilized libraries in this work are introduced. This applies both for chapter 5 as well as chapter 6.

- NumPy, <https://numpy.org/>: By its own words, NumPy aims to be the "fundamental package for scientific computing in Python". It offers a specific object, the *ndarray*, which differs from standard Python lists by having a fixed size and requiring all elements in the array to be of same data type, thus having the same size in memory, unlike Python lists, that can grow organically. In the background, the *ndarray* and other NumPy routines are handled with a pre-compiled C code. This way, using NumPy allows the user the same ease-of-use as normal Python while taking advantage of the faster computational speed that is offered by the C language. In addition, NumPy supports a wide range of numerical computing tools

such as specific mathematical functions, random number generators or Fourier transforms. NumPy is distributed under the open-source Berkeley Software Distribution (BSD3-clause) license. A recent review article on NumPy was published by Harris et al. [2020].

- Matplotlib, <https://matplotlib.org/>: A 2D graphics package with a wide range of uses such as generation of plots, application development or interactive scripting [Hunter, 2007]. Matplotlib is distributed under the open-source Python Software Foundation (PSF) license.
- SciPy, <https://www.scipy.org/>: An open-source ecosystem for scientific computing that contains several popular packages, among them NumPy and Matplotlib. The ecosystem also provides the *SciPy library*, a collection of numerical algorithms and toolboxes that can be used for a wide range of applications such as optimization, signal processing or statistics. The SciPy library is operating under the open-source SciPy license. Part of the overlaying *ecosystem* is also *Jupyter notebook*, a console-based approach of interactive computing and editor for programming in the Python language. The majority of the code in this work was written within such Jupyter notebooks.
- OpenCV, <https://opencv.org/>: The Open Source Computer Vision Library (OpenCV) was created as an infrastructure for computer vision applications and to accelerate machine learning [Bradski, 2000]. It offers more than 2500 optimized algorithms from common operations, such as cropping or rotating images, to much more involved procedures, such as object tracking, deep learning or facial recognition. OpenCV is available for the programming languages C++, Java, Matlab and Python. It is distributed under the open-source Berkeley Software Distribution (BSD3-clause) license.

5.3 Python implementation

After the initial loading, an adaptive threshold would be applied to the frames, followed by a 2D filter to remove noise and an edge detection algorithm. In the initial implementation of the code, `cv2.canny` was used for the edge detection.

Canny edge detect is an algorithm developed by John F. Canny in 1986 [Canny, 1986]. It utilizes several stages to detect the edges of particles, starting with a noise reduction by applying a 5 x 5 Gaussian filter and then finding the intensity gradient of the image. This is followed by a step called the "non-maximum suppression", which removes pixels that may not constitute the edge. This mainly concerns local intensity maxima, which could be mistaken for the particle edge. An hysteresis threshold is finally applied. This threshold uses two values: a minimum and a maximum threshold value. An intensity above the maximum value should be an edge and a value that is below the minimum value should be a non-edge, and therefore is discarded. Values between these thresholds are either classified as an edge or non-edge, depending on their connectivity with other, verified edge-points¹.

After *Canny edge detect* was executed, the code with an operation called morphological closing ensued, which closed any holes that appeared inside the now thresholded image. Lastly, the two OpenCV functions `cv2.findContours` and `cv2.fitEllipse` were run, resulting in a set of parameters describing the location and shape of the fitted ellipse.

However, since `cv2.canny edge detect` was showing non-optimal results in identifying the edges of some of the particles in the experiment, it was later replaced by a self-implemented intensity gradient approach. One possibility is that the inherent transparency of the particles, combined with the blurriness of some of the images (due to the particle being outside of the focal plane),

¹https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_canny/py_canny.html, accessed 14.04.2021

led to *Canny edge detect* having problems detecting the edges of the particles correctly in this implementation.

5.4 Matlab implementation

This part of the code was created by Dr. Jan Molacek, and was created in Matlab instead of Python, like the rest of the code covered in this work. The code was compiled using a similar approach to the Canny edge detect method that was covered prior, and was composed of the following functions:

Process_AllVideos

The main loop. Reads all videos and sends them to the next function, `Process_Video`, that will process the individual videos.

Process_Video

The core of the Matlab implementation. It's job is to extract all ellipsoidal parameters from the given particle images in the videos. It needs to be provided with the video name including the file extension, a desired output folder, where the results will be stored as .png and .txt files, and the desired format of the .txt file, which can either store the result values separated by spaces without a header or separated by a semicolon with a header. In preparation of running the function, a circular filter is defined for later use and several tunable parameters are stated. While running `Process_Video`, the following sub-functions are called (in order): `Read_Header`, `Read_Frame`, `Get_ContinuousRegion`, `Get_GlarePoint`, `Optimise_Ellipse`. These functions will be covered in the following.

Read_Header

This function deals with reading in information from the camera files that come in the .cine format. It extracts information such as the image height and width, amount of frames in the video, area of the image, and the frame rate.

Read_Frame

This function loads a single frame of the video. It can be used for reading in 8 or 16 bit images and has an option to read in either cine files, mcin fires, or gmv files. Here, it was used with the option for cine files, the default file format of the cameras that were utilized.

Get_ContinuousRegion

Finds the largest connected set of pixels, starting from the maximum value of pixels that lie above a given threshold intensity. It has an option to state a maximum allowed size in number of pixels for the final set and returns the largest continuous region found.

Get_GlarePoint

This function deals with the specific issue of the material used for the settling velocity experiments. The issue being that the material is slightly transparent with a yellow tint. When using shadowgraphy, some of the light can pass through regions in the ellipsoids, causing glare points that have a high intensity in the recorded image and are therefore not recognized as part of the ellipsoid. The function uses the continuous set that was created in the prior function, `Get_ContinuousRegion`. It further utilizes a sub-function called `PixMissing`. `PixMissing` estimates the amount of pixels that would be needed to close the apparent hole in the ellipsoid caused by the glare point. Specifically, it tries to close regions that lie between two or more continuous sets.

Optimise_Ellipse

Optimizes the parameters of an ellipsoidal fit of a particle contained in an image. Within this function, the sub-function `Fit_Ellipse` is being solved with the in-built MatLab routine `lsqcurvefit`. `lsqcurvefit` solves non-linear curve-fitting problems by using least squares minimization. `Fit_Ellipse` outputs the intensity value at X of a ellipsoidal image fit where the ellipse is described as follows :

$$PAR(3) * x^2 - 2 * PAR(4) * x * y + PAR(5) * y^2 \quad (5.1)$$

where

$$x = x_{pos} - PAR(1), y = y_{pos} - PAR(2) \quad (5.2)$$

and the parameters are:

- PAR(1) : x-position of circle center
- PAR(2) : y-position of circle center
- PAR(3) : cxx coefficient
- PAR(4) : cxy coefficient
- PAR(5) : cyx coefficient

At the end, the function returns the best fit of an ellipse to the provided image to `Process_Video`. An example of the quality of the edge detection achieved with the Matlab implementation can be seen in figure 5.2.

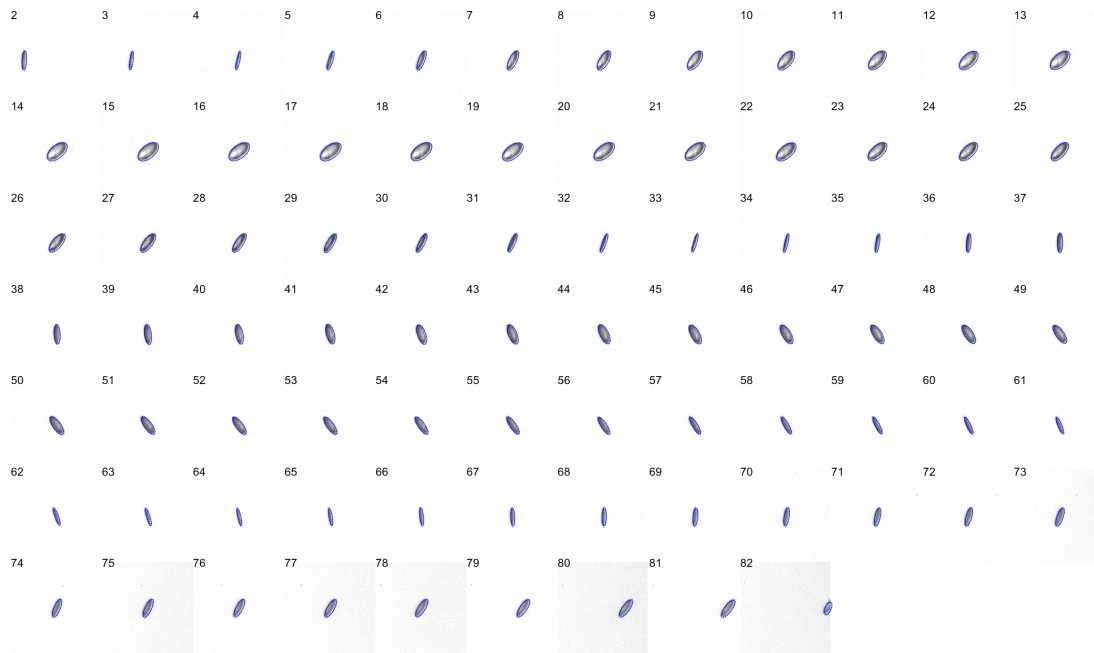


Figure 5.2: Example of ellipse fitting results created with the Matlab implementation for a particle of shape EL 0.5, FL 0.2. Shown are the original particle images overlaid with the detected contours (blue). There is little to no visible difference between the real and the detected edges, indicating a good fit.

5.5 Post processing and plotting

As a final step, the change of angles of the ellipsoids was calculated and jumps in the data were removed. Specifically, the code cannot distinguish between the front or back tip of an ellipsoid as only the shadow of the ellipsoid is recorded, which has no other directional indicators. Therefore, the code would assume a jump between steps if the values of the extracted angles would suddenly shift by $\pm 180^\circ$. Filtering out these jumps extracts the extrinsic rotation of the ellipsoids when observed from a coordinate system fixed outside of the ellipsoid. Combined rotations about more than one axis of the ellipsoids would not be represented in this type of measurements.

Finally, several plots were produced such as trajectories, the change of angle over frame number, and combined images that show all the recorded ellipsoids over the falling distance. These plots and other results can be found in the next chapter.

5.6 Results

5.6.1 Data set

This set of experiments used the following camera configuration:

- One camera of type Phantom Miro 341, from company Vision Research, was recording the falling particles from 20 mm below the insertion point onward. The observed distance was approximately 14.4 mm.
- Two cameras of type Phantom VEO 4K 990 L, from company Vision Research, 90° apart, were recording the falling particles from 70.4 mm below the insertion point onward. The observed distance was approximately 11.2 mm.

- The cameras were used with a 200 mm objective and two 1.5 x magnification rings each. This lead to resolutions of:
 - Camera 1 (top): 5.62 $\mu\text{m}/\text{px}$
 - Camera 2 (bottom): 2.732 $\mu\text{m}/\text{px}$
 - Camera 3 (bottom): 2.73 $\mu\text{m}/\text{px}$

Table 5.2 shows the composition of the analyzed data set.

Shape	Dimensions [μm]	Re_p	no. <i>exp</i>
EL0.2 FL0.2	700.0 x 140.0 x 28.0	2.19	24
EL0.2 FL0.5	515.8 x 103.2 x 51.6	2.98	10
EL0.2 FL1.0	409.4 x 81.9 x 81.9	3.67	21
EL0.5 FL0.2	380.0 x 190.0 x 38.0	2.85	10
EL0.5 FL0.5	280.0 x 140.0 x 70.0	3.82	19
EL0.5 FL1.0	222.2 x 111.1 x 111.1	4.44	10
EL1.0 FL0.2	239.4 x 239.4 x 47.88	3.38	10
EL1.0 FL0.5	176.4 x 176.4 x 88.2	4.26	10
EL1.0 FL1.0	140x0 x 140.0 x 140.0	4.90	10

Table 5.2: Single-camera tracking: Dimensions (largest x intermediate x smallest), particle Reynolds number Re_p and number of experiments for each shape. The shape descriptors elongation (EL) and flatness (FL) were introduced in chapter 3.4.1.

5.6.2 Terminal velocity

The first parameter of interest to extract from the data, was the terminal velocity with the aim of comparing it to the empirical model of Bagheri and Bonadonna [2016]. Extracting it from the data was done by calculating the mean velocity of the particles as follows:

$$v_t = \frac{x_{end} - x_{start}}{n_{frames} * t_{step}} \quad (5.3)$$

where x_{start} and x_{end} are the pixel positions of the center of the detected first and last ellipses along the falling direction. The number of considered frames is given by n_{frames} , excluding empty frames before or after x_{start} and x_{end} , if any occurred. Lastly, t_{step} is the time between two frames, calculated from the frame rate of the camera.

In principle this velocity is only providing the mean velocity of the recorded video. Therefore, depending on the particle shape, a minimum distance after the insertion point had to be considered before measuring the velocity. The choice of this minimum distance was informed by the empirical model from Bagheri and Bonadonna [2016], with which the measured final velocities were also compared. The calculations were made for air at room temperature of $T = 20^\circ\text{C}$ at sea level atmospheric pressure $p=101.325 \cdot 10^3$ Pa, fluid density $\rho_{fluid} = 1.205$ kg/m³ and fluid dynamic viscosity $\nu = 1.8 \cdot 10^{-5}$ Pa·s. The particle density $\rho_p = 1200.0$ kg/m³, particle equivalent diameter $d_{eq} = 140$ μm and a particle initial velocity of $v_{initial} = 0.7$ m/s were chosen as the parameters to define the particle. Gravity was assumed to be $g=9.81$ m/s². According to the model, in order to reach 95% of the terminal velocity, the particles would need to fall a minimum of 23.1 mm in the case of the shape EL 0.2, FL 0.2 (an elongated and flat ellipsoid), up to a distance of 51.2 mm for the shape EL 1.0, FL 1.0 (a sphere). Particles were then decided to be observed 70 mm after the insertion point to be on the safe side.

The terminal velocities were then extracted from the experiments mentioned in table 5.2. Median and mean velocities of all videos were calculated for both the two 90° apart cameras. Differences in measured and expected terminal velocities were evaluated per camera each and as a combined value. The results of the comparison are shown in table 5.3.

The shape EL 1.0, FL 0.5 (a bulky disc) showed the best agreement when comparing the median (difference of 0.86%), as well as the mean (difference of 0.22%) recorded terminal velocities with the model. The largest differences were found for shape EL 0.2, FL 1.0 with differences of 12.50% for the median and 10.47% for the mean terminal velocities. Spheres (EL 1.0, FL 1.0) were fairly accurately predicted with differences of 2.07% (median) and 2.26% (mean). When taking into account all 9 shapes, the median differences were 2.12% for the median velocities and 5.13% for the mean velocities. This indicates a good match between the empirical predictions and the measured results and validates the suitability of the model for use with the chosen particles.

Shape [EL,FL]	Terminal velocity v_t [m/s]							Difference calculated to measured [%]					
	BB16F	Cam 1			Cam 2			Cam 1		Cam 2		Avg. both	
		Med.	Mean	Std. Dev.	Med.	Mean	Std. Dev.	Med.	Mean	Med.	Mean	Med.	Mean
0.2,0.2	0.236	0.240	0.220	0.068	0.242	0.246	0.011	1.67	7.27	2.48	4.07	2.07	5.67
0.2,0.5	0.321	0.295	0.296	0.003	0.295	0.297	0.004	8.81	8.45	8.81	8.08	8.81	8.26
0.2,1.0	0.396	0.351	0.361	0.027	0.353	0.356	0.008	12.82	9.70	12.18	11.24	12.50	10.47
0.5,0.2	0.308	0.301	0.303	0.007	0.314	0.337	0.055	2.33	1.65	1.91	8.61	2.12	5.13
0.5,0.5	0.405	0.393	0.394	0.089	0.413	0.427	0.055	3.05	2.79	1.94	5.15	2.50	3.97
0.5,1.0	0.479	0.482	0.481	0.005	0.487	0.485	0.006	0.62	0.42	1.64	1.24	1.13	0.83
1.0,0.2	0.365	0.347	0.346	0.006	0.345	0.343	0.012	5.19	5.49	5.80	6.41	5.49	5.95
1.0,0.5	0.459	0.463	0.458	0.015	0.463	0.460	0.015	0.86	0.22	0.86	0.22	0.86	0.22
1.0,1.0	0.519	0.528	0.529	0.011	0.532	0.533	0.010	1.70	1.89	2.44	2.63	2.07	2.26

Table 5.3: Comparison between the recorded terminal velocities and the empirical drag model of [Bagheri and Bonadonna, 2016]. The terminal velocities of all 9 tested shapes are compared to the predictions of the full model (BB16F). Both median and mean values of the recorded velocities are presented per shape with the amount of videos as indicated in table 5.2. The stated standard deviations were attained by comparing the mean velocities for every set of videos, e.g. 24 videos per camera for shape EL0.2 FL0.2. Next, the difference between the median and mean velocities is calculated and shown as a difference in percent. Finally, the average of the values of both cameras are shown in the last columns.

5.6.3 Steady-state orientation and rotation rates

As covered in chapter 1.3, within *Stokes regime*, the majority of particles with well-defined symmetry properties have no preferred orientation during free-fall. Therefore, a random orientation should be assumed. On the other hand, particles tend to fall with the largest projection area normal to the falling direction, when their particle Reynolds number increased to $Re_p \approx 100$ [Bagheri and Bonadonna, 2016]. As the tested particles fell into the range $Re_p = 2.19 - 4.90$, they would be found in the *intermediate* regime, which marks the transition between Stokes' and Newton's regime. Therefore an assumption could be made that the orientations should be found somewhere between random orientation and settling with the largest projection area first.

Examples of typical particle trajectories and orientations are shown in figure 5.3 (top camera) and figure 5.4 (bottom cameras). Particles displayed stronger changes in orientation between frames in the top camera as compared to the bottom cameras. In the top view, several particles display apparent oscillations along their longest axis, with some particles exhibiting tumbling behavior, such as shape EL 0.5, FL 0.2. At 20 mm after the insertion, particles typically exhibited oscillations, tumbling or tumbling behavior transitioning into oscillation. This tumbling/oscillations would then slow down and at 70 mm from the insertion particles generally tended to fall with their largest projection area first.

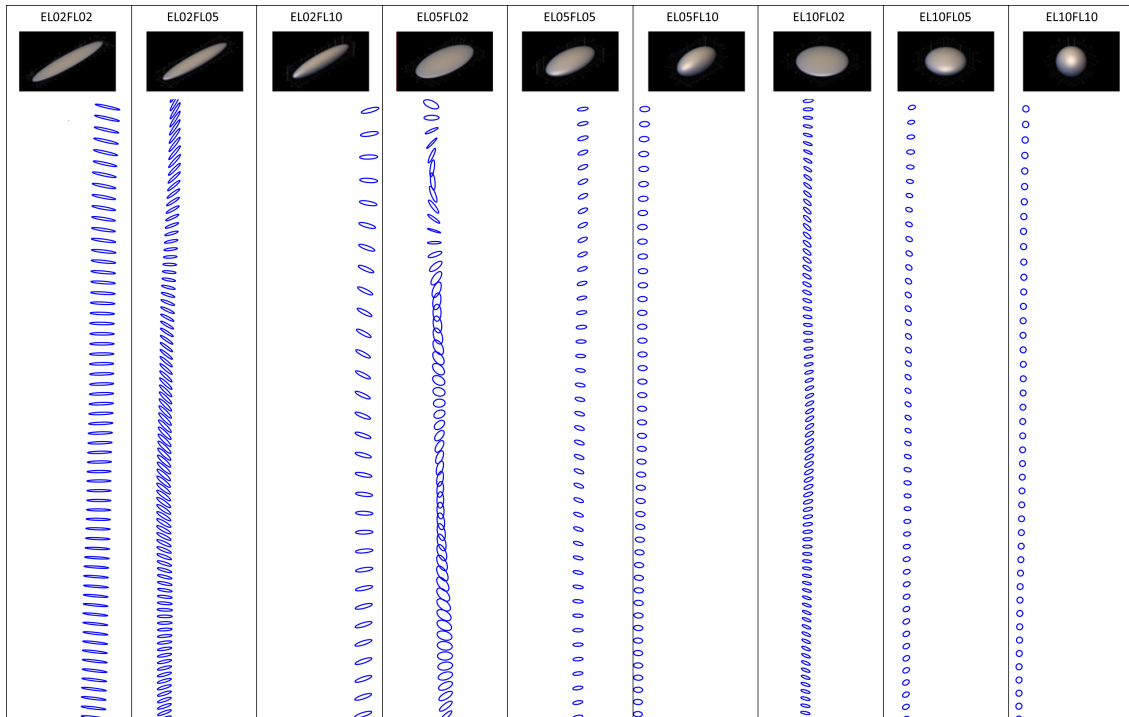


Figure 5.3: Example ellipsoids as recorded 20 mm after insertion - top camera view.

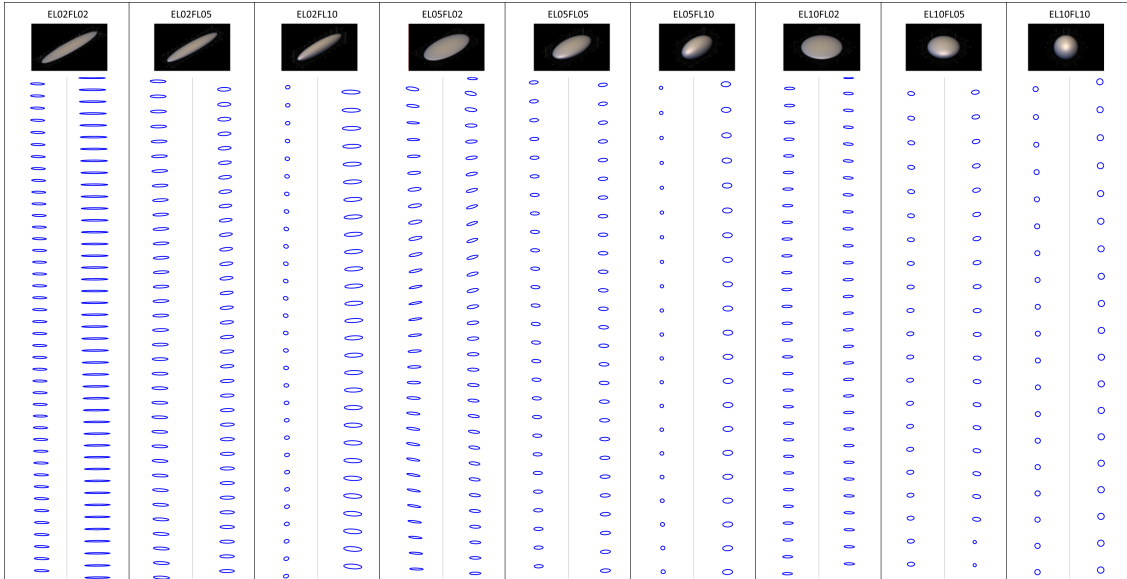


Figure 5.4: Example ellipsoids as recorded 70 mm after insertion - views from the two bottom cameras. The cameras are aligned 90° to each other so that the particle is seen from the front and from one side view. At this point, most particles settle with their largest projection area normal to the falling direction with some remaining oscillation.

In order to verify the tendency of the particles to fall with their largest projection area first, and to quantify the change in rotation rate, the change in angle between frames was calculated for all videos of all shapes. The results were then filtered for any 180° jumps that would occur. When tracking the angles of the ellipsoids in free-fall, it became apparent that distance, that the particles were tracked, on average showed only one period of oscillation before the particle left the field-of-view. Therefore, the reduction in rotation rate could not be linked easily to the particles assuming a steady-state orientation. However, comparing the videos 20 mm after the insertion point with the videos 70 mm after insertion point shows a reduction in the rotation rate that is consistent across all shapes.

Figures 5.5, 5.6 and 5.7 show the recorded rotation rates across all videos, sorted by shape and grouped by elongation. Specifically, they show the angle Θ_p , fixed to the longest axis of the ellipses matched to the particle shape seen in the image plane, as a function of time. Please note that EL 1.0, FL 1.0, the sphere, is not shown, as no rotation rate can be extracted from a sphere with the given shadowgraphy setup. Further note that for the top camera of the shape EL 0.5, FL 0.5, the shape was tumbling at a rate too high for the frame rate and is therefore undersampled. This undersampling leads to an overestimation of the rotation rate of the ellipsoids in the top camera. When encountering the bottom cameras, the rotation rate had slowed down and provided reasonable results again.

The results show a clearly discernible dampening taking place between the top and bottom observation heights. In the majority of cases, the rate of rotation decreased from around $2 - 10^\circ/\text{ms}$ to small single digit numbers of $1-2^\circ/\text{ms}$. Together with the visual confirmation of particles assuming an orientation so that the largest projection area was normal to the falling direction, the reduction in rotation rate indicates that the particles would then also remain in this orientation.

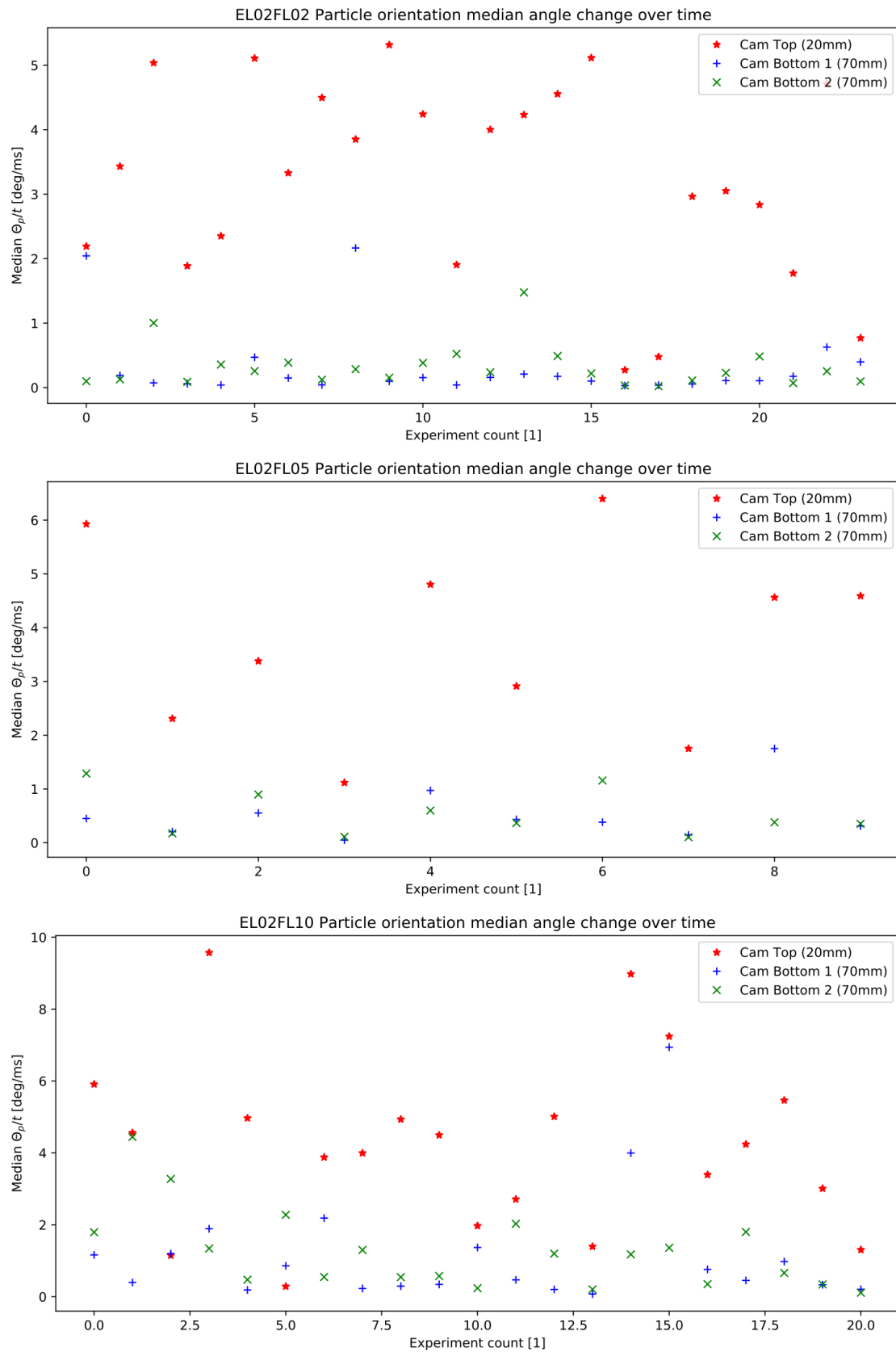


Figure 5.5: Particle rotation rate as a function of median change of particle orientation angle Θ_p over time for different experiments with shape EL 0.2, data set 1. Every experiment count represents a single experiment where a particle is being observed in free-fall within the drop tower.

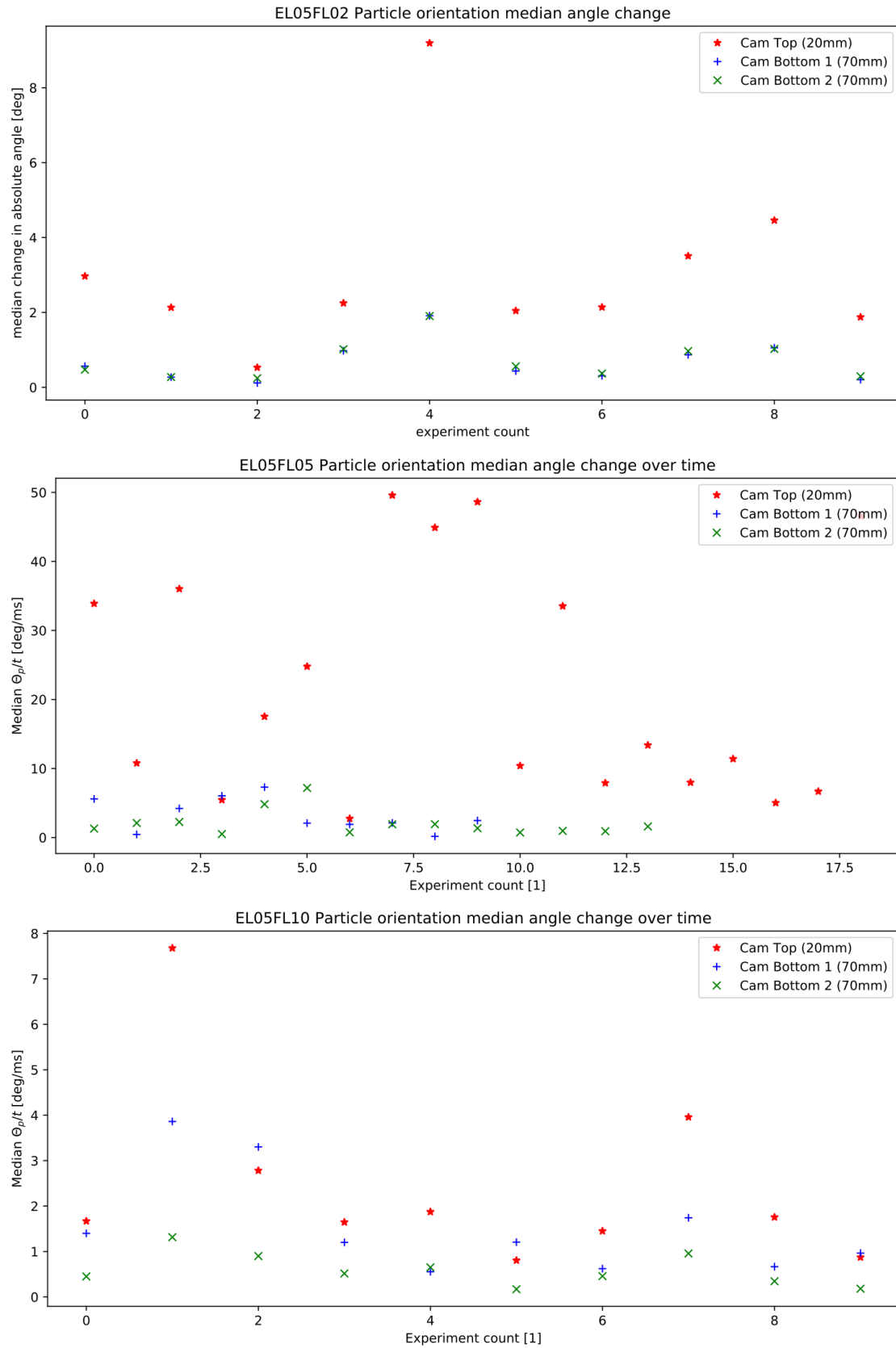


Figure 5.6: Particle rotation rate as a function of median change of particle orientation angle Θ_p over time for different experiments with shape EL 0.5, data set 1. Note that for EL 0.5, FL 0.5 the top camera was undersampling the tumbling particles and wrongly estimates the rate of rotation. The bottom cameras aren't affected by this issue.

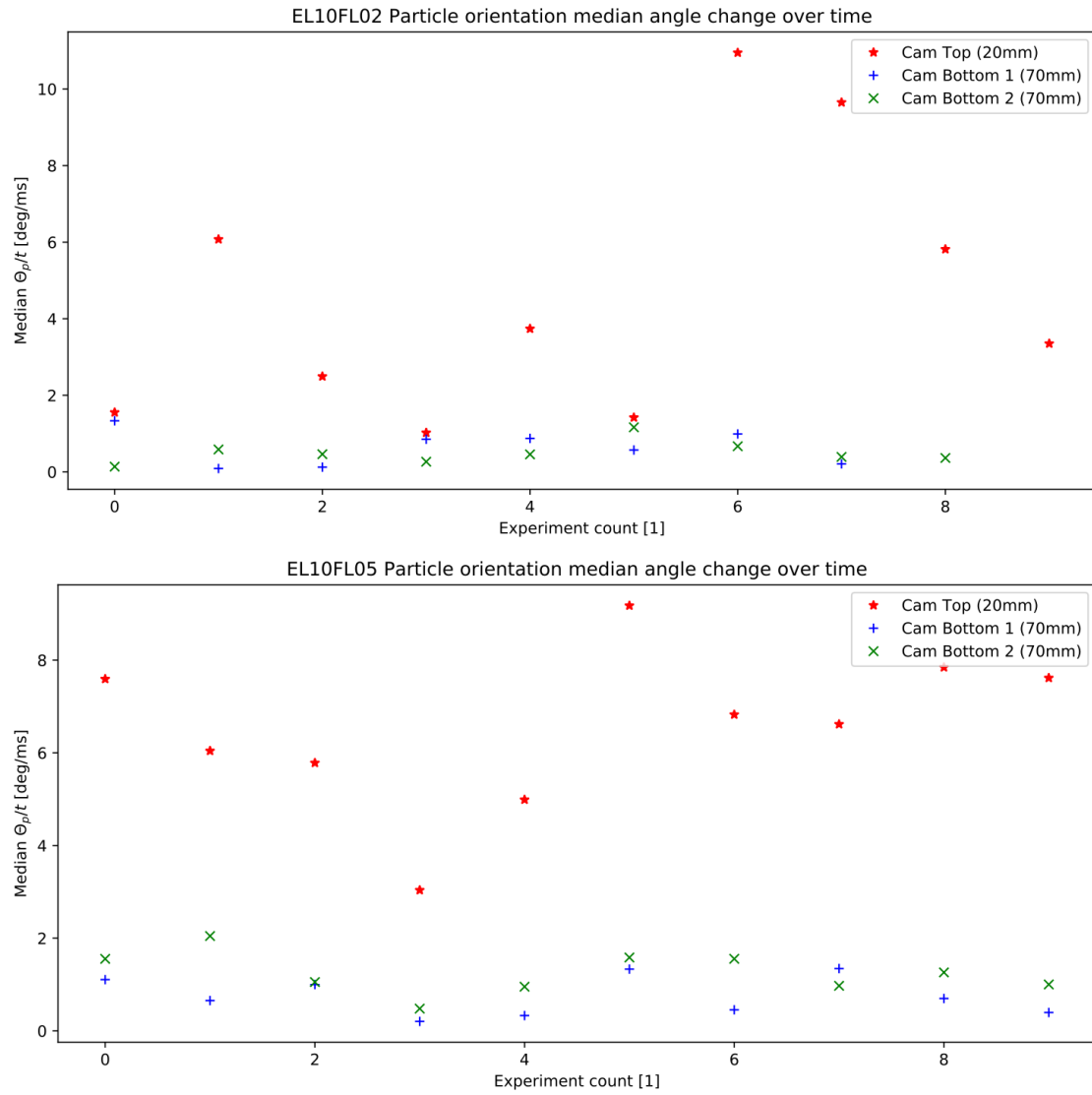


Figure 5.7: Particle rotation rate as a function of median change of particle orientation angle Θ_p over time for different experiments with shape EL 1.0, data set 1. Every experiment count represents a single experiment where a particle is being observed in free-fall within the drop tower.

5.7 Conclusions

Tracking the particle flight with a single camera view each revealed information about the particle behavior during free-fall. Across all shapes, the particles tended to settle with their largest projection area first and appeared to tend to a steady-state orientation, even if some particles were still oscillating when they reached the end of the lower observation volume. The terminal velocities recorded agreed well with the model proposed by Bagheri and Bonadonna [2016] which implies that the corrected drag model is applicable in the intermediate regime, located between Stokes' and Newton's regimes.

In terms of the particle rotation rates, some dampening was observed, when taking into account the difference in rotation rates between the positions observed by the top and bottom cameras. However, the time during which the particles were recorded in the cameras was too short to track the particle oscillation and thus the damping of the oscillations for more than about one oscillation period. This implicated a need for a larger field of view (FOV) to track the particle rotations. The particle trajectories were only observed from one view each without reconstruction of a full 3D tracking, and therefore revealed only part of the path the particle took while falling.

Another downside of the single-camera tracking was that all tracking of the orientation of the particles was heavily dependent on the ellipse fit to the particles found in the individual frames. Therefore, blurriness or glare points could potentially lead to under-/oversizing the particles or recording their angles incorrectly. In the case of such an incorrect recognition, it could happen that the particle rotation rate was overestimated due to changes in the quality of the fitted ellipses from one frame to the other. Finally, all tracking of the orientation of the particles was of the *extrinsic* type, meaning that the particle orientation was tracked in reference to an exterior coordinate system. A more ideal solution would be to track the particle orientation *intrinsically*, as this would describe the rotation of the particle by referring to its internal axes. This would correspond to the common maneuvers of roll, pitch, and yaw that are often used in aerospace applications.

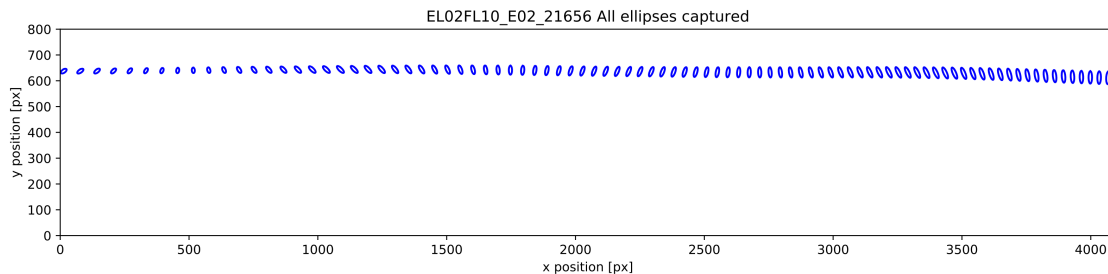


Figure 5.8: Particle track of shape EL 0.2, FL 1.0, experiment 2. Plotted is a composite image of all detected particle contours for the flight of the particle through the cameras' field-of-view. Oscillations rotations of the particle are easily visible.

In the following chapter 6, a solution to many of the problems mentioned will be discussed. Another set of experiments was performed, this time focusing on the transient dynamics and the dampening of the particle rotations shortly after the insertion. In this case, two cameras were placed on top, 90° apart, and one camera on the bottom. From now on, all cameras that were used were of type Phantom VEO 4K 990 L, from company Vision Research. As a further step, the macro-lenses that were used up until this point were removed and therefore the magnification of the camera setup reduced. The cameras were then used directly with 200 mm lenses. On the flip-side, this greatly increased the FOV, enabling tracking of the particles for roughly 3 times further in terms of vertical length. The data set is detailed in chapter 6.8.1, though it should be mentioned here already that the heights of the cameras were adjusted as follows:

- The two top cameras, 90° apart, were recording the falling particles from 20 mm below the insertion point onward for a distance of 27.5 mm. The cameras were therefore monitoring the distances 20 mm to 47.5 mm after the insertion point.
- The one bottom camera was recording the falling particles from 43.9 mm below the insertion point onward for a distance of 27.7 mm. The camera was therefore monitoring the distances 43.9 mm to 71.6 mm after the insertion point. This created an overlapping field of view with the top cameras for 3.9 mm between 43.9 mm after the insertion up to 47.8 mm after the insertion.

Before embarking on the creation of a more complex code, the data set was tested with the single-camera tracking solution from this chapter. Already, the increased FOV was revealing more of the particle dynamics, often displaying dampened sine waves such as the ones shown in figure 5.9, with one of the ellipsoidal trajectories and orientations displayed as an example in figure 5.8. These results were promising, even with their limitation of only tracking the particles extrinsically.

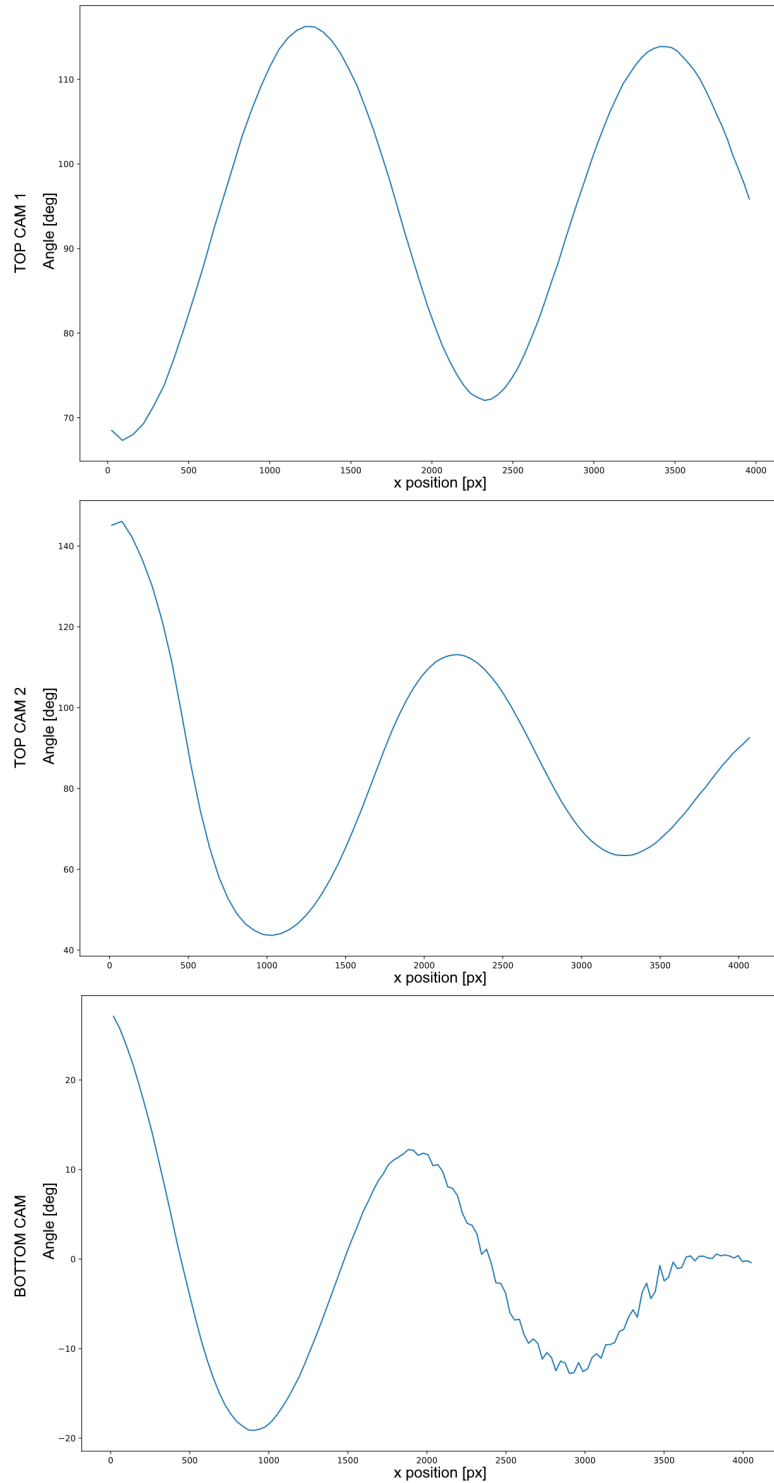


Figure 5.9: Recorded angles of experiment 2 for shape EL 0.2, FL 1.0 as an example for a typical result of the single camera tracking code. **Top and middle:** the two cameras that were recording on the top of the observation volume. **Bottom:** the single camera that was placed lower.

Compared to the rotation rates that were covered in chapter 5.6.3, the rotation rates shown in figures 5.10, figure 5.11 and figure 5.12 are generally higher due to the different observation heights. Particles were routinely still displaying significant rotation rates and on-going dampening when comparing their rotation rates between the top and bottom cameras. Here, the measured rotation

rate at the top cameras was typically significantly higher than the measured rotation rate observed at the bottom camera.

With the change in FOV, about 3 oscillation periods could be monitored on average. The particles were not visible when they reached their terminal velocities or steady-state orientations in this data set due to the changed observation height. Instead, the particles were monitored close to the top of the observation volume when they were still tumbling in order to catch the moment in time during which they experienced dampening. The figures show the median change of angle and, hence, the effect of rotation dampening is only apparent when comparing between the top cameras and the bottom camera. What can be seen in figures 5.10-5.12, however, is the higher median rotation rate found in the top cameras compared to the rotation rates seen in figures 5.5-5.7, indicating the feasibility of measuring rotation dampening in the chosen observation heights.

Typically, the median rotation rates of the two top cameras were faster compared to the rotation rate observed in the bottom camera. Notably, the median rotation rates were slowest for shape EL 0.2, FL 0.2, likely due to its long and flat shape counteracting changes in its orientation. The same elongation EL 0.2, however, displayed faster rotation rates, compared to EL 0.2, FL 0.2 for flatness FL 0.5 and FL 1.0. The fastest rotation rates were seen for shapes with elongation EL 1.0, independent of their flatness FL.

Apart from the typical case, where the two top cameras showed faster rotation rates compared to the bottom camera, the following exceptions were recorded: For EL 0.2, FL 0.2, experiment 5, the rotation rates were roughly the same between top and bottom cameras, showing only very slow rotation. In three cases, the bottom camera displayed faster rotation rates than the two top cameras, cases [EL0.2,FL0.2 exp. 0], [EL0.5,FL1.0 exp. 3], and [EL1.0,FL0.5 exp. 3]. The reason for this apparent acceleration is of yet unknown. Finally, in eight cases, the rotation rate of the bottom camera was found to be between the values of the two top cameras, indicating that particle was primarily rotating about an axis normal to the field-of-view (FOV) of one of the two top cameras. This also highlights the need for a total of four cameras (two top, two bottom) in future experiments. Notably, half of these occurrences were recorded for shape EL 1.0, FL 0.5 with other shapes displaying no more than one such occurrence.

As the preliminary analysis of the new data set exhibited faster rotation rates and apparent dampening between the top and bottom cameras, it was then used as a test case for the multi-camera tracking code that will be discussed in the the following chapter 6. Already, however, with the simple single camera tracking it was verified that the larger FOV would enable better results due to a longer tracking of the particles while still maintaining a resolution of ca. $6.75 \mu\text{m}/\text{px}$, high enough for orientation determination of particles with the chosen size.

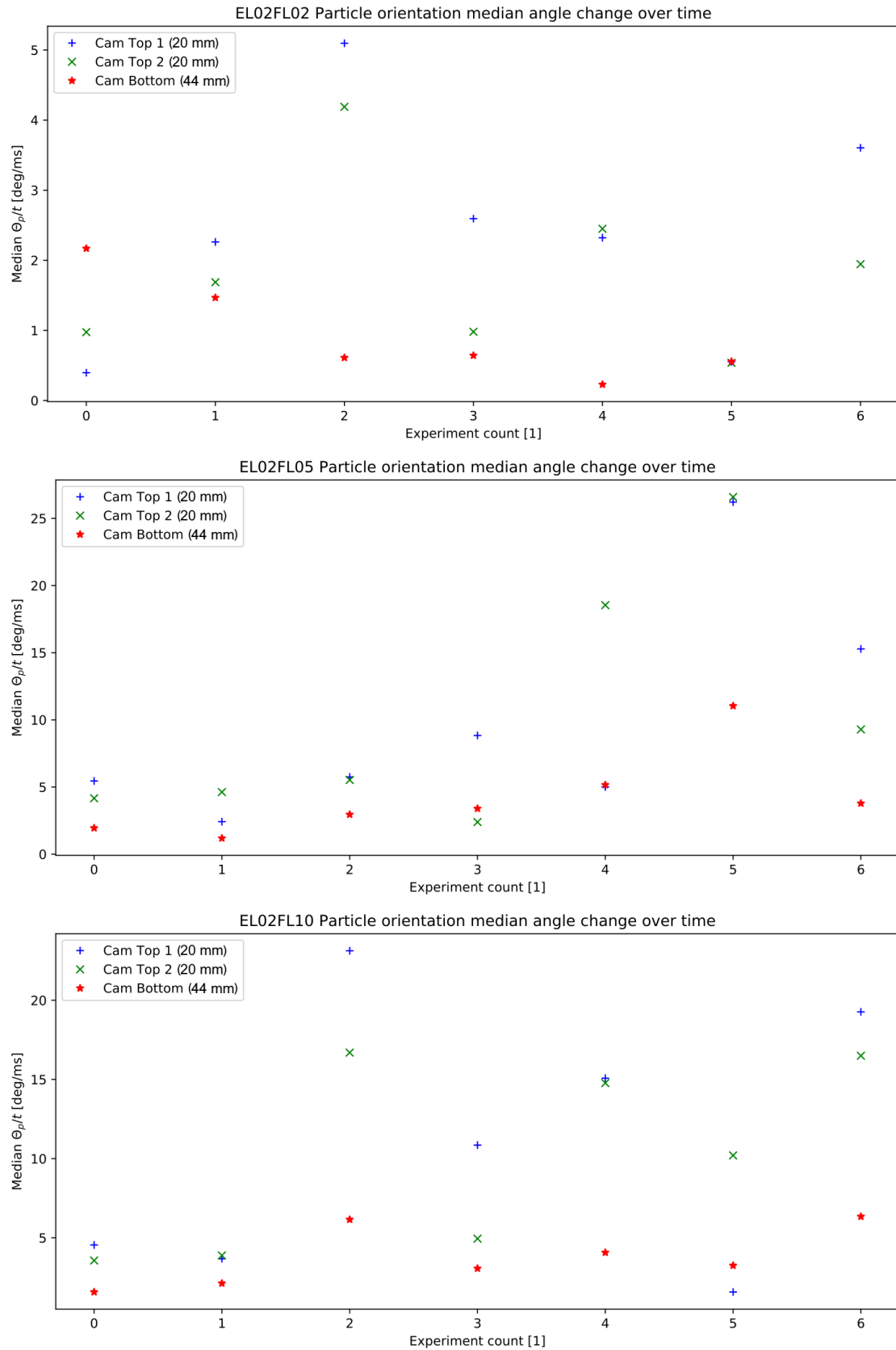


Figure 5.10: Particle rotation rate as a function of median change of particle orientation angle Θ_p over time for different experiments with shape EL 0.2, data set 2. Every experiment count represents a single experiment where a particle is being observed in free-fall within the drop tower. From top to bottom: shapes (EL 0.2 FL 0.2), (EL 0.2 FL 0.5), and (EL 0.2 FL 1.0)

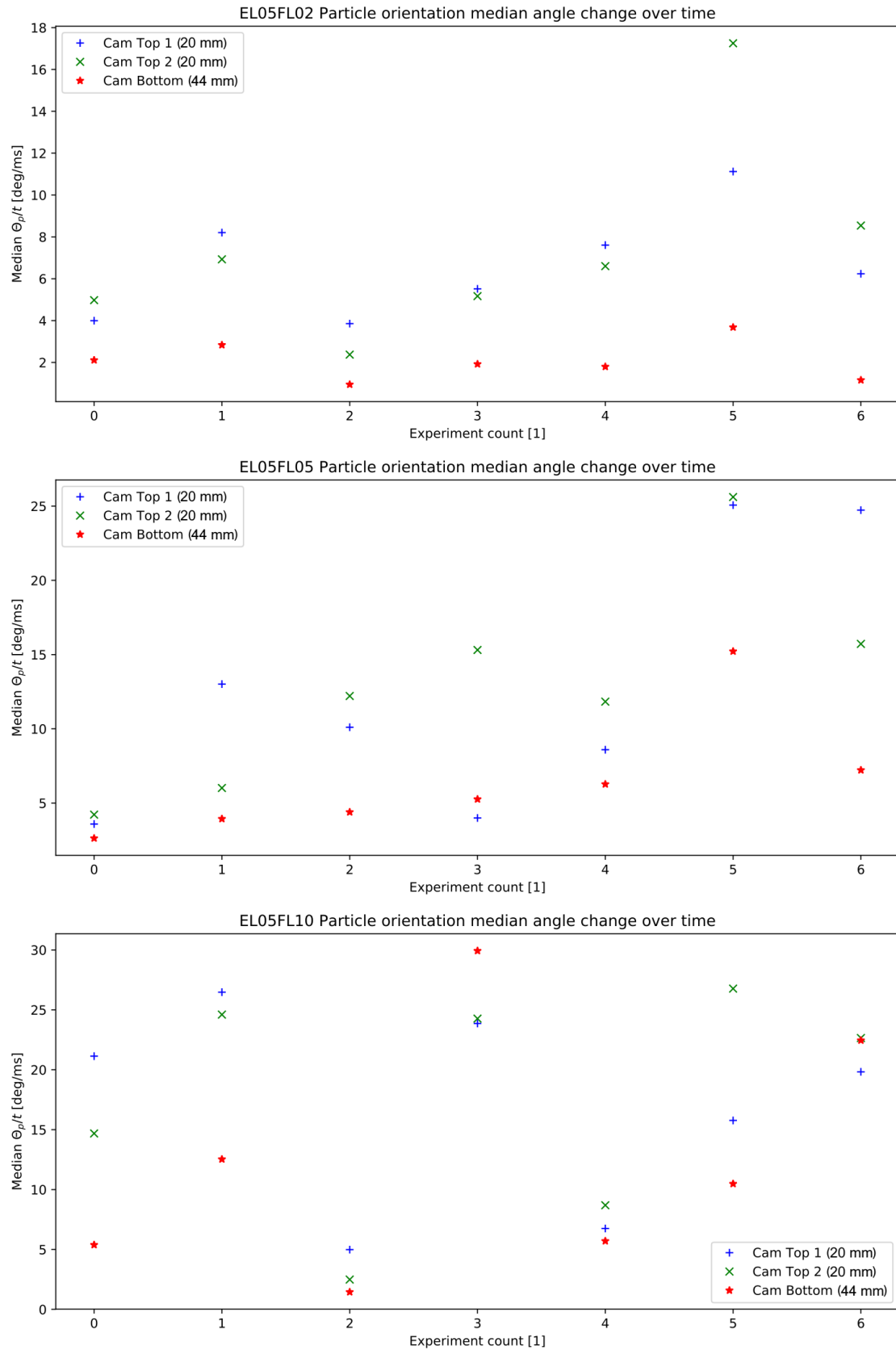


Figure 5.11: Particle rotation rate as a function of median change of particle orientation angle Θ_p over time for different experiments with shape EL 0.5, data set 2. Every experiment count represents a single experiment where a particle is being observed in free-fall within the drop tower. From top to bottom: shapes (EL 0.5 FL 0.2), (EL 0.5 FL 0.5), and (EL 0.5 FL 1.0)

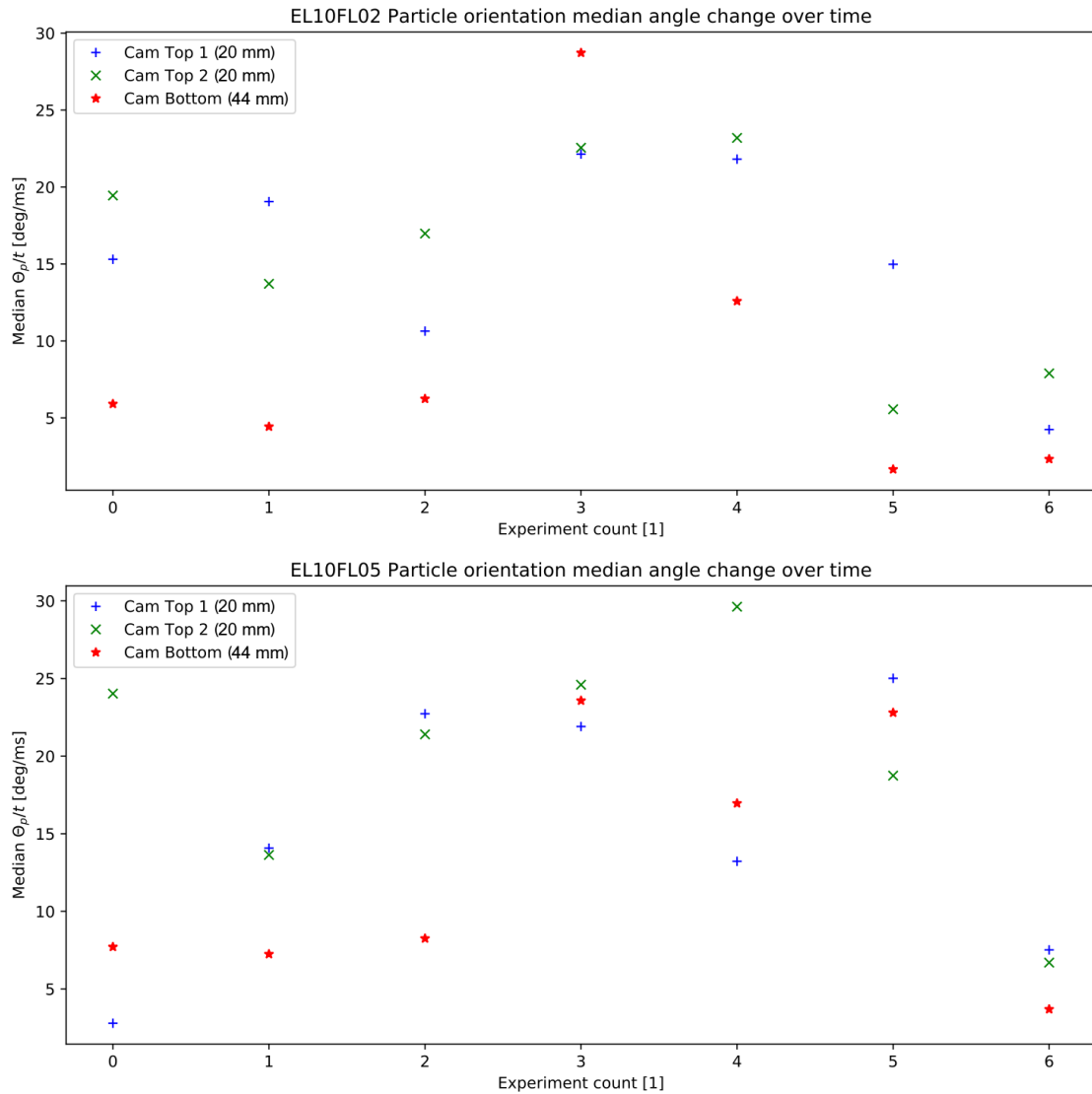


Figure 5.12: Particle rotation rate as a function of median change of particle orientation angle Θ_p over time for different experiments with shape EL 1.0, data set 2. Every experiment count represents a single experiment where a particle is being observed in free-fall within the drop tower. From top to bottom: shapes (EL 1.0 FL 0.2), and (EL 1.0 FL 0.5)

Chapter 6

Free-fall dynamics of highly inertial ellipsoids: Multi-camera tracking

6.1 Introduction

Building onto the ideas of the single-camera tracking code from chapter 5, the multi-camera tracking code considers both camera views as a combined set of frames. It extracts the particle trajectory and orientation during all of the observed free flight in full 3D. Unlike the single-camera tracking code, all parts were developed in Python while using the packages already covered in chapter 5.2, as well as additional packages that will be introduced in the relevant steps. The approach is similar to other works, such as Will et al. [2020]. The main idea is to generate projection images from a 3D model for both camera views, then compare 3D model projection images to the camera images and minimize the difference between the two sets. A reference library is created and an optimization scheme is utilized to minimize a cost function. Cost functions will be covered in chapter 6.6.3. For now, it is sufficient to know that they are used to represent the difference between the orientation of the real ellipsoid and the 3D model.

Figure 6.1 shows an overview of the code. Beginning with a set of camera frames, several steps are performed: Information from the calibration step is read in and parameters of the recorded ellipsoids are extracted. After choosing a thresholding method, the shift between the two cameras is estimated. Then, the trajectory of the particle is calculated and plotted. Further steps are performed in preparation for the orientation determination, such as reading in a 3D model of the ellipsoid and generating a reference library of rotations. After choosing to analyze either a synthetic test or actual camera images, the orientation estimation is started. The reference library is used as a starting point and the library image with the smallest cost relative to the first set of camera frames is picked. The 3D model is rotated into the position of the reference image and projection images are created. The cost is then further minimized with a Nelder-Mead solver [Nelder and Mead, 1965]. Resulting angles are saved as the new initial guess for the following sets of camera frames. As soon as two sets of frames have been analyzed, the rotation rate of the ellipsoid is calculated and serves as the input initial guesses for later sets. Should the camera frames contain ellipsoids whose projection ellipse exhibits a high circularity, a mitigation strategy is performed. In this case, the Nelder-Mead method is replaced by a bounded L-BFGS-B solver¹ [Zhu et al., 1997] that can move the 3D model of the ellipsoid only in tight margins around the initial guess from the rotation rate estimation. Finally, the results are stored as plots and in a json file.

¹A limited memory ("L") and bounded ("B"), "BFGS" algorithm, after its creators: Broyden, Fletcher, Goldfarb and Shanno, as detailed in [Nocedal and Wright, 2006]

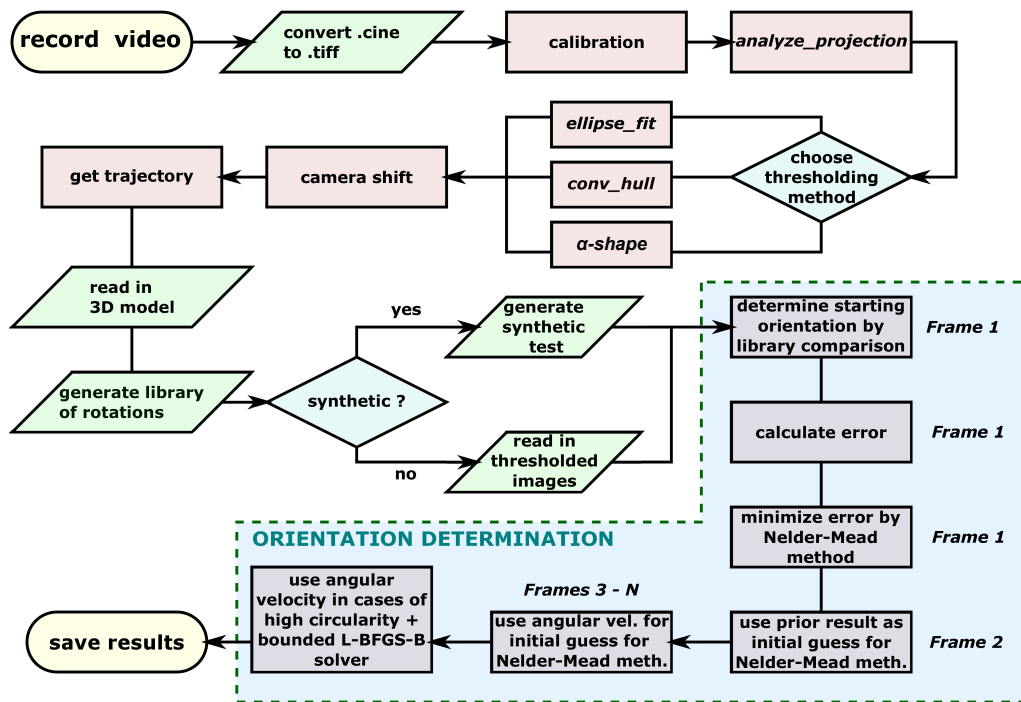


Figure 6.1: Flow chart of the multi-camera tracking code. Several steps are performed in a sequential order, beginning with reading in the recorded video, extracting ellipse parameters from it and generating thresholded images. The 3D model of the ellipsoid in question is read in and a reference library created. After choosing either a synthetic or a real test case, the orientation determination is performed with a mixture of utilizing the reference library, as well as the Nelder-Mead and L-BFGS-B solvers. Finally, the results are stored as plots and in a json file.

6.2 Analyzing frames: extraction of ellipsoid parameters

As a first step, the camera frames are analyzed by a subroutine that extracts various parameters of the ellipsoids and saves this information in a dictionary. This part of the code also reads in the calibration data that was obtained in a prior step, as detailed in chapter 4.5, and stores it in the same dictionary. After storing some additional information in the dictionary, the main routine is started. Here, each frame is individually processed by the function `analyze_projection`. The function receives the file path of the current frame, reads in the frame as an 8-bit image and thresholds the image. It extracts the contour with the maximum area and analyses its features using OpenCV functions.

Three different methods are available for the threshold:

1. `cv2.threshold` : a constant global threshold value is applied to the image.
2. `cv2.adaptive_thresh_mean_c` : an adaptive threshold method is used, where the threshold value is the mean of the neighboring area.
3. `cv2.adaptive_thresh_gaussian_c` : also an adaptive threshold method, but in this case the threshold value is the weighted sum of neighboring values where the weights are a Gaussian window.

Depending on the threshold method which was chosen in the prior step, the threshold is applied differently. For option one, the threshold is the constant global threshold. For option two, the threshold is a constant that is deducted from the weighted mean. For option three, the threshold

is the multiplier of the Gaussian kernel size.

The function can be used with dark features on a bright background or vice versa. It can optionally be informed about the scaling of the image but analyzes the features using pixel values by default. It should be noted that this function is meant to deal with single features and not with camera frames that hold several or many features. For example, an image displaying a lot of particles would cause the function to only consider one of all these particles.

Within the `analyze_projection` function, two more sub-functions are being called: `find_max_contour` and `process_contour`.

Find_max_contour

This sub-function calls the OpenCV function `cv2.contourArea`. It computes a contour area using the Green formula [Green, 1828]. The sub-function runs `cv2.contourArea` for all contours that it found and picks the maximum area contour which it then returns.

Process_contour

Processes a list of provided points to characterize the dimensions and shape features that it finds within these points. It then returns them as a dictionary. Input parameters are: the list of boundary points, the image scale, and a statement if the contour is closed or open. A large number of ellipse parameters is returned such as: area, centre point, circularity, major and minor axis, bounding rectangle (and enclosing rectangle), maximum and minimum ferets (similarly to using the caliper measurement tool), and the aspect ratio between the axes.

6.3 Pre-processing of data: estimating camera temporal offset

While both of the top cameras were recording images in sync with each other, the trigger signal was not synchronized. Therefore, one of the cameras could receive the trigger signal earlier than the other and start recording a couple of frames earlier. This temporal offset was a constant and unpredictable number of frames for each individual video. The temporal offset was estimated by minimising the mean difference in the particle vertical position (in sensor coordinates) between the two cameras, over all frames for which it was visible on both.

6.3.1 Line-of-sight projection

For verification of the temporal offset obtained above, a small tool was created to visually compare the two views of the cameras. Specifically, it would display the view of camera one, overlaid with the assumed particle position from camera two according to the frame shift. If the frame shift was estimated correctly, the actual position of the particle from camera one should lie inside the line-of-sight (LOS) projection from camera two.

6.4 3D-tracking: trajectory of the particle flight

Building on the prior work of analyzing the frames (chapter 6.2) and estimating the shift of the cameras (chapter 6.3), the 3D-tracking of the particle trajectory was now straightforward: With the Z (along falling direction), X and Y (lateral movement) positions of the particle and the temporal offset, all necessary components were given. The coordinates that were chosen for the

trajectory were the Z position from camera one (z_{cam1}), X position from camera one (x_{cam1}), and the Y position from camera two (y_{cam2}). Figure 6.2 shows the trajectory extracted from an experiment with particle shape EL 0.5, FL 0.2 as an example. The figure shows the real world coordinate system where movement in the XY-plane corresponds to lateral movement and the Z axis is aligned with the falling direction.

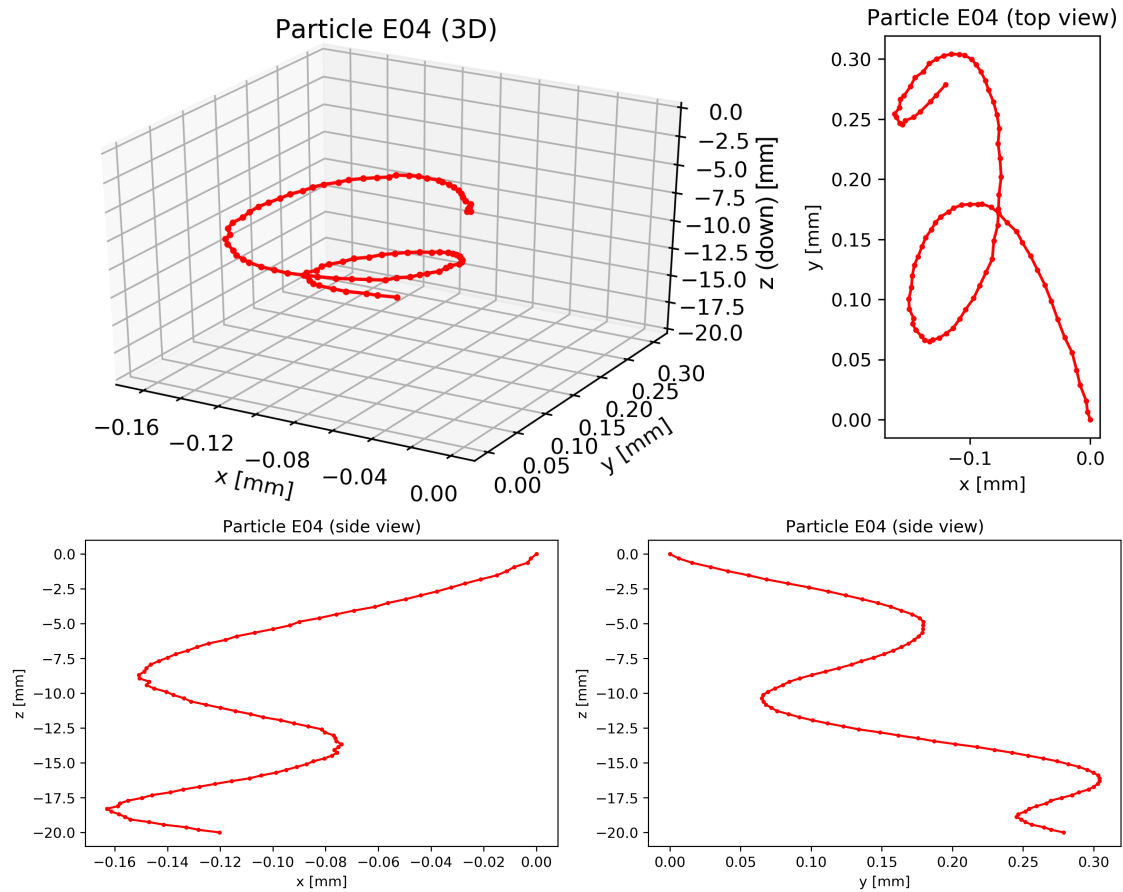


Figure 6.2: Trajectory extracted from particle with shape EL 0.5, FL 0.2. **Top left:** 3D visualization of the particle trajectory during its free-fall through the settlement chamber. The particle performs a light circling motion. However, the absolute amount of lateral movement is very small compared to the falling distance (note that the axes have not the same scaling). **Top right, bottom left and right:** top and side views of the particle's trajectory.

6.5 Pre-processing of data: extraction of camera frames and thresholding

In order to prepare the camera frames for the orientation estimation covered in chapter 6.6, a threshold was applied to the particle images and they were reduced to black and white images that contained only the particle itself while the empty space around it was cropped. Reducing the images to black-and-white (bw) causes pixelation, also known as aliasing. To counteract this, the images had to be upscaled prior to the thresholding and reduction to bw images. After the upscaling, the particles could be processed in one out of three ways:

1. Ellipse fit
2. Convex hull

3. Alpha shape

These methods will be covered in their respective subsections.

6.5.1 Cropping

Cropping of the images was done to reduce overhead and speed up later calculations. The cropping was centred around the X and Y pixel positions of the ellipsoids that had been identified by the analyzing frames function, covered in chapter 6.2. The maximum dimension of each particle size has to be entered to avoid cropping parts of the particle. However, since the particles were produced by artificial means the maximum dimensions were prior knowledge. Next, the minimum size ℓ_{\min} that needed to be cleared for the particle was calculated as follows:

$$\ell_{\min} = \frac{d_{\max}}{\mathcal{M}} \quad (6.1)$$

where d_{\max} is the ellipsoid's longest dimension (for example: $d_{\max} = 700 \mu\text{m}$ for an elongated and flat ellipsoid of shape EL0.2 FL0.2) and \mathcal{M} is the camera resolution, for example $\mathcal{M} = 6.75 \mu\text{m}$ per pixel in the case of the used VEO cameras together with a 200 mm objective. To make sure that the particle edges would not clip into the border of the image, some padding was introduced through multiplication by a pre-factor of 1.2. The cropped images were then square with a side length of $1.2\ell_{\min}$ pixels.

6.5.2 Calculating particle contours

Calculating the particle contours was done with a specific function called `calc_ellipse`. While the function would take into account the original centre points of the ellipses calculated by the `analyze_projection` function, the way the ellipse contour is calculated here differs from the maximum area approach used in `analyze_projection`. It follows the following procedure:

1. The image is cropped so that only the part containing the particle is being analyzed.
2. The image is optionally smoothed with a Gaussian filter to reduce noise
3. The cropped image is normalized by the maximum intensity value.
4. The edge points of the contour are found by checking the image intensity of the particle.

In the last step, the cropped image is being split into horizontal and vertical lines. Here, a horizontal line would correspond to the code scanning x pixels at a fixed y value and a vertical line (a column) would correspond to the code scanning y pixels at a fixed x value. Then, a 2D-polynomial is fit wherever the normalized image intensity is equal to:

$$I_{\min} + q_{\text{thr}} (I_{\max} - I_{\min}) \quad (6.2)$$

where q_{thr} is the normalized intensity threshold, a user defined parameter. When going through the different lines, the function takes into account the amount of neighbouring pixels for the fitting of the polynomial. The number of neighbours that the function considers is given by the `n_neighbours` parameter and is also a user defined parameter. How both parameters affect the final black and white images will be covered at the end of this section. For the actual fitting of the polynomial, the SciPy function `curve_fit` is being used. `curve_fit` uses non-linear least squares to fit a function to data. The method chosen for the least squares was the `dogbox` method. The `dogbox` method is based on the `dogleg` algorithm, an iterative optimisation algorithm created in 1970 by Michael J. D. Powell [Nocedal and Wright, 2006]. It is typically used for bounded problems. An example of an intensity profile obtained from an ellipsoid image is shown in figure 6.3.

Several parameters are being returned that are used in the following functions. They are: optimized results for the centre point of the ellipse, the ellipse major and minor axis lengths, the X and

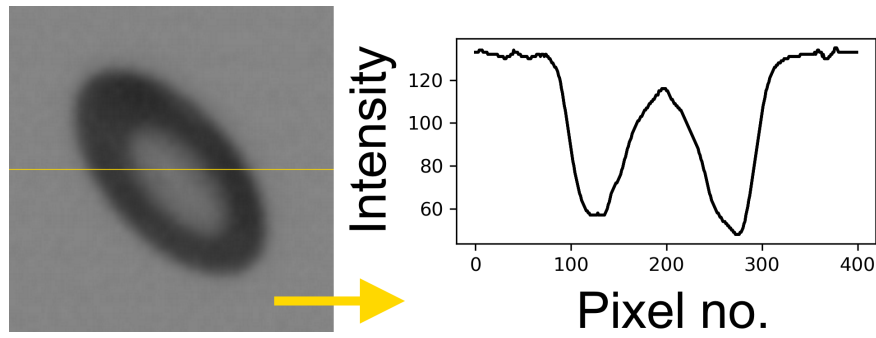


Figure 6.3: Intensity profile of a single line of a camera frame containing an ellipsoid of shape EL 0.5, FL 0.5. To estimate the position of the contour points defining the ellipsoid, the image is split into lines and rows. An intensity profile of such a line is shown on the right. Clearly visible are two dips at which the intensity dips as low as 50 on a scale going from zero to 255. The two dips correspond to the dark areas that make up the particle edges. The code will then take the slope of the intensity profile into account to estimate the position of the contour.

Y coordinates of the contour points found by the polynomial fit, and coordinates for an optimized ellipse fit. These results are a closer representation of the true ellipsoid contours compared to the initial results found by `analyze_frames`.

In the next step, the contour that was found is transformed into a filled, black-and-white image, using one of the following three methods: Ellipse fit, convex hull, or alpha shape. The different outcomes of these methods are shown in figure 6.4 and discussed in the following.

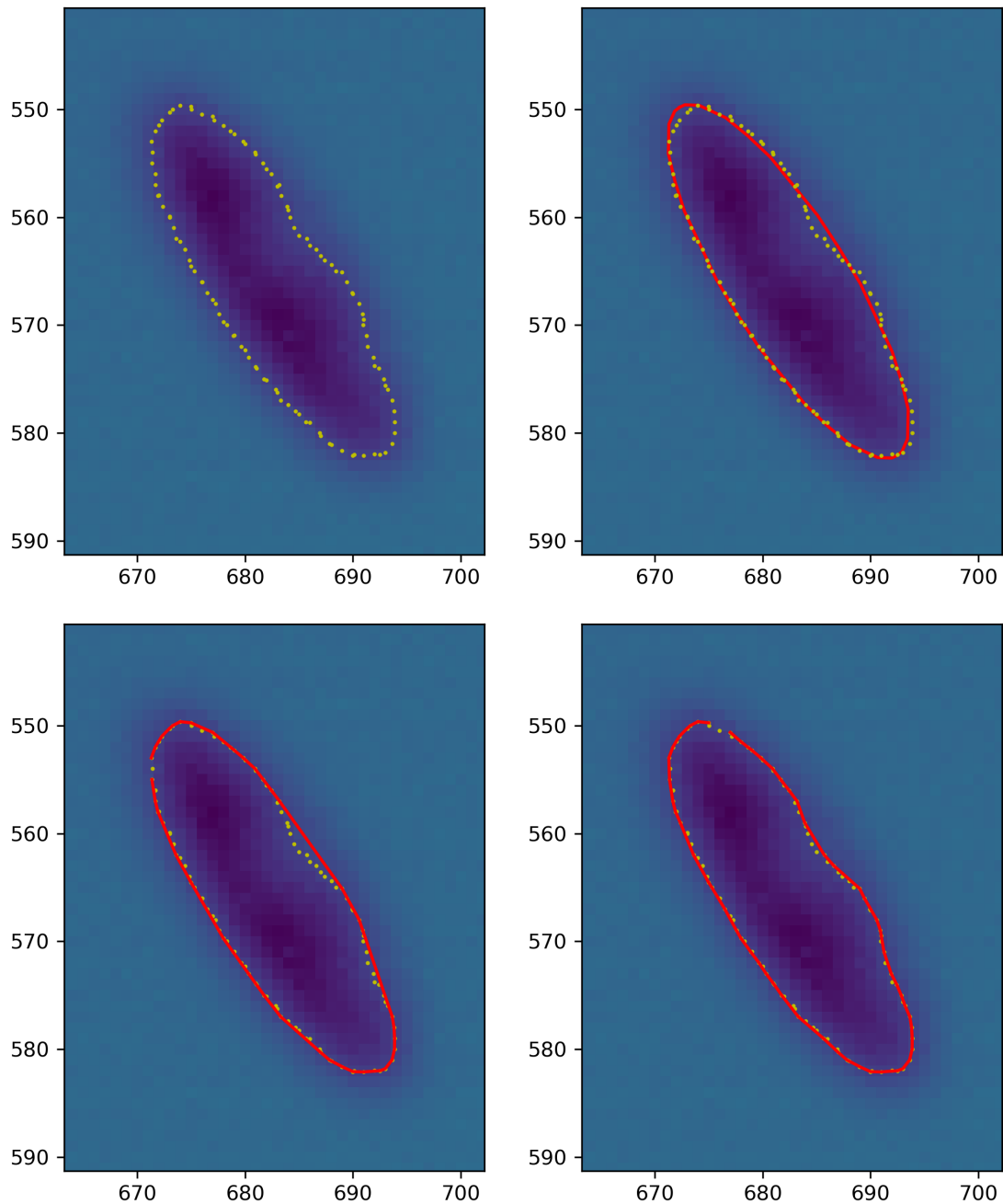


Figure 6.4: Effect of different particle contour fitting options. **Top left:** No fitting. A camera image of a partly out-of-focus ellipsoidal particle (EL 0.5, FL 0.5) is visible in the background. The yellow points mark the contour points that have been detected by the `calc_ellipse` function that takes the intensity gradient of the image into account to detect the ellipsoid’s contours. **Top right:** Ellipse fit to the detected contour points. The ellipse does not follow the contour points exactly but attempts a best fit to the given points. Fitted ellipses can extend further than the contour points, as visible on the top of the particle, or bridge dents such as seen on the right side of the particle. **Bottom left:** Convex hull fitted to the contour points. A convex hull follows the outer contour points closely, similarly to how a rubber band would. Dents aren’t represented with this fitting method. **Bottom right:** Alpha shape fitting follows the contour points more or less tightly, depending on the chosen alpha parameter. At $\alpha = 0$, the fitted shape is a convex hull. With increasing alpha, the shape is fitted increasingly tighter, resulting in a concave hull at $\alpha = 2$. The value chosen in the image was $\alpha = 0.10$ at which point the shape follows the contour points closely. **All:** The axes correspond to the position in pixel values where the particle was located in the camera frame.

6.5.3 Method 1: Ellipse fit

This method uses the ellipse that was fitted by the `calc_ellipse` function. When calling `calc_ellipse`, the user can choose between four different options of fitting an ellipse to the contour points that have been found by the polynomial fit. Options one and two are based on Fitzgibbon et al. [1996]. Option three uses the SciKit library’s ellipse fitting function `EllipseModel` which implements work from Halir and Flusser [1998]. `EllipseModel` performs a least squares minimization and claims to guarantee an ellipse-specific solution even for scattered or noisy data. Option four initially starts by performing option one and then additionally runs the results through a minimization scheme using `SciPy.optimize.minimize` using the conjugate gradient algorithm that is based on work by Polak and Ribiere [Nocedal and Wright, 2006]. This method is suitable if it is known that the shape in question is indeed an ellipsoid.

6.5.4 Method 2: Convex hull

Instead of using the ellipse points that are provided by `calc_ellipse`, it is possible to directly use the contour points that have been calculated. In this case, the contour points are used as the input for the Scipy function `ConvexHull`. `ConvexHull` is based on the Qhull library² which itself uses the O’Rourke algorithm [O’Rourke, 1994] to find the smallest convex set containing the provided points. This function has a tendency to over-size the particles and can lead to issues if there’s a small dust particle on top of the ellipsoidal particle. The method allows for the representation of non-ellipsoidal shapes as long as the particle in question doesn’t have any strongly concave surface features or holes. A common analog to explain a convex hull is to imagine a rubber band that is being wrapped around a set of points. The rubber band would then represent a convex hull.

6.5.5 Method 3: Alpha shape

An alpha shape is a generalization of the convex hull for a set of points in a plane. This generalization was first proposed by Edelsbrunner et al. [1983] and leads to a family of straight-line graphs, the so called alpha shapes or “ α shapes”. Unlike the convex hull, alpha shapes can be tuned by changing a parameter to fit more or less tightly to a set of points. In principle, it is even possible to represent holes in this set of points.

The alpha shape library used in this work was developed by Kenneth E. Bellock³. Setting its main parameter $\alpha = 0$ corresponds to fitting a convex hull around the provided set of points, just like in the prior method, while larger values of α create successively tighter fits around the set of points. Increasing the parameter further and further will lead to a concave hull fit at $\alpha = 2$ and eventually will result in placing points of the data set outside the generated shape.

The main advantage of the alpha shape is that the method is less prone to oversizing the particle and can follow more complex shapes than an ellipse or convex hull fit. On the counter side, oversizing the α parameter can lead to very thin and ragged particle representations that exhibit edges that aren’t present in the actual particles. Therefore, the parameter needs to be well chosen to obtain the desired results. To avoid choosing the parameter by oneself, the alpha shape library provides an automatic optimization module. This module tries to determine the alpha parameter that would best read the wrap around the set of points in one polygon before dropping any points. However, when testing this automatic optimization module, particles were routinely displaying holes in the resulting ellipse images.

6.5.6 Scaling and final touches

Prior to running through either of the three methods that were covered above, the contour points were upscaled by a scale factor of 8. After the thresholding, the images display steps the size of

²<http://www.qhull.org>

³<https://pypi.org/project/alphashape/>

pixels, also known as aliasing. Since the ellipse images gained from the cameras are later compared to images generated from a 3D model, they needed to have the exact same dimensions and had to be upscaled to match the 3D model images' size. This was mainly done to reduce the effect that small scale rotations of the 3D model of the particles could have on the cost calculation covered in chapter 6.6.3.

Next, a white image with the desired dimensions was created. Then the calculated contours were filled by using the OpenCV function `cv2.fillPoly` and inserted as black pixels into the centre of the white image. `cv2.fillPoly` can fill any monotonic polygon without self-intersections. Saving the image, the frame number would be retained so that the frame number order was kept. In the case that no ellipsoids were detected by `analyze_frames`, an empty white image would be created to keep frame numbers correct. Examples of fitting results from all three methods are shown in figure 6.5.

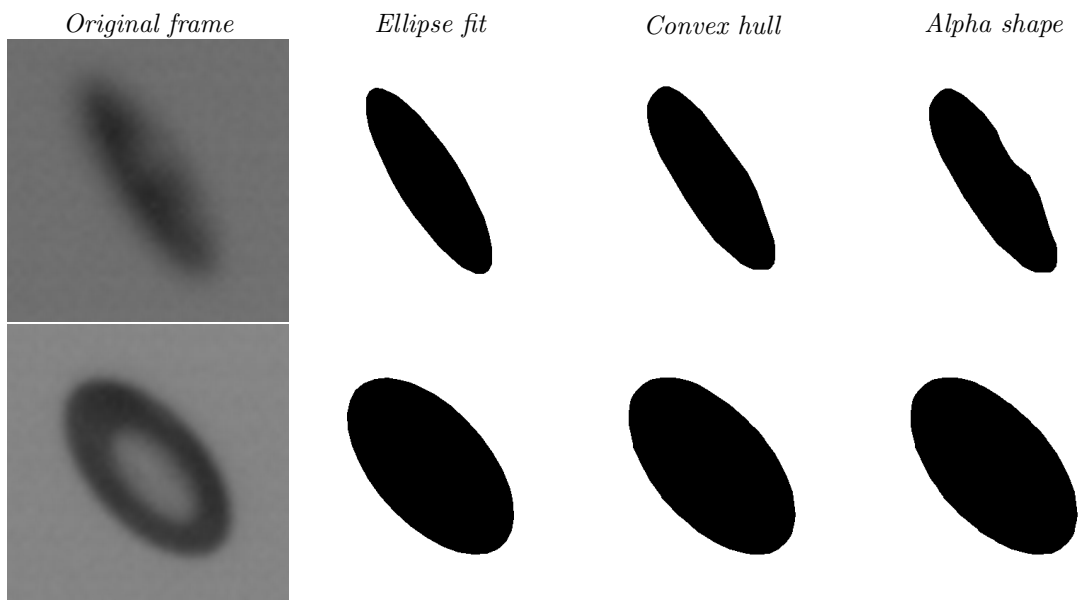


Figure 6.5: Results of the different contour fitting methods. **Columns from left to right:** Original frame, method 1: ellipse fit, method 2: convex hull, method 3: alpha shape. **Top row:** Particle of shape EL 0.5, FL 0.5 (bulky ellipsoid), frame 17, camera 1. **Bottom row:** Particle of shape EL 0.5, FL 0.5, frame 39, camera 2. Going from ellipse fit to alpha shape, a more tightly fitted contour is applied. Depending on the particle at hand, this might be more or less desired. The contour of an irregular particle would only be traceable with method 3: alpha shape, but since the particles in this case are ellipsoids, method 1 is providing the most agreeable results and follows the shape of an ellipse closely.

6.6 Orientation determination

Aside from the typical benefits that come from going into a laboratory compared to doing field experiments such as a controlled environment and repeatable experiments, the major benefit in this specific settling experiment was that the particle shapes were well defined and known a priori. A counter-example would be measuring atmospheric particles in-situ where the shapes and sizes are not known in advance. Therefore, the aim of this approach was to benefit from knowing the shape of the particles and reading in a 3D model of these particles to compare it against the measured results. This would then make determining the orientation of the particles easier in an in-situ case.

In brief, the orientation is determined through the following steps: first, a 3D model of the particle in question is read in and stored in the memory. The first frame of the thresholded images that are created with the methods shown in chapter 6.5.2, is then loaded and compared against a reference library to find a starting point for the orientation estimation. The comparison itself is done by calculating a cost (see chapter 6.6.3) that represents the difference between the camera and the 3D model images. Continuing from the starting point, the rate of rotation is tracked and provides new initial guesses for a solver, that would try to minimize the cost between each subsequent camera and 3-D model pair. A short overview of the general procedure is shown in figure 6.6. The details are discussed next.

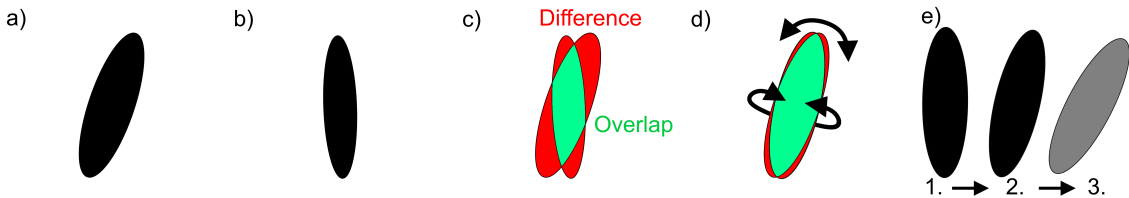


Figure 6.6: Orientation determination solver procedure. Several steps are performed: **a)** A thresholded camera image is loaded. **b)** The camera image is compared to a reference library and the reference image that results in the smallest difference is chosen. **c)** Determining the area of the difference is done by overlapping the two images, camera and reference. The red areas indicate the difference between the two ellipses and the green area indicates the overlap between them. **d)** Once the initial guess has been gained from the reference library, the solver is rotating the 3D model to find the best fit to the given camera image. **e)** After the optimization is done, the next initial guess is calculated from 1) the last set of angles or 2) from the rotation rate, once there are at least two prior camera images that have been analyzed.

6.6.1 3D model, Euler rotations and projection images

The 3-D models of the ellipsoids were created in OpenSCAD⁴. OpenSCAD is an open-source CAD (Computer Aided Design) software that, unlike most other CAD software suites, utilizes a code-based approach instead of the more typical graphical user interface. Its main advantage is that it can create simple shapes very fast by short statements. Thus, parameters such as the resolution of the ellipsoidal shapes are easily changeable. As it creates shapes by use of a programming language, it also allows for more flexibility to change parts of the shapes at a later time. A counter-example being traditional CAD software that is often structured in an ordered fashion where subsequent alterations of the 3D model depend on the steps that were taken earlier. The 3D models are then exported in the STL ("Standard Triangle Language" or "Standard Tessellation Language") file format mode, which is a commonly used format and heavily utilized in 3D printing applications.

Back in the main code, the STL files are read in with the NumPy STL package. They are then converted into an object that contains all points forming the ellipsoid, in the form of a set of coordinates. Creating this object is done with a class called BodyProject. This class can be

⁴openscad.org

called with an option to pre-align the STL file, and thus the 3D model, in a desired manner. The options for alignment cover aligning the longest, intermediate and/or smallest dimensions of the body with the coordinate system axes. In preparation for the comparison between the 3D model and the camera images, the particles were aligned so that they would take the most likely orientation: largest projection area first. This orientation was only a first guess and would by no means determine the orientations that the particle can take during the analysis step that is covered in chapter 6.6.4.

The object can then be rotated with the function `rotate_body_euler2`. Using SciPy, the body is rotated around up to three Euler angles. First, a rotation matrix is calculated and converted into a rotation vector, which is then used together with the desired change of angle to rotate the object. The SciPy package used for this is `scipy.spatial.transform.Rotation.from_euler`. With this package, the user can choose a sequence of rotations around up to three axes. Importantly, it is possible to choose if the rotation is to be done extrinsically, meaning that the rotations occur about the axis of a fixed coordinate system, or intrinsically. An intrinsic rotation is a rotation around the axes of a coordinate system that is fixed to a moving body. Since the goal of this work was to track the rotations of an ellipsoid, the intrinsic approach would be more valuable, granting insight into the roll, pitch and yaw that the ellipsoidal particle encounters during its free-fall.

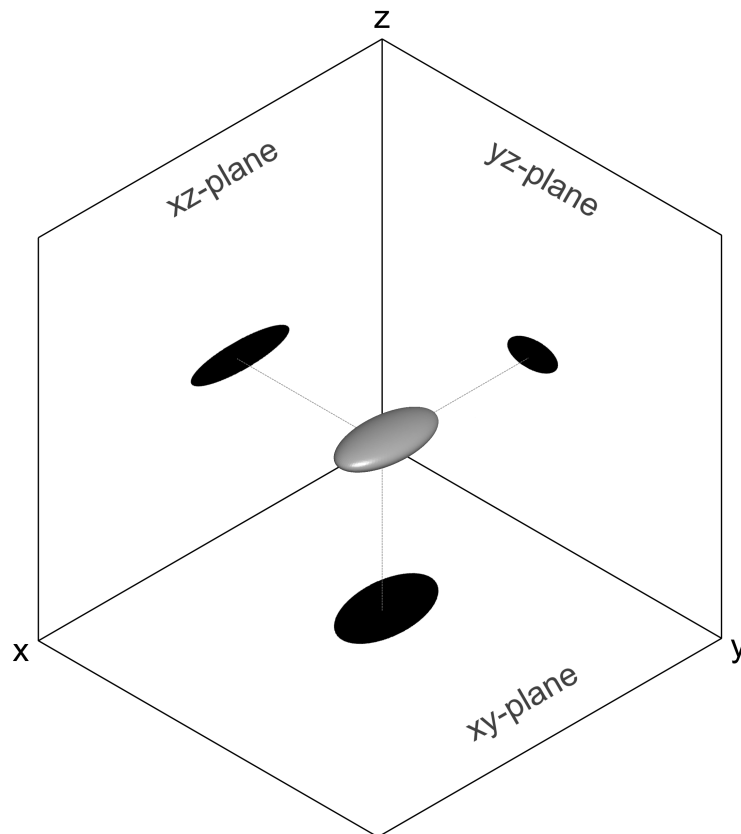


Figure 6.7: Projection images generated from a 3D model of an ellipsoid with shape EL 0.5, FL 0.5. The xz- and the yz-plane were then chosen for the comparison with the thresholded camera images.

In the next step, projection images of the 3D model are created with the function

`make_projection_image`. It generates a projection image either from 2D coordinates or from a 3-D model together with a provided axis that defines the view point. An image relative size parameter is determining the amount of white space padding around the created black particle area. Another parameter takes care of the general scaling so that the resulting image has the same resolution as the camera images. The function is processing the 3D model as follows: first the view point that gets parsed to the function is pre-selected by the user. This is done by choosing the desired plane that is spanned by combinations of the x-, y-, and z-axes. Then, a Matplotlib figure is created into which the hull of the 3D-model is parsed by `ConvexHull` (see chapter 6.5.2). The hull is filled by a filling function `plt.fill`. Saving and scaling the image is carried out with OpenCV functions. However, to receive the figure created with Matplotlib, the current buffer file has to be read out by a sub-function which returns an image as a NumPy array, `get_img_from_fig`. This is done as there is no built-in solution to communicate between the two libraries. OpenCV can then process the image it received as a NumPy array, upscale it according to the given scaling factor and save it under a desired file path and file format.

6.6.2 Creation of reference libraries

In order to reduce the difference between the camera images (representing the actual orientation of the particle) and the 3D model, an initial first guess for the particle orientation optimization scheme was needed. This first guess was obtained by comparing the two perpendicular camera images against a reference library. The reference library consisted of a large quantity (5832) of projection images that were generated by rotating the 3D model about all applicable axes. Applicable axes were all three possible axes (x, y, z) in the case of properly defined ellipsoids with three distinct ellipsoid axes. In the case that an ellipsoid displayed axisymmetry, only two axes would be used for the rotation. The axis that wouldn't create a distinguishable difference in the shadowgraphy image, was omitted. This was done as the code would not be capable of distinguishing any rotations that would happen around an axisymmetric axis due to the lack of markers on the ellipsoid.

The reference libraries were created with the method that was covered in chapter 6.6.1. Two options were available to choose from: 1) a smooth rotation from -180° to $+180^\circ$ or 2) a pseudo-random distribution in the same range, created by using the numpy random package. The smooth rotation has the advantage of being equally distributed, while the random rotation offers a larger number of unique orientations. Using the same methods as for the creation of the reference libraries, it was also possible to generate smooth or random artificial test cases that could be used to test the optimization routine. These test cases would later play a critical role in verifying the orientation determination (covered in chapter 6.7).

6.6.3 Cost calculation

A key component of an optimization problem is a **cost function** (also known as *objective function*, *error function* or *loss function*) that is being minimized. Minimization in this regard means that a quantity, called the *cost* should be reduced with each subsequent step of the optimization routine. Importantly, the cost does not necessarily represent a physical quantity but can be any quantity that the optimization code can reliably minimize (or maximize) [Raschka and Mirjalili, 2019]. An examples for this work could be the amount of pixels that are different between two images representing the orientation of a particle. As the amount of pixels that are different between two images depends on the image dimensions, it becomes clear that the pixel area is unique to the images that are analyzed. Depending on the chosen cost, it might not be comparable between different sets of images and experiments. As far as the optimizer is concerned, this does not matter: its only aim is to minimize the current value of the cost.

Calculating the cost between the orientation of the particle in the camera images and the 3D model was realized by two functions: `cost_img_compare` and `intersect_union`. When provided with a set of angles, `cost_img_compare` would call further functions to first rotate the 3D model about these angles and then generate projection images of the rotated 3D model for two view points. This was done to match the two view points of the camera setup. The rotation and the generation of the projection images are performed as detailed earlier. Both 3D model projection images are then compared to their respective camera images in `intersect_union`. Here, the following steps are performed:

1. Both the camera and the 3D model images are normalized by the maximum intensity and their values inverted. After this step, the black pixels of the ellipsoid area all have the value "1" and all of the white space around the ellipsoid has the value "0".
2. The normalized camera and 3D model images are summed. After the summation, the summed image is structured as follows:
 - Pixels with value "0" are white padding.
 - Pixels with value "1" are the difference between the camera and the 3D model.
 - Pixels with value "2" are the overlap between the camera and the 3D model.
3. The NumPy function `np.where` is used to find the difference.
4. The difference is returned to `cost_img_compare`.

The difference and the overlap between two ellipses is visualized in figure 6.8. Expressed as a formula, the cost was:

$$C_i = A_{diff} \quad (6.3)$$

Back in `cost_img_compare`, the costs for both view points are then summed as follows:

$$C = C_{cam1} + C_{cam2} \quad (6.4)$$

and returned to the main code. Please note that C_i is the same as C_1 in chapter 6.7.

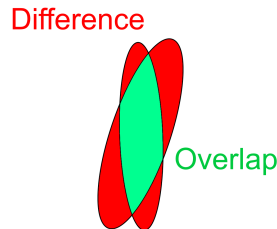


Figure 6.8: Visualization of the difference (red) and overlap (green) between two ellipses. The code will try to minimize the difference between the camera images and the 3D model by rotating the 3D model.

6.6.4 Minimizing cost: solvers and optimization approaches

In order to determine the orientation of the particle during its flight, the 3D model of the particle needed to be rotated such that it would correspond to the actual particle orientation for every set of camera frames. This was achieved in several steps:

1. In preparation of the analysis, a file was created that contained information about the start and end point of the video that would be used in the following. Specifying a start and end points was necessary as some of the frames at the beginning and end of the videos could be truncated. As truncated ellipses would lead the code to assume wrong orientations, these frames needed to be removed from the analysis. Therefore, the start and end points

corresponded to the first and last frame numbers that should be analyzed. Another reason to shorten a video would be if the particle had left the field of view in one camera but not the other. Since both camera views are necessary for proper analysis, frames where the particle was only visible in one view were excluded. The created file would then be read in and the information stored for the analysis steps that followed.

2. In the main loop, all camera frames were loaded and ran through the analyze projection function that was already covered in chapter 6.2. Next, the STL files that corresponded to the current shape were loaded as well, according to the method covered in chapter 6.6.1. The code would then begin looping through all camera frames in backwards order, beginning with the last recorded frame and ending with the first recorded frame. The logic behind this was to account for the apparent dampening that occurred during the free fall of the particle. Since the particle rotation rate appeared to be slower at the end of the video as compared to the beginning of the video, the rotation rate tracking would have an easier time starting with a slow rotation rate. This was especially true for particles that were tumbling with a very high rotation rate. In those cases parts of the beginning of the video could have been under-sampled the particle rotation due to the frame rate of the camera being too slow.
3. The actual minimization of the cost was performed by the SciPy function `optimize_minimize`. It acts as a framework into which the user can enter a function to be minimized and choose a solver to be used. `optimize_minimize` offers both constrained and unconstrained solvers, where constrained solvers minimize a given function within stated bounds and unconstrained solvers minimize without such restrictions. Three numerical solvers were tested by feeding them artificial test images and letting them rotate a 3D model of the particle in question to create minimal differences between the test images and the projection of the 3D model.

The solvers tested were: Powell, Nelder-Mead, and L-BFGS-B, where L-BFGS-B stands for limited-memory 'L' Broyden, Fletcher, Goldfarb and Shanno 'BFGS' (the algorithm's creators [Nocedal and Wright, 2006]) with bounds 'B', i.e. constrained. It was quickly decided that Powell would not provide the desired results and was therefore omitted. Of the two remaining solvers, Nelder-Mead was of the unconstrained kind and L-BFGS-B of the constrained type. The way the solvers perform their work will be covered in detail shortly.

Having a mixture of unconstrained and constrained solvers available would soon prove useful as covered in step 6. In any case, the solver would need to be provided with an initial guess for the particle orientation that is either taken by comparing against the reference library or, for later frames, acquired by estimating the current rotation rate and calculating the next likely set of angles from it. A short overview of the general procedure was already shown in figure 6.6.

In the following a short description of both solvers is given:

- (a) Nelder-Mead:

Nelder-Mead [Nelder and Mead, 1965] is an unconstrained optimization method for minimizing non-linear functions $f(x)$ for $x \in R$. For every iteration of the Nelder-Mead solver, a current simplex is defined. A simplex is a generalization of a triangle or tetrahedron to arbitrary dimensions. For example, a 0D simplex is a point, a 1D simplex a line segment, 2D simplex is a triangle and so forth. Consider that the gradient of the problem at hand was structured like a 2D height map, similar to a map showing elevation of mountains and valleys. A 2D simplex could be placed onto the surface of this height map to find local minima. For this, Nelder-Mead is ordering the current set of vertices $x_1^{(k)}, \dots, x_{n+1}^{(k)}$ that form the edge points of the simplex at each iteration k such that:

$$f(x_1^{(k)}) \leq f(x_2^{(k)}) \leq \dots \leq f(x_{n+1}^{(k)}). \quad (6.5)$$

As the goal is to minimize the function f , $x_1^{(k)}$ would be denoted the best point and $x_{n+1}^{(k)}$ would be the worst point. In the example of a 2D height map, the best point would exhibit the smallest elevation and the worst point would exhibit the highest elevation. In its quest to minimize f , the 2D simplex in the form of a triangle will now perform one of four possible operations: reflection, expansion, contraction (inside/outside) or shrinkage. An excellent overview of the different operations and their mathematical details is given by Wright [1996]. Figure 6.9 provides a visual overview of the different operations.

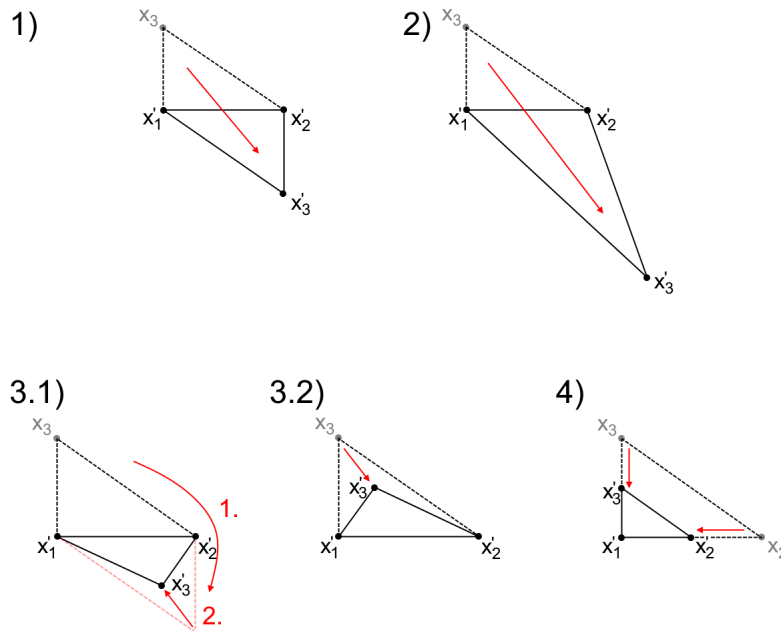


Figure 6.9: The different operations performed by the Nelder-Mead method: **1)** Reflection; the triangle is folded along one of its edges while retaining the same dimension. **2)** Expansion; folding over an edge while stretching the triangle. **3.1)** Outside contraction; the triangle is folded over an edge its area decreased. **3.2)** Inside contraction; the triangle remains in place and its area is decreased. **4)** Shrink; the triangle is shrunken while retaining its aspect ratios. Adapted from Wright [1996]

SciPy.optimize.minimize is a very convenient package to run Nelder-Mead and as such tempts the user to utilize it as a black box by providing only the function, an initial guess and a set of function parameters. Tunable parameters for running Nelder-Mead were the maximum amount of iterations and function evaluations.

(b) L-BFGS-B:

The L-BFGS-B method is a constrained limited memory variant of the unconstrained quasi-Newton optimization method BFGS.

Let's disentangle this statement: Unconstrained optimization, as mentioned in the section above, covering the Nelder-Mead solver, is the problem of minimizing a non-linear function $f(x)$ for $x \in R$ without stating bounds. Unconstrained problems can often be solved using techniques based on Newton's method, a *line search* strategy. Newton's method is to develop a model function, typically a quadratic one, which is derived from a local gradient and information about the *Hessian*, a specific type of

symmetric matrix that describes the local curvature of a function. From here, a suitable iteration "step" is calculated to minimize the function [Wright, 1996]:

$$\min f(x_k + \alpha p_k) \quad (6.6)$$

Here, α is the step length with $\alpha > 0$, p_k the search direction along a gradient and x_k the current iterate. With every iteration, a limited number of trial steps lengths α are created until a solution is found that loosely approximates the minimum. Then, a new search direction and step length are computed and the algorithm keeps repeating this process until a value below a threshold is reached or further iterations no longer improve the result [Nocedal and Wright, 2006].

The first quasi-Newton method was created by W.C. Davidon in the mid 1950s, a physicist attempting a long optimization calculation using Newton's method that would routinely crash before completing. The algorithm turned out to be much faster and more reliable than other existing methods at the time. Quasi-Newton methods only need to be supplied with the gradient of the function at each iteration. Measuring the changes in gradients, a model of the function good enough for convergence is created, speeding up the calculation process on the whole [Nocedal and Wright, 2006]. Today, a variety of quasi-Newton optimization methods exist with the most popular one being the BFGS method, named after its creators Broyden, Fletcher, Goldfarb and Shanno [Nocedal and Wright, 2006]. In short, its approach is the following:

Begin the derivation by forming a quadratic model of the objective function at the current iterate x_k :

$$m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p \quad (6.7)$$

where B_k is a symmetric positive definite matrix, called the *approximate* Hessian, that is updated at every iteration. p_k is the minimizer that computes the search direction of the algorithm and is given by:

$$p_k = -B_k^{-1} \nabla f_k. \quad (6.8)$$

The key difference between BFGS and Newton's method is its treatment of B_k that is updated in a simple manner with every iteration instead of being computed afresh at every iteration in Newton's method. The algorithm is said to be robust with a superlinear rate of convergence. Newton's method itself converges quadratically and therefore faster, but its cost per iteration usually is higher due to needing solution of a linear system and second derivatives. The information in the paragraph above was taken from [Nocedal and Wright, 2006] where the inclined reader can find a much more thorough mathematical description of the BFGS algorithm.

Continuing from the BFGS algorithm, the final step to cover L-BFGS-B is the limited memory ("L") and constrained, or bounded ("B"), approach taken there. In actuality, the variant L-BFGS [Byrd et al., 1995] was created prior to the variant on that variant, L-BFGS-B, but for simplicity L-BFGS-B will be covered directly without specifically detailing L-BFGS. L-BFGS-B was introduced by Zhu et al. [1997] with the aim to minimize a non-linear function of n variables $f(x)$, with stated upper (u) and lower (l) bounds such that

$$l \leq x \leq u. \quad (6.9)$$

The algorithm updates its Hessian with a limited-memory BFGS approximation once per iteration. This limited-memory matrix is then used to define a quadratic model of the function in question. Finally, a search direction is computed. The bounded part of L-BFGS-B is realized by presenting the limited-memory BFGS matrices in a compact form, following Byrd et al. [1995], which is efficient for bound-constrained problems.

According to its creators, the advantages of L-BFGS-B are 1) and easy-to-use code, 2) modest storage requirements and 3) low computational cost per iteration, independent of the function to be minimized. Disadvantages are that 1) it does not converge rapidly and can take many function evaluations to converge, 2) it may fail to obtain high accuracy on highly ill-conditioned problems and 3) no knowledge about the structure of the problem can be used as an input to accelerate convergence.

Both Nelder-Mead and L-BFGS-B solve for local minima. Therefore they depend on a well-chosen initial guess. In this work, the initial guess comes in the form of the three Euler angles that determine the orientation of the particle. The function that `optimize_minimize` is solving is `cost_image_compare`, which in turn calculates the cost with `intersect_union`, as described in chapter 6.6.3. Strategies to find an initial guess are discussed next.

4. The first initial guess is found by comparing the camera images to the reference library that was introduced in chapter 6.6.2. The function `library_comparison` is comparing all available reference images against the camera images and picks the reference images that displayed the smallest cost. The angles with which the reference images were created had been stored during the creation of the reference images and are therefore now available and used as the initial guess for the first frames. After being optimized by the solver, the resulting angles are stored in an array.
5. Starting from the second set of frames, the rotation rate is being calculated and tracked for a number for use a defined number of points. This is done by calling the function `extract_local_rotation_rate`. Internally, `extract_local_rotation_rate` is calling the NumPy function `polyfit`, which computes a least squares polynomial fit by minimizing the squared error

$$E_{\text{polyfit}} = \sum_{j=0}^k |p(x_j) - y_j|^2 \quad (6.10)$$

in the equations:

$$\begin{aligned} p_0 x_0^n + \dots + p_{n-1} x_0^1 + p_n &= y_0 \\ p_0 x_1^n + \dots + p_{n-1} x_1^1 + p_n &= y_1 \\ &\dots \\ p_0 x_k^n + \dots + p_{n-1} x_k^1 + p_n &= y_k \end{aligned} \quad (6.11)$$

where p are the equation coefficients⁵. With the polynomial, `extract_local_rotation_rate` calculates the most likely angles for the following set of frames and returns it to the main loop. This new set of angles can then be used as the next initial guess for the solver once at least two sets of frames have been analyzed.

6. Projection images displaying an almost round particle pose a particular challenge for an unbounded solver such as Nelder-Mead. The circularity, also called roundness, used in this work is the Cox circularity [Cox, 1965], defined as

$$\Theta_{\text{Cox}} = 4\pi A/P^2 \quad (6.12)$$

where A is the particle area visible in the projection image and P is the particle perimeter. Ranging from zero to one, the circularity gives an estimate of how circular an object is, where a circularity of one would correspond to a perfect circle.

A high circularity in one or both camera views, defined in this work as circularities where $\Theta_{\text{Cox}} \geq 0.9$, insufficiently constrains the particle orientation to estimate its orientation. Interestingly, regions (where a region is a number of frames) of high circularity do not create

⁵<https://numpy.org/doc/stable/reference/generated/numpy.polyfit.html>, accessed 14.04.2021

a high cost. Instead, a high cost is typically encountered after leaving an area of high circularity. This is due to the unconstrained nature of Nelder-Mead: when one camera view exhibits a high circularity, the solver is granted a lot of freedom to rotate the 3D-model into different orientations, as long as the final outcome matches the input camera image. Typically, regions of high circularity span at least several frames during which the solver can rotate the particle freely. When leaving this region, the particle is likely to be orientated in a manner that contradicts the orientation that is now given by the camera images. The solver will eventually correct its mistake and re-align the 3D-model but this process takes several frames to complete, creating a large cost for these frames.

One way of solving the issue is to constrain the possible rotation direction that the particle can take in high circularity regions. By utilizing the tracked rotation rate, together with strict bounds on the possible rotation that the solver has available, it is possible to better estimate the current position of the particle even in high circularity regions. When entering a high circularity region, the first step is then to estimate the initial guess for the new set of angles from the rotation rate, as detailed earlier. Then, the solver is given only a narrow angle to "wobble" the 3D-model into position. In this work, the freedom granted to the solver was $\pm 3^\circ$. The solver chosen for this was the L-BFGS-B solver, introduced in step 3.

6.7 Validation/Verification of the orientation determination

A successful validation seemed likely as the code uses similar methods to other works (e.g. Will et al. [2020]) and was showing promise in the simple form (see chapter 5). Verification of the performance of the orientation determination was done mainly by use of synthetic test cases that were simulating ideal input conditions. A new function was created that would rotate the 3D model of a chosen shape from -180° to $+180^\circ$ in a given number of steps. The user could also define which angles to rotate about, so combinations of alpha, beta, and/or gamma were possible. The function would then output black-and-white images, similar to the images created by the functions covered in chapter 6.6.2. These images then formed the input for the main code which would analyze the images as described in prior chapters. Finally, the expected angles were compared to the angles gained by the orientation determination scheme.

Before deciding on the cost that was introduced in chapter 6.6.3, several ways of calculating the cost were tested. These costs were mainly variations of combining the area of difference in pixels A_{diff} , the overlapping area A_{ovlp} and the area of the ellipse in the camera image (thresholded). The different costs that were tested are shown in table 6.1:

Cost name	Formula	Unit
C_1	A_{diff}	$[C_1] = \text{px}^2$
C_2	$A_{\text{diff}}^2/A_{\text{ovlp}}$	$[C_2] = \text{px}$
C_3	$A_{\text{diff}}/A_{\text{ellipse}}$	$[C_3] = 1$
C_4	$A_{\text{diff}}^2/(A_{\text{ovlp}} \cdot A_{\text{ellipse}})$	$[C_4] = 1$
C_5	$A_{\text{diff}}/(A_{\text{ovlp}} \cdot A_{\text{ellipse}})$	$[C_5] = 1/\text{px}$

Table 6.1: Different ways of calculating ellipse overlap cost. Please note that C_1 is the same as C_i in chapter 6.6.3.

C_1 is simply the area that defines the differences between the two ellipses that are being overlapped. It is a straight-forward solution that turned out to work best when compared to the other costs. C_2 - C_5 were different endeavours to normalize the cost and thus make it more comparable than the plain number of pixels that make up an area of the difference. Dividing the area of difference by the overlapping area like in C_2 would increase the speed with which the cost would increase or decrease compared to C_1 . As the difference was shrinking the overlap would increase

and vice versa. The cost was susceptible to a particular issue, however. If the camera thresholding failed and no ellipse was produced the projection of the 3D model would be compared against an empty, white image. This would lead the code to attempt to divide by zero and result in failure. C_3 , C_4 and C_5 were similar variations but here the area of the ellipse in the camera image was also used as another normalization method.

Figure 6.10 shows the large difference that changing the cost calculation can have on the quality of the optimization results. The top left shows a near-ideal result when tracking an ellipsoid of shape EL 0.2, FL 0.2 (an elongated and flat ellipsoid) that is rotated linearly from -180° to $+180^\circ$ around all three axes. This graph was created after estimating the rotation of the ellipsoid throughout a video consisting of 100 artificial frames using cost C_1 . The only notable deviation from an ideal result is seen at the last few frames. This small deviation is caused by the fact that the code is analyzing the frames in backward order (as in real camera images the ellipsoids are expected to have reduced rotation rates at the end of the video). As the first 2 frames are analyzed with help of a reference library, they cannot take advantage yet of the implemented rotation rate estimation.

In contrast to this, the graphs shown in the bottom left, calculated with C_3 , exhibit strong deviations and while the overall linear behavior is still partly visible, the deviation is clearly too large. Considering the corresponding costs of both rotation estimation results, it also becomes clear that they are quantitatively different. The highest cost values occur in different locations and notably C_1 only exhibits a high cost at the end of the video while C_3 has high cost values throughout. Please note that due to the different ways of calculating the costs, their absolute values are widely different between methods.

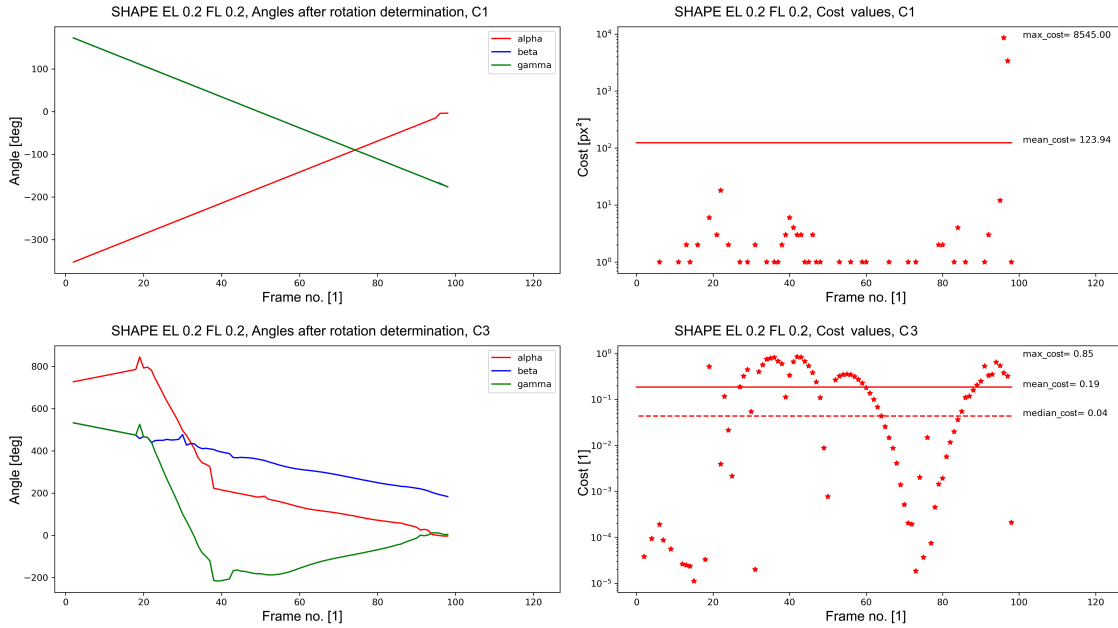


Figure 6.10: Effect of cost calculation type on orientation determination. **Top:** Orientation determination carried out with cost C_1 on 100 artificial test images. Left: The angles determined by the orientation determination follow the expected values with a small cost at the end of the video. The beta and gamma graphs overlap, hiding the beta graph behind the gamma graph. Right: The cost is generally small with exception of the last few frames that are representative of the small cost seen in the angles. The cost values are high as they correspond to the pixel area that makes up the difference between the overlaid projection images of the camera images and the 3D model. **Bottom:** Orientation determination carried out with cost C_3 on 100 artificial test images. Left: The angles determined by the orientation determination do not follow the expected values well. The general trend is still visible but large costs are apparent between frames 20 to 40. Right: The cost is generally high with exception of the first frames. In general, the cost C_3 is quantitatively and in unit different from the cost calculated with C_1 .

6.8 Results

Note: At the time of writing the code was not final. All results are therefore to be considered preliminary!

6.8.1 Data set

This set of experiments used the following camera configuration:

- Two cameras, 90° apart, were recording the falling particles from 20 mm below the insertion point onward for a distance of 27.5 mm. The cameras were therefore monitoring the distances 20 mm to 47.5 mm after the insertion point.
- One camera was recording the falling particles from 43.9 mm below the insertion point onward for a distance of 27.7 mm. The camera was therefore monitoring the distances 43.9 mm to 71.6 mm after the insertion point. This created an overlapping field of view with the top cameras for 3.9 mm between 43.9 mm after the insertion up to 47.8 mm after the insertion.
- The cameras were used with a 200 mm objective and no magnification rings. This led to a resolution of $6.75 \mu\text{m}/\text{px}$ for all three cameras.
- All cameras had their frame rate set to 2100 frames per second.

Table 6.2 shows the composition of the analyzed data set.

Shape	Number of experiments
EL0.2 FL0.2	7
EL0.2 FL0.5	7
EL0.2 FL1.0	7
EL0.5 FL0.2	7
EL0.5 FL0.5	7
EL0.5 FL1.0	7
EL1.0 FL0.2	7
EL1.0 FL0.5	7
EL1.0 FL1.0	5

Table 6.2: Multi-camera tracking: Amount of experiments per shape

As the code was still in development at the time of writing, not all of the videos were analyzed with sufficiently high confidence by the multi-camera tracking code. Therefore, results from three selected experiments will be covered as an outlook on the capabilities of the full code in the following sections. The shapes chosen for this were the following:

1. A prolate (EL 0.2, FL 1.0)
2. A bulky ellipsoid (EL 0.5, FL 0.5)
3. A disc (EL 1.0, FL 0.5)

6.8.2 Particle trajectories

Obtaining particle trajectories was an important goal in order to quantify the drift of the particles, estimate the influence of their rotation rates on the falling direction and to see if they exhibited settling behaviors such as fluttering or chaotic falling. All of these factors could influence the particle's drag coefficients and dictate how long the particle would remain in the atmosphere in a natural environment. Figures 6.11, 6.12 and 6.13 show the recorded trajectories of the three example particles from top and side views, as well as in a 3D view.

In general, the particles did not exhibit a lot of drift during the recorded distance. The prolate particle (EL 0.2, FL 1.0) showed the largest lateral drift of 0.6 mm over the distance of 17.5 mm along the y axis. Both the ellipsoid (EL 0.5, FL 0.5) and the disc (EL 1.0, FL 0.5) showed smaller lateral drift of 0.25 mm and 0.20 mm, respectively, also along the y axis. Lateral drift along the x axis was very small with a recorded value of 0.08 mm for all particles. Therefore, the particles went through a light zig-zag motion during their fall. All three particles displayed mostly smooth trajectories with one exception where the particle centre possibly was not correctly determined towards the end of the trajectory for the ellipsoid (EL 0.5, FL 0.5).

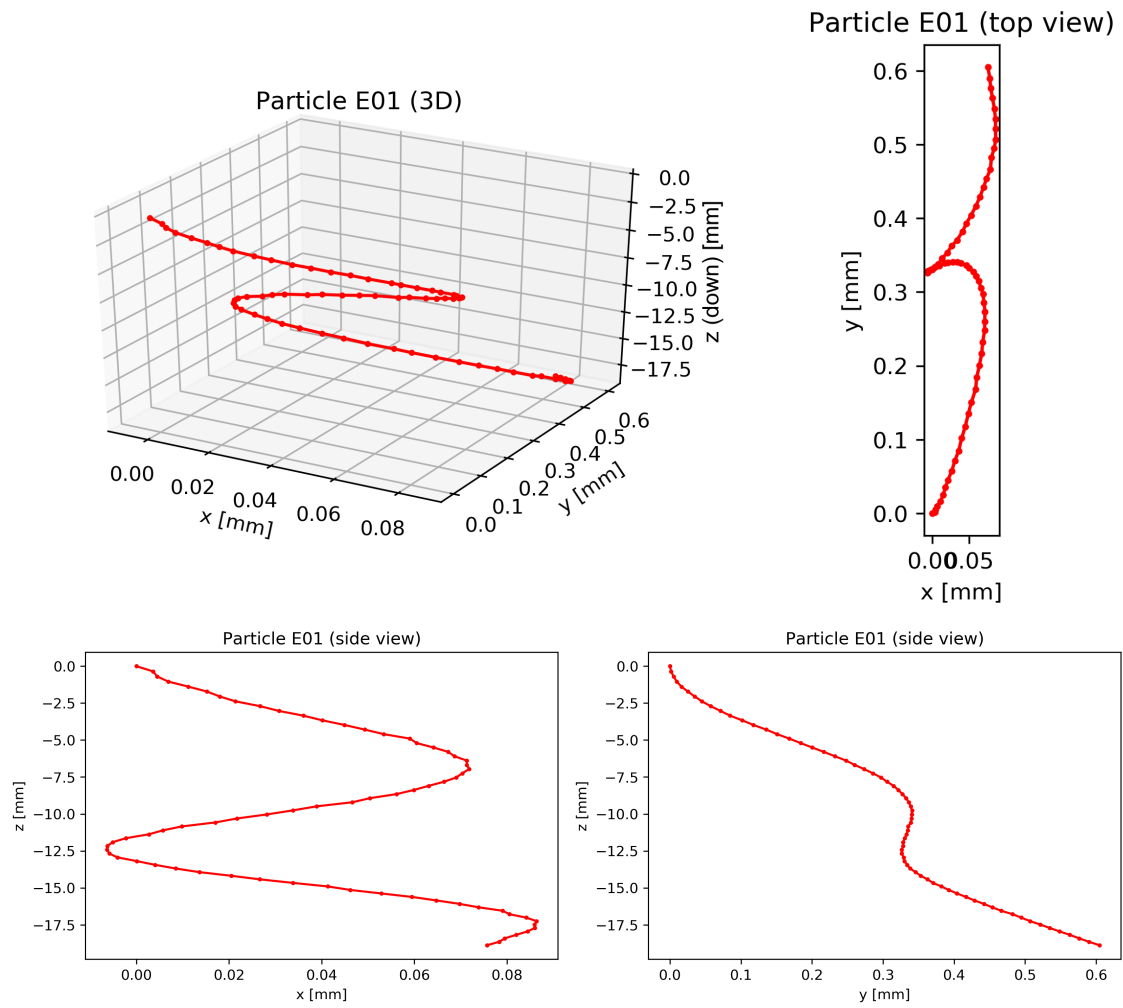


Figure 6.11: Trajectory extracted from particle with shape EL 0.2, FL 1.0 (prolate). **Top left:** 3D visualization of the particle trajectory during its free fall through the settlement chamber. The particle is settling mostly in a straight line while experiencing a light zig-zag motion in the xz -plane and an s-curve in the yz -plane. Axes not equal. **Top right, bottom left and bottom right:** top and side views of the particle's trajectory.

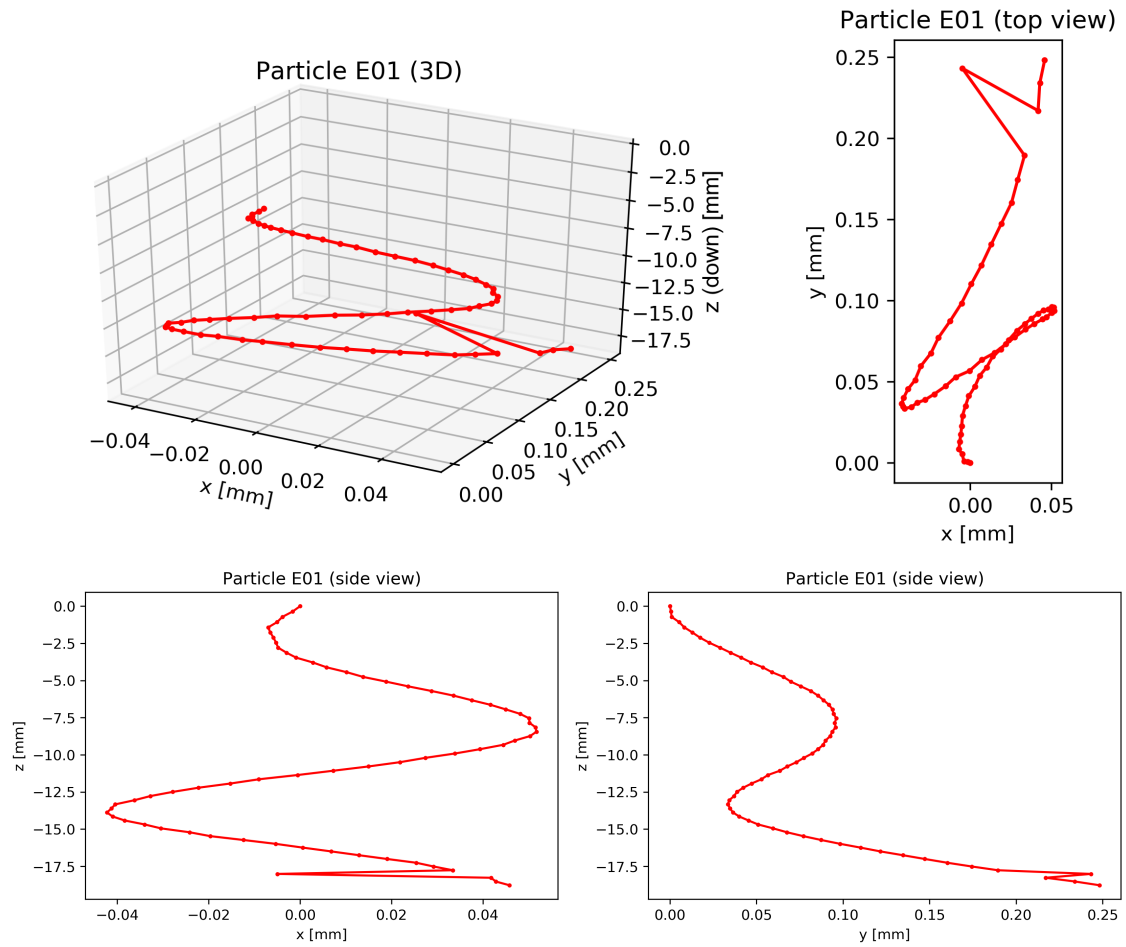


Figure 6.12: Trajectory extracted from particle with shape EL 0.5, FL 0.5 (ellipsoid). **Top left:** 3D visualization of the particle trajectory during its free fall through the settlement chamber. The particle is settling mostly in a straight line with while experiencing a light zig-zag motion in the xz -plane and an s-curve in the yz -plane. The sudden jump towards the end of the trajectory is likely due to an incorrect detection of the particle centre in an otherwise smooth trajectory. Axes not equal. **Top right, bottom left and bottom right:** top and side views of the particle's trajectory. Lateral drift occurs mostly along the y -axis but is small compared to the length of the falling distance.

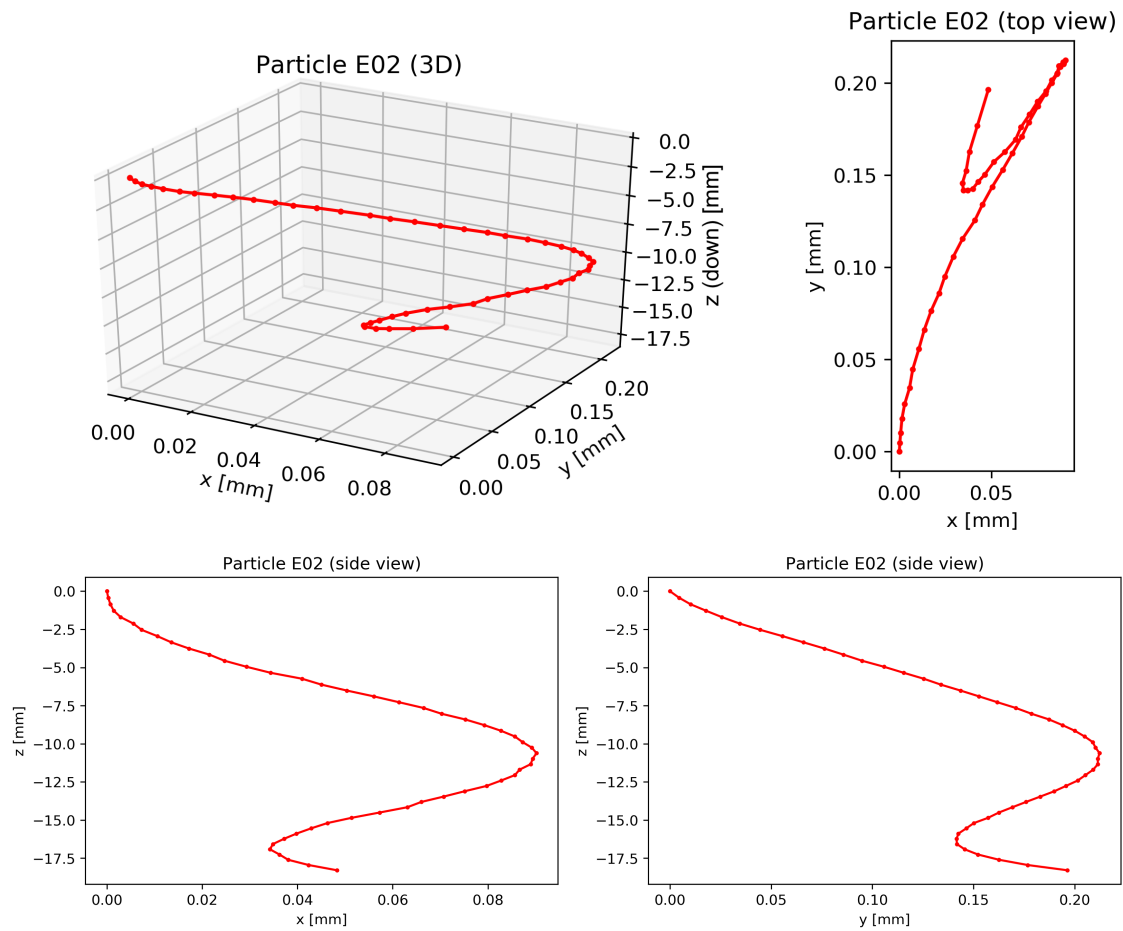


Figure 6.13: Trajectory extracted from particle with shape EL 1.0, FL 0.5 (disc). **Top left:** 3D visualization of the particle trajectory during its free fall through the settlement chamber. The particle is settling mostly in a straight line while experiencing a light zig-zag motion in the xz -plane and an s-curve in the yz -plane. Axes not equal. **Top right, bottom left and bottom right:** top and side views of the particle's trajectory. Lateral drift occurs mostly along the y -axis but is small compared to the length of the falling distance.

6.8.3 Transient dynamics

In this work, transient dynamics describe the transition period between the insertion point and the particle reaching its terminal velocity and, possibly, its steady-state orientation (depending on whether or not a steady-state orientation exists for the particle in question). Desired results of the multi-camera tracking code are centred around obtaining particle orientation statistics: Do the particles settle with a random orientation or with the largest/smallest projection area first? Do stable fix-points in the orientation exist or do the particles keep performing oscillatory motions while they fall? Other parameters of interest include the rotation rate, as well as the oscillation amplitude, frequency, dampening and ultimately the particle relaxation times.

All of the mentioned parameters can be obtained from tracking the angles of the ellipsoids during the transition period. The same three particles (prolate, ellipsoid, disc) from the trajectory results and their angles will be covered in the following. Figures 6.15, 6.16, and 6.17 show both composite images of the particles during their flight through the observation volume, as well as the corresponding angles and costs obtained from the orientation determination algorithm. For comparison between the three figures, the cost was normalized as follows:

$$C_{normalized} = \frac{1}{2} * \frac{C_1^{cam1} + C_1^{cam2}}{A_{mp}} \quad (6.13)$$

where A_{mp} is the maximum projection area that the particle can attain.

Please note that the absolute values of the angles can be shifted by $\pm 180^\circ$ for shadowgraphy projections of an ellipsoid recorded with two views. The projection would appear exactly the same if the particle was seen from one side and from the side 180° rotated about one of its axes. The initial guess that is gained from the comparison with the reference library would then determine the starting angle. For example, with starting angles $\alpha = 0$, $\beta = 0$, $\gamma = 0$, the particle rotation would then evolve from these initial angles. If, on the other hand, the starting angles were $\alpha = 180$, $\beta = 0$, $\gamma = 0$, then the recorded angles would evolve around these starting angles. An automatic method of shifting the angles to all be centred around the zero degree angles was considered but not yet implemented. Furthermore, particles that are tumbling during their free fall, meaning that they rotate around one of their axes repeatedly, display growing or decreasing angles up until their momentum is dampened and they begin to transition into an oscillatory motion. Figure 6.14 displays the definitions of angles alpha, beta and gamma.

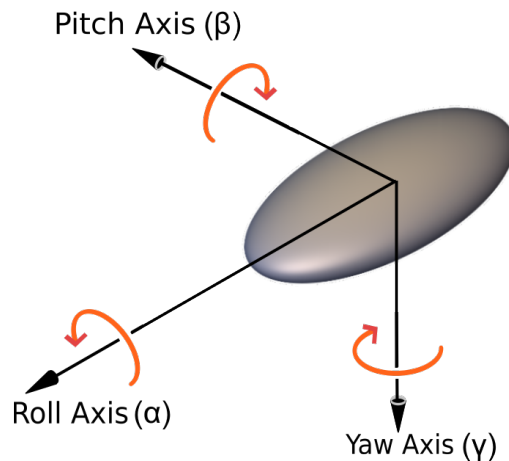


Figure 6.14: Particle orientation angles. Alpha is aligned with the longest axis of the ellipsoid, beta with the intermediate axis, and gamma with the shortest axis. In aeronautics, it is typical to name orientation changes along these axis "roll", "pitch" and "yaw".

Prolate spheroid (EL 0.2 , FL 1.0)

The settling behavior of the prolate spheroid, shown in figure 6.15, consists of a primary sinusoidal oscillation around the beta axis and a less pronounced sinusoidal oscillation around the gamma axis that is dampened in amplitude during the observed distance. Since the spheroid is axisymmetric, no oscillation around alpha could be recorded and alpha angles were set to zero. The oscillation around beta is centred at the angle $\beta = 13.7^\circ$, with $\beta_{max} = 34.1^\circ$, $\beta_{min} = -42.9^\circ$, and $\Delta\beta_{largest} = 77.0^\circ$. Two maxima are recorded at frames 26 and 80, as well as one minimum at frame 53 and a curve that indicates the existence of another minimum close to the first frame which wasn't recorded. The oscillation around gamma is centred at the angle $\gamma = -52.3^\circ$, with $\gamma_{max} = -36.3^\circ$, $\gamma_{min} = -60.9^\circ$, and $\Delta\gamma_{largest} = 24.6^\circ$. It goes through several maxima at frames 19, 43, and 77, as well as minima at frames 11, 34, 62 and possibly at the end of the recorded distance. The recorded angles are mostly changing in a smooth manner, indicating a good tracking. Around frame 20, a dip is seen in the beta curve, likely caused by imperfect tracking and subsequent self-correcting of the orientation determination. Fittingly, just before the dip is visible, the cost reaches its maximum and coincides with a jump in gamma. A high cost is generally found in the slopes between maximum and minimum angles, potentially hinting at the solver reacting too slowly to fast changes in angles.

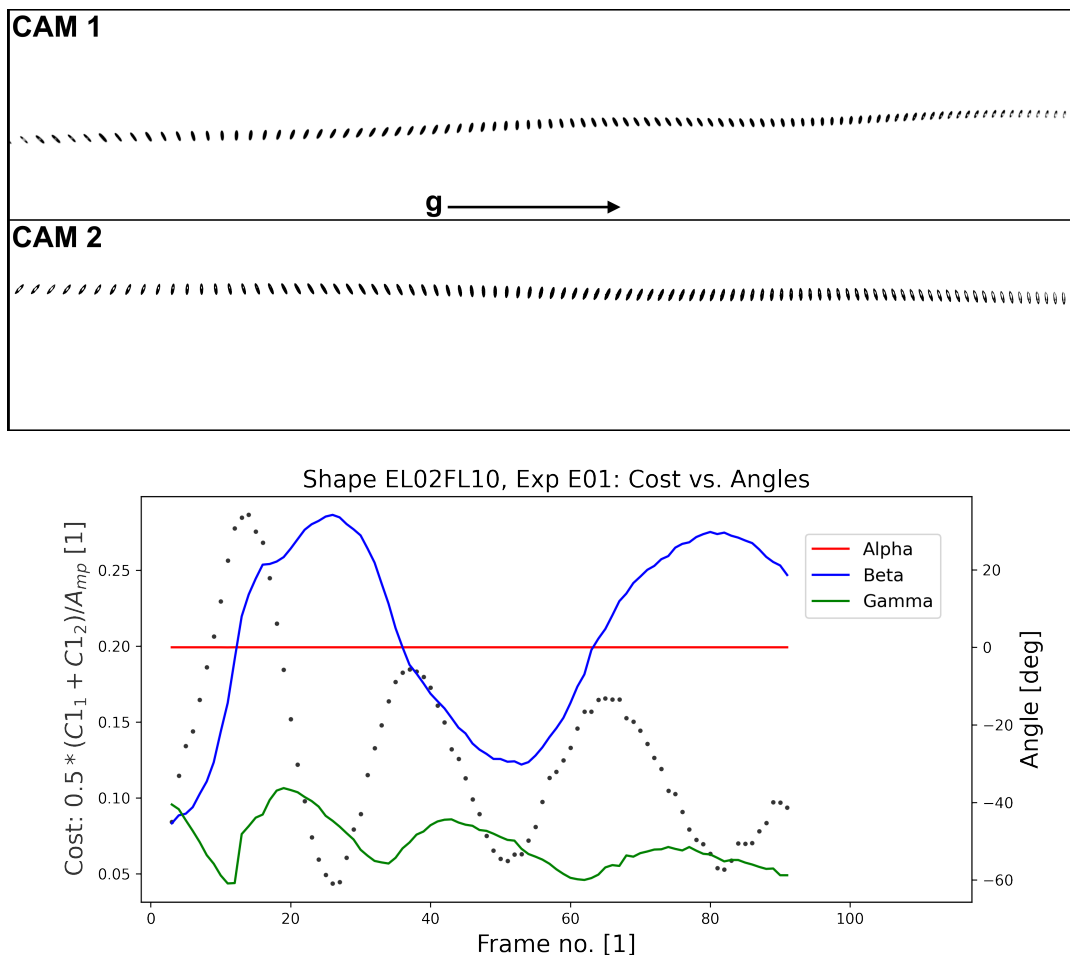


Figure 6.15: Particle flight composite image (**top**) and extracted angles (**bottom**) with cost for a **prolate** spheroid (EL 0.2, FL 1.0). The settling behaviour consists of a primary sinusoidal oscillation around the beta axis and a dampened secondary sinusoidal oscillation around the gamma axis. Alpha was not tracked due to the axisymmetry making it impossible to detect using the current setup.

Bulky ellipsoid (EL 0.5 , FL 0.5)

The settling behavior of the bulky ellipsoid, shown in figure 6.15, is similar to the one of the prolate spheroid but with less pronounced oscillations. It consists of a primary sinusoidal oscillation around the alpha axis and a less pronounced sinusoidal oscillation around the beta axis that is dampened in amplitude during the observed distance. It also exhibits small movements around gamma that do not lead to an oscillation. The oscillation around alpha is centred at the angle $\alpha = -179.8^\circ$, with $\alpha_{max} = -159.4^\circ$, $\alpha_{min} = -204.3^\circ$, and $\Delta\alpha_{largest} = 44.9^\circ$. This marks one of the aforementioned cases where the angles were shifted by -180° which could be shifted to zero. Two maxima are recorded at frames 31 and 74, as well as two minima at frames 11 and 51. The oscillation around beta is centred at the angle $\beta = 180.6^\circ$, with $\beta_{max} = 200.4^\circ$, $\beta_{min} = 164.1^\circ$, and $\Delta\beta_{largest} = 36.3^\circ$. In this case the angles could be shifted to zero, as well. Maxima and minima are more difficult to determine but can be found at frames 16 and 51 for the maxima and frame 36 in case of the minimum. The oscillation around gamma is centred at the angle $\gamma = -90.7^\circ$, with $\gamma_{max} = -78.2^\circ$, $\gamma_{min} = -96.2^\circ$, and $\Delta\gamma_{largest} = 17.9^\circ$. It goes through no discernible maxima and minima. The recorded angles of alpha are mostly changing in a smooth manner, indicating a good tracking. The beta and gamma curves, on the other hand display more jitter. The highest costs are found towards the beginning and the end of the recording.

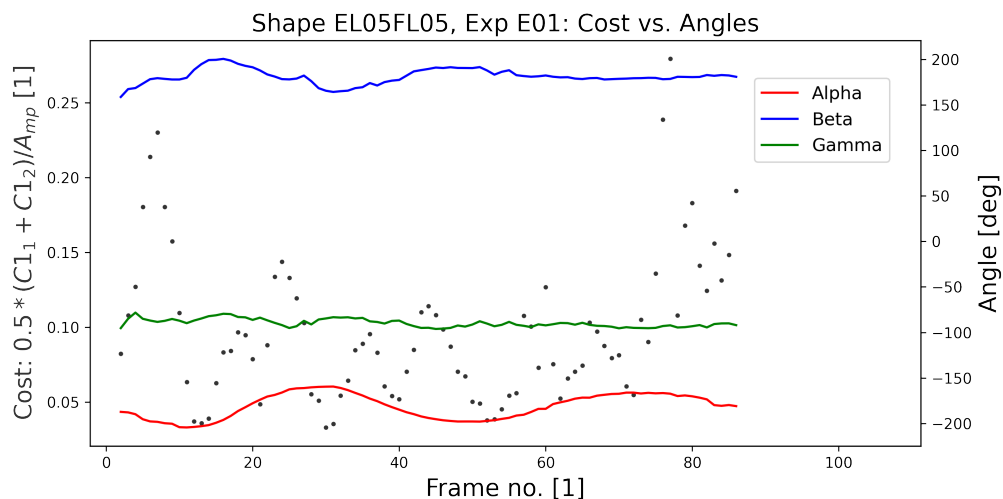
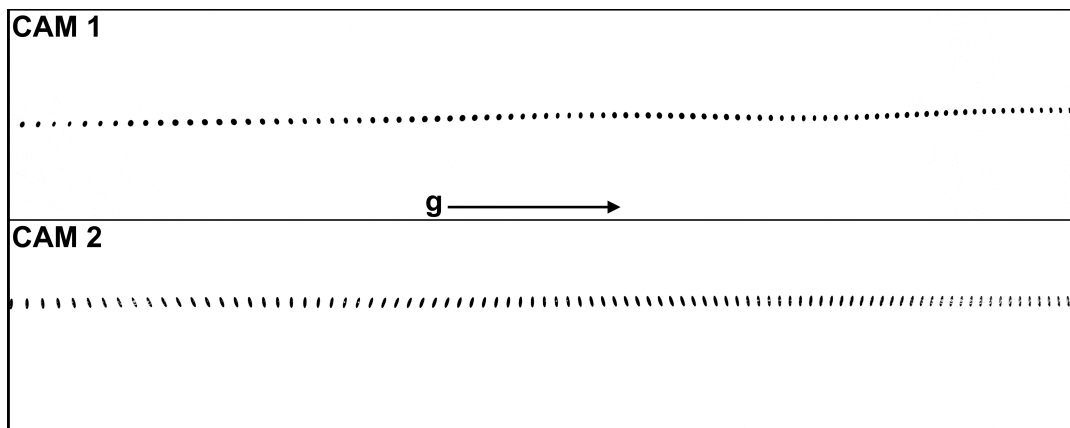


Figure 6.16: Particle flight composite image (**top**) and extracted angles (**bottom**) with cost for a bulky **ellipsoidal** particle (EL 0.5, FL 0.5). The settling behavior of the ellipsoid consists of a primary sinusoidal oscillation around the alpha axis, a dampened secondary sinusoidal oscillation around the beta axis and light jitter around the gamma axis.

Oblate spheroid (Disc) (EL 1.0 , FL 0.5)

The settling behavior of the disc, shown in figure 6.17, is unlike the behavior of the prior two shapes in that it indicates tumbling of the disc around the alpha angle during the first 30 frames after which the motion is transitioning into a sinusoidal oscillation. During the same time, the beta axis is exhibiting a pronounced sinusoidal oscillation. Since the disc is axisymmetric, no oscillation around gamma could be recorded and gamma angles were set to zero. For alpha the following angles were recorded: $\alpha_{max} = 241.8^\circ$, $\alpha_{min} = -21.2^\circ$, and $\Delta\alpha_{largest} = 263.0^\circ$, almost exceeding 270° which would correspond to three fourths of a complete rotation. One maximum is recorded at frame 34 and one minimum at frame 65. The oscillation around beta is centred at the angle $\beta = -18.8^\circ$, with $\beta_{max} = 42.1^\circ$, $\beta_{min} = -49.5^\circ$, and $\Delta\beta_{largest} = 91.5^\circ$. It goes through one maximum at frame 33, as well as two minima at frames 13 and 61. In the case of alpha, the recorded angles are mostly changing in a smooth manner, indicating a good tracking. Beta, however, exhibits jumps at three locations, at frame 8, 21 and 49. The cost is smallest where the angles reach their maxima.

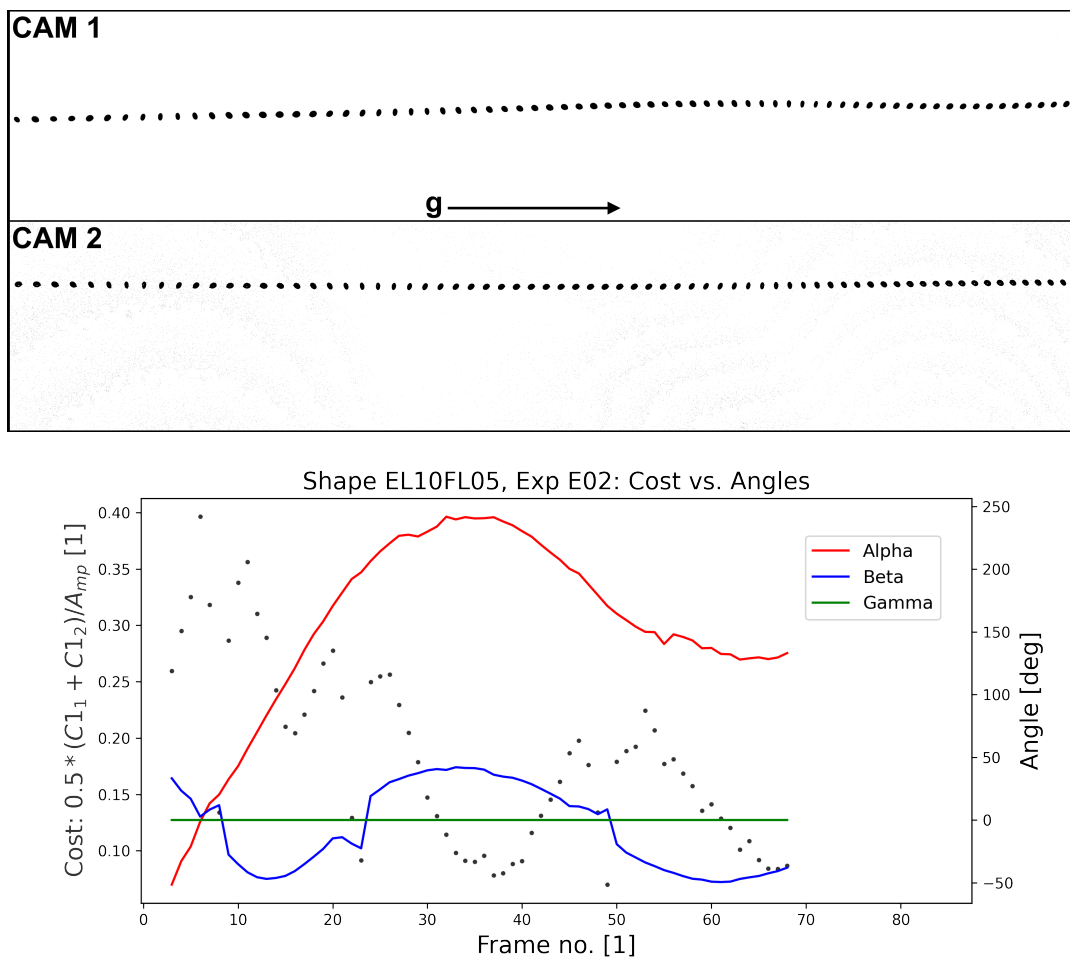


Figure 6.17: Particle flight composite image (**top**) and extracted angles (**bottom**) with cost for a **disc-shaped** particle (EL 1.0, FL 0.2). The disc first performs a tumbling motion around the alpha axis which then transitions into a sinusoidal oscillation. Further, a pronounced sinusoidal oscillation is visible around the beta axis with three jumps that are likely due to incorrect tracking and subsequent self-correction from the solver.

6.9 Conclusions

In conclusion, the multi-camera tracking code has produced first results and successfully reconstructed the trajectories and angles of several particles. The code is complex with many functions and sub-functions that needed to be thoroughly investigated in order to remove bugs and find optimal input parameters. At the same time, the videos provided to the code were often showing blurry particles and gradients in the background illumination. Between the technicalities of the code and the images of 3D models or zoomed-in camera frames, it is easy to lose the appreciation for the fact that these particles had an equivalent diameter of only 140 μm , about the thickness of human hair. In light of this, the code is performing fairly well. Nonetheless, several improvements are envisioned for further experiments, including but not necessarily limited to:

Code improvements:

1. Tuning the parameters of the solvers Nelder-Mead and L-BFGS-B could further improve the results. Specifically, faster responses from the solver in the general case and better fine-tuning of the input parameters for the high circularity case are sought after. As the multi-camera tracking code has grown in complexity, more parameter tests should be performed. For this, artificial test cases with added noise should be designed.
2. Getting the thresholding of the camera images correct is critical for a reliable rotation estimation. Oversized particles provide too much freedom to rotate the particle while undersized particles constrain the orientation in nonphysical ways. Potential solutions include doing an adaptive thresholding at the very beginning or reconsidering the black-white thresholding to instead make use of the gradient around the ellipsoid inside the camera images.

New experiments:

1. More uniform lighting and larger particles would provide higher quality inputs for the code to analyze.
2. Setup improvements such as automatic triggering of the cameras and a new calibration mask.
3. A fourth camera should be added so that 3D-tracking can be performed for both height ranges.

Chapter 7

Conclusion

7.1 Summary

Several technical developments, focused on atmospheric research on the microscale, were carried out in this work. The developments can roughly be divided into two parts: hardware and software.

On the hardware side, a droplet generator, capable of generating droplets either on-demand or as a continuous stream, was designed, built and successfully tested (Ch. 2). The nozzles used by the droplet generator, as well as micrometric particles, with precisely defined shapes, were created by using two-photon-polymerization (TPP). The commercial availability of TPP is still a relatively new development and as such, some difficulties were faced, leading to an extensive parameter search before the parts could be produced in desired qualities (Ch. 3). In order to perform experiments with the TPP particles, a settling chamber was designed, built and tested (Ch. 4). A matching shadowgraphy setup was built, calibrated and verified. This led to first free-fall experiments with ellipsoids in the intermediate regime with Reynolds particle numbers of $O(1)$ and particle-to-fluid density ratios of 1000. This was then used to verify the empirical drag coefficient model of Bagheri and Bonadonna [2016].

In order to process the videos of the falling particles, an analysis software was developed in two steps: First, a single-camera tracking code was designed as a precursor to the more complex multi-camera tracking code that followed (Ch. 5). The single-camera tracking approach acted as a proof-of-concept and established several methods that were later utilized in the multi-tracking code, such as the gradient based contour detection. The multi-tracking code, then, aimed at providing full 3D trajectory and orientation tracking (Ch. 6). Its condition is satisfactory, but requires a few more improvements before becoming fully operational. Preliminary results are promising and, hence, the work should be continued.

7.2 Outlook

7.2.1 Different particles and particle combinations

The selection of nine shapes that were used for the settling experiments immediately revealed the huge range of particle shapes that are omitted when choosing any selection of shapes. While ellipsoids with three distinct axes offer an adequate coverage of many shapes, the range of possible shapes in nature is vast. Many particles exhibit irregular shapes or composites of shapes such as the many types of ice crystals, snow flakes, sand particles or pollen to name a few. The list can be easily extended with dust, marine “snow”, ramified, non-homogeneous particles and complex shapes.

Another area of interest for experiments with TPP particles could be the interaction between two or more particles. In this case, the particles would likely affect each other in their settling behavior through their wakes. Going forward, turbulence experiments involving particles created with TPP are conceivable. Printed particles with well defined micrometric dimensions could be inserted into a vortex or a small-scale von-Karman flow chamber to explore undersampled regimes and particle-fluid interactions.

7.2.2 Setup improvements

Additional technical improvements could be carried out to increase the quality of the trajectory and orientation determination. Already first improvements have been implemented that were not part of this work anymore. These include an automated solution for the triggering of the cameras: A small electrical connection is bridged in the moment the particle rod hits the collision area and the resulting shorted circuit is used to communicate with the cameras. This solves the camera shift problem that was described in chapter 6.3. The new calibration mask that was mentioned in chapter 4.5 is also already included in the planning for future experiments. For instance, further enhancement to improve the setup would be a better illumination by simply exchanging the LEDs that strobe lights could be synchronized with the cameras. Additionally, a fourth camera should be added for 3D-tracking along the full height range. On the software side, many possible improvements have been outlined in chapter 6.9.

7.2.3 Lattice-Boltzmann simulations

At the time of writing, direct numerical simulations are being conducted by Dr. Taraprasand Bhowmick (Max Planck Institute for Dynamics and Self-Organization, the same institute as the author) using the Lattice-Boltzmann (LB) method. These numerical studies will be used to compare the dynamical transients of the settling particles of non-isotropic shapes with the experiments detailed in this work. The numerical results capture the details of the flow dynamics. It can be seen that for these ellipsoidal particles settling in air with a density ratio of 998, the particles reach their stable orientation of maximum projection area perpendicular to gravity through transient oscillations.

The Reynolds numbers of these investigated particles are in range of 2.5 to 4.5, which belong to the intermediate regime. There are limited studies in this range of Reynolds number, density ratio and range of particle shapes. The use of experimental data to recover the particle trajectories and terminal velocities in order to validate the results of numerical simulations, and conducting further simulations with a larger range of particle shapes and density ratios is the future aim of this project. With the help of newly available experimental and numerical data, the empirical models for the prediction of terminal velocities and drag coefficients of non-isotropic particles can be formalized as well as fine tuned. Figure 7.1 shows first results from the Lattice-Boltzmann method for the settling velocities and the particle orientations. The shapes considered for this set of simulations were combinations of elongation (EL) 0.2, 0.5, 0.8, 1.0 and flatness (FL) 0.2, 0.5, 0.8, 1.0. Please note that in the figure, EL is abbreviated as "e" and FL as "f".

Shown in the top part of figure 7.1 is the particle velocity v as a function of time for the particles while settling within the simulation bounds. The particles are at rest and when released with $v = 0$ m/s. Unlike in the experiments, the particles have no initial torque and are orientated with their longest axis aligned with the falling direction. Therefore, they begin their fall with an orientation where the sharpest edge of the particles determines the encountered drag. This "edge first" orientation is experiencing the smallest amount of possible drag when compared to other orientations. The particle will remain in this orientation indefinitely as long as there aren't any forces acting upon it that would lead to an orientation change.

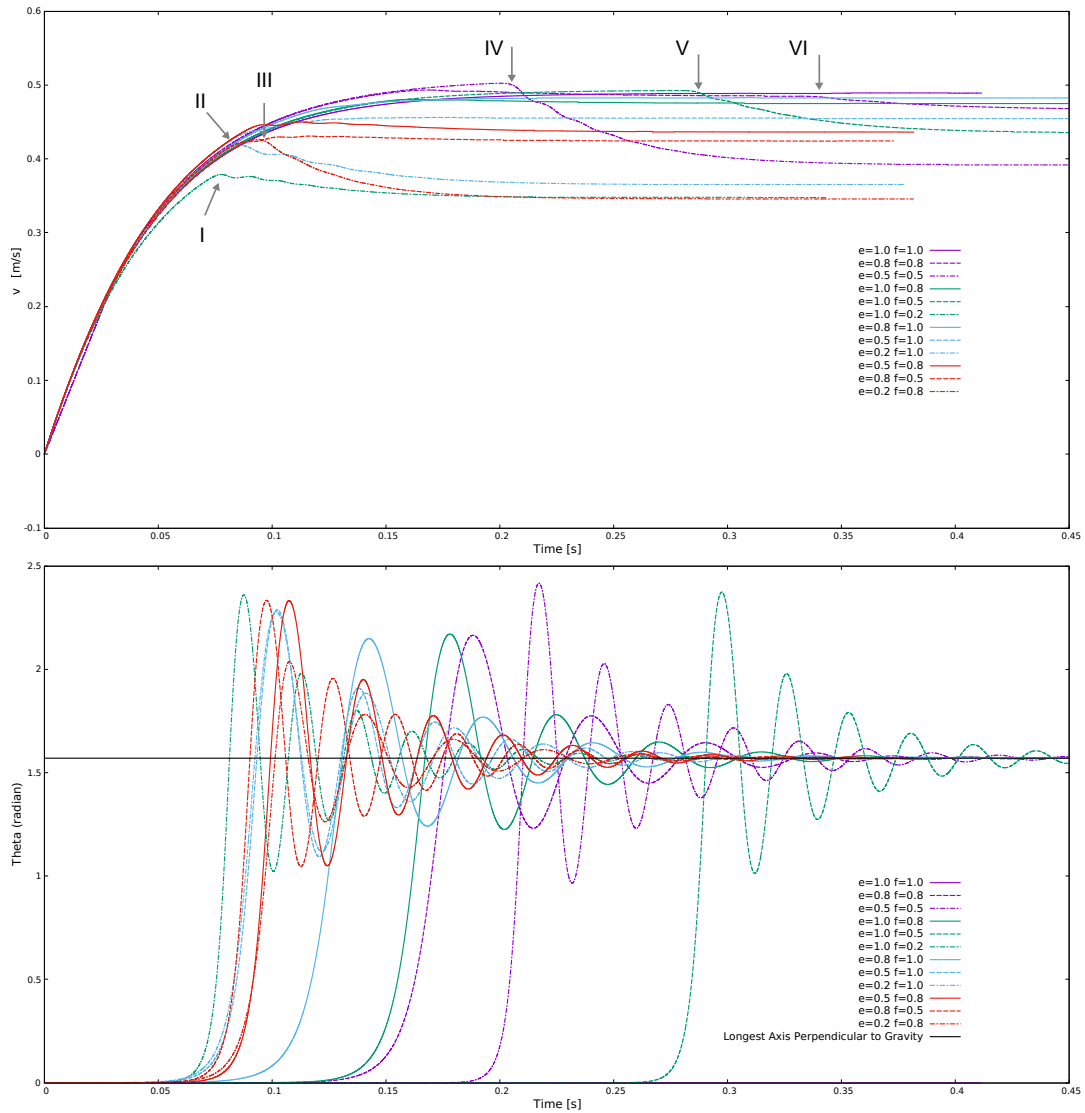


Figure 7.1: Lattice-Boltzmann simulations for several ellipsoids with combinations of elongation (e) 0.2, 0.5, 0.8, 1.0 and flatness (f) 0.2, 0.5, 0.8, 1.0. Preliminary results provided by Dr. Bhowmick, MPI DS.

Top: Particle velocity v as a function of time. Generally, the particle velocity follows a hyperbolic curve with a quick initial acceleration, followed by a plateau where the particles assume a steady terminal velocity. However, as indicated with markers **I-VI**, for several shapes, v exhibits a dip after the initial acceleration. The dip is caused by a sudden change of orientation of the particles, where the particles begin oscillating and eventually settle with their largest projection area orientated toward the falling direction. As the largest projection area is the orientation with the largest possible drag, the velocity of the particles is smaller than before the dip occurs.

Bottom: Angle theta (Θ) as a function of time. Θ describes the angle of the longest axis of the ellipsoid compared to the z coordinate (where z is aligned with gravity) of an external coordinate system fixed to the simulation domain. Here, $\Theta = 0 = 0^\circ$ equates to the particle falling with its edge first and $\Theta \approx 1.57 = 90^\circ$ equates to the particle falling with its maximum projection area aligned toward gravity. The particles are initially setup to be orientated with $\Theta = 0$. Shortly after release, all particles experience a fast change of orientation, indicated by an exponential growth of their Θ . The onset of this change of orientation strongly depends on the particle shape, with shape EL 1.0, FL 0.2 being the first to change its orientation and shape EL 1.0 FL 0.5 being the last to do so. All simulated particles overshoot their steady orientation and begin a dampened oscillation until they eventually reach $\Theta \approx 1.57$, the orientation with the largest projection area normal to the settling direction.

However, as can be seen in the bottom part of figure 7.1, all particles very quickly experience a change in their angle Θ , where Θ is the angle between the falling direction (the z axis of the computational domain) and the longest axis of the ellipsoidal particle. Indeed, the markers **I-V** in the top part of figure 7.1 show visible dips in particle velocity v . These dips correspond to the exponential growth of Θ in the bottom part of figure 7.1. The reason for the occurrence of such a dip is that the particles begin to tumble but then overshoot the angle that would provide a steady state orientation. The exception in this case is marker **VI** (Shape EL 0.8 FL 0.8), where the particle already went through its largest change of orientation prior to the recorded dip in v . However, this dip in v is much smaller than the dips at markers **I-V**.

As can be seen in the bottom part of figure 7.1, a steady state orientation is found at $\Theta \approx 1.57 = 90^\circ$ for all particles. This corresponds to the particles settling with their largest projection area oriented normal to the flow and, simultaneously, the orientation of maximum drag. Typically, the particles oscillate several times before assuming this largest projection area orientation. Hence, a dampened oscillation centered around $\Theta \approx 1.57$ can be seen following the exponential growth in Θ .

Of the particles that are marked, most shapes (**II**: EL 0.2 FL 1.0, **III**: EL 0.2 FL 0.8, **IV**: EL 0.5 FL 0.5, **V**: EL 1.0 FL 0.5, **VI**: EL 0.8 FL 0.8) have a flatness $FL \geq 0.5$ and are therefore found to have relatively rounded edges (for a quick overview of the shapes, refer to figure 3.8). The exception to this is found at marker **I** with shape EL 1.0 FL 0.2, a very flat disc. However, this particular shape is also the fastest in changing its orientation to assume largest projection area settling. Therefore, it is plausible that the absence of a dip in v comes from the particle orientation not overshooting the steady state as much as the other shapes that were marked.

Finally, table 7.1 shows the terminal velocities v_t that were obtained from the Lattice-Boltzmann (LB) simulations in comparison to the values from the experiments and the empirical model of Bagheri and Bonadonna [2016]. As can be seen from studying the table, the simulated values largely agree with the experiments and the empirical model. This marks a great first step in verifying the LB simulations and enhances the confidence in the simulations going forward. Additional tests and simulations are planned, including starting with largest projection area orientations and applying initial torque to the simulated particles in order to have an even closer representation of the experiments. On the flip-side, the LB simulations allow the examination of interesting details, such as the observed dips in v that were described earlier.

Shape	a, b, c [μm]	\mathbf{v}_t^{Sim} [m/s]	\mathbf{v}_t^{BB16F} [m/s]	\mathbf{v}_t^{Cam1} [m/s]	\mathbf{v}_t^{Cam2} [m/s]
EL 0.2 FL 1.0	410.0, 81.8, 81.8	0.365	0.396	0.351	0.353
EL 0.5 FL 0.5	280.0, 140.0, 70.0	0.392	0.405	0.393	0.413
EL 0.5 FL 1.0	222.2, 111.0, 111.0	0.455	0.479	0.482	0.487
EL 1.0 FL 0.2	239.4, 239.4, 47.9	0.348	0.365	0.347	0.345
EL 1.0 FL 0.5	176.4, 176.4, 88.2	0.435	0.459	0.463	0.463
EL 1.0 FL 1.0	140.0, 140.0, 140.0	0.489	0.519	0.528	0.532
EL 0.2 FL 0.8	441.0, 88.0, 70.6	0.346	0.371	-	-
EL 0.5 FL 0.8	239.4, 119.7, 95.8	0.436	0.456	-	-
EL 0.8 FL 0.5	204.7, 163.7, 81.9	0.424	0.443	-	-
EL 0.8 FL 0.8	175.0, 140.0, 112.0	0.467	0.490	-	-
EL 0.8 FL 1.0	162.0, 130.0, 130.0	0.483	0.509	-	-
EL 1.0 FL 0.8	150.8, 150.8, 120.64	0.475	0.502	-	-

Table 7.1: Comparison of terminal velocities for ellipsoidal particles attained from Lattice-Boltzmann simulations \mathbf{v}_t^{Sim} , the empirical model from Bagheri and Bonadonna [2016], \mathbf{v}_t^{BB16F} , and experiments. Here, \mathbf{v}_t^{Cam1} and \mathbf{v}_t^{Cam2} are the median terminal velocities for cameras one and two from table 5.3. The terminal velocities are given for several shapes of differing elongation (EL) and flatness (FL). The dimensions "a", "b" and "c" denote the lengths of the principal axes of the ellipsoidal particles. For shapes with either EL or FL = 0.8, no experiments have been performed at the time of this writing and hence, only the LB and BB16F values are stated. Equally, several shapes for which experiments have been performed are omitted as no finished LB simulations were available at that time.

7.2.4 Final remarks

In conclusion, two setups were created from scratch with the main goal of enabling the exploration of the dynamics of cloud microphysics. Both the droplet generator for drop-on-drop collision experiments, and the settling particles experiment were designed, built, and successfully delivered initial results. During the building phase of the droplet generator, it was realized that the recently commercially available Two-Photon-Polymerization (TPP) printing that was used for the nozzles, held an opportunity for the study of the settling dynamics of precisely defined 3D-printed particles. The particles would enable studies in so far unreachable regimes in terms of size and density ratio to the settling medium (air). The focus of the work was then switched to mainly encompass this new settling experiment.

The settling particles setup was designed, iterated upon and tested several times before moving on to the creation of a complex code base for the analysis of the experiments. This code base was then built up from single camera tracking to encompass full 3D tracking in space and time, as well as the determination of the orientation of the particles. Only with this orientation determination would it be possible to gain accurate statistics for the particle drag/settling velocities, the particle response times, and expected particle trajectories.

Several improvements were already planned and their implementation set in motion during the writing of this thesis. On the mechanical side, four cameras were to be used for future experiments, the camera mounts would be improved upon by making them more sturdy and providing a fixed base for the cameras, and better calibration masks and calibration procedures were sought after. The setups observation volume could also be widened/lengthened to study larger particles. Additionally, a new light source to remove gradients in the images, and enable synchronization between the light source and the camera frame rate were goals for future experiments. With such a light source, possible heating of the experimental setup could be widely avoided. Additionally, an automatic trigger of the cameras could be of use for faster processing of the videos.

In terms of software, the Euler tracking was to be replaced with Quaternion rotations to avoid the problem of gimbal locking where effectively rotation directions become locked. Another plan was to enable more pre-processing of the data, for example by creating experiment information dictionary files and pre-cropping the individual frames around the ellipsoids which would later speed up processing times. The numerical solvers could also be further fine-tuned and initial guesses improved, for example by utilizing splines or a Savitzky–Golay filter to determine the next likely orientation going from one frame to the next.

In the end, studying micrometric particles turned out to be a complex endeavour, necessitating many parts (mechanical, software, procedures) to work together in tandem. The author hopes to have furthered their study with this work and wishes future researchers of this topic the best of luck.

Bibliography

- A simple large-scale droplet generator for studies of inkjet printing. *Review of Scientific Instruments*, 79(7), 2008. ISSN 00346748. doi: 10.1063/1.2957744.
- In situ investigation of the shrinkage of photopolymerized micro/nanostructures: the effect of the drying process. *Optics Letters*, 37(4):710, feb 2012. ISSN 0146-9592. doi: 10.1364/OL.37.000710. URL <https://www.osapublishing.org/abstract.cfm?URI=ol-37-4-710>.
- E. Achenbach. Experiments on the flow past spheres at very high reynolds numbers. *Journal of Fluid Mechanics*, 54(3):565–575, 1972.
- M. L. Albertson. Effect of shape on the fall velocity of gravel particles. *CER*; 54-5, 1953.
- A. L. Alldredge and M. W. Silver. Characteristics, dynamics and significance of marine snow. *Progress in oceanography*, 20(1):41–82, 1988.
- Z. Arsenijević, Ž. Grbavčić, R. Garić-Grulović, and N. Bošković-Vragolović. Wall effects on the velocities of a single sphere settling in a stagnant and counter-current fluid and rising in a co-current fluid. *Powder Technology*, 203(2):237–242, nov 2010. ISSN 00325910. doi: 10.1016/j.powtec.2010.05.013. URL <https://linkinghub.elsevier.com/retrieve/pii/S0032591010002512>.
- N. Ashgriz. *Handbook of atomization and sprays: theory and applications*. Springer Science & Business Media, 2011.
- H. Awbi and S. Tan. Effect of wind-tunnel walls on the drag of a sphere. *Journal of Fluids Engineering*, 103:461–465, 1981. URL <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Effect+of+Wind-Tunnel+Walls+on+the+Drag+of+a+Sphere#0http://asmed1.org/getabs/servlet/GetabsServlet?prog=normal&id=JFEGA4000103000003000461000001&idtype=cvips&gifs=Yes&ref=no>.
- G. Bagheri and C. Bonadonna. On the drag of freely falling non-spherical particles. *Powder Technology*, 301:526–544, 2016.
- G. P. Bewley, E.-W. Saw, and E. Bodenschatz. Observation of the sling effect. *New Journal of Physics*, 15(8):083051, 2013.
- E. Bodenschatz, S. P. Malinowski, R. A. Shaw, and F. Stratmann. Can we understand clouds without turbulence? *Science*, 327(5968):970–971, 2010.
- S. Bony, B. Stevens, F. Ament, S. Bigorre, P. Chazette, S. Crewell, J. Delanoë, K. Emanuel, D. Farrell, C. Flamant, et al. Eurec 4 a: a field campaign to elucidate the couplings between clouds, convection and circulation. *Surveys in Geophysics*, 38(6):1529–1568, 2017.
- O. Boucher, D. Randall, P. Artaxo, C. Bretherton, G. Feingold, P. Forster, V.-M. Kerminen, Y. Kondo, H. Liao, U. Lohmann, et al. Clouds and aerosols. In *Climate change 2013: the physical science basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, pages 571–657. Cambridge University Press, 2013.

- C. Bourotte, M. Forti, A. Melfi, and Y. Lucas. Morphology and solutes content of atmospheric particles in an urban and a natural area of são paulo state, brazil. *Water, Air, and Soil Pollution*, 170(1):301–316, 2006.
- L. Bourouiba, E. Dehandschoewercker, and J. W. Bush. Violent expiratory events: on coughing and sneezing. *Journal of Fluid Mechanics*, 745:537–563, 2014.
- G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- A. Bransky, N. Korin, M. Khoury, and S. Levenberg. A microfluidic droplet generator based on a piezoelectric actuator. *Lab Chip*, 9(4):516–520, 2009. ISSN 1473-0197. doi: 10.1039/B814810D. URL <http://xlink.rsc.org/?DOI=B814810D>.
- N. Brosse and P. Ern. The motion of an axisymmetric body falling in a tube at moderate reynolds numbers. *Journal of Fluid Mechanics*, 714:238–257, 2013.
- R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, sep 1995. ISSN 1064-8275. doi: 10.1137/0916069. URL <http://epubs.siam.org/doi/10.1137/0916069>.
- J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.
- H. L. Capelo, J. Moláček, M. Lambrechts, J. Lawson, A. Johansen, J. Blum, E. Bodenschatz, and H. Xu. Observation of aerodynamic instability in the flow of a particle stream in a dilute gas. *Astronomy & Astrophysics*, 622:A151, 2019.
- A. Castrejón-Pita, J. Castrejón-Pita, and G. Martin. A novel method to produce small droplets from large nozzles. *Review of scientific instruments*, 83(11):115105, 2012.
- R. K. Chakrabarty, N. D. Beres, H. Moosmüller, S. China, C. Mazzoleni, M. K. Dubey, L. Liu, and M. I. Mishchenko. Soot superaggregates from flaming wildfires and their direct radiative forcing. *Scientific reports*, 4(1):1–9, 2014.
- A. U. Chen and O. A. Basaran. A new method for significantly reducing drop radius without reducing nozzle radius in drop-on-demand drop production. *Physics of Fluids*, 14(1), 2002. ISSN 10706631. doi: 10.1063/1.1427441.
- R. Chhabra. Wall effects on free-settling velocity of non-spherical particles in viscous media in cylindrical tubes. *Powder Technology*, 85:83–90, 10 1995. ISSN 00325910. doi: 10.1016/0032-5910(95)03012-X. URL <http://linkinghub.elsevier.com/retrieve/pii/003259109503012X>.
- R. Chhabra, L. Agarwal, and N. K. Sinha. Drag on non-spherical particles: an evaluation of available methods. *Powder technology*, 101(3):288–295, 1999.
- J. Chicheportiche, J. Zainoun, J. Renaudeau, M. Jenger, and G. Liu. A new vibro-rotating spray generator. *ReT*, 1000(10000):100000, 2011.
- R. Clift and W. Gauvin. Motion of entrained particles in gas streams. *The Canadian Journal of Chemical Engineering*, 49(4):439–448, 1971.
- R. Clift, J. R. Grace, and M. E. Weber. *Bubbles, drops, and particles*. Dover Publications, 2005.
- R. Cox. The steady motion of a particle of arbitrary shape at small reynolds numbers. *Journal of Fluid Mechanics*, 23(4):625–643, 1965.
- J. C. Crocker and D. G. Grier. Methods of digital video microscopy for colloidal studies. *Journal of Colloid and Interface Science*, 179(1):298–310, 1996. ISSN 0021-9797. doi: <https://doi.org/10.1006/jcis.1996.0217>. URL <https://www.sciencedirect.com/science/article/pii/S0021979796902179>.

- P. Dames, B. Gleich, A. Flemmer, K. Hajek, N. Seidl, F. Wiekhorst, D. Eberbeck, I. Bittmann, C. Bergemann, T. Weyh, et al. Targeted delivery of magnetic aerosol droplets to the lung. *Nature nanotechnology*, 2(8):495, 2007.
- N. Daphalapurkar, F. Wang, B. Fu, H. Lu, and R. Komanduri. Determination of mechanical properties of sand grains by nanoindentation. *Experimental Mechanics*, 51(5):719–728, 2011.
- A. B. Davis, A. Marshak, H. Gerber, and W. J. Wiscombe. Horizontal structure of marine boundary layer clouds from centimeter to kilometer scales. *Journal of Geophysical Research: Atmospheres*, 104(D6):6123–6144, 1999.
- R. G. Denning, C. F. Blanford, H. Urban, H. Bharaj, D. N. Sharp, and A. J. Turberfield. The Control of Shrinkage and Thermal Instability in SU-8 Photoresists for Holographic Lithography. *Advanced Functional Materials*, 21(9):1593–1601, may 2011. ISSN 1616301X. doi: 10.1002/adfm.201002653. URL <http://doi.wiley.com/10.1002/adfm.201002653>.
- D. Deponte, J. Mckeown, U. Weierstall, R. Doak, and J. Spence. Towards etem serial crystallography: Electron diffraction from liquid jets. *Ultramicroscopy*, 111(7):824–827, 2011.
- J.-P. W. Desforges, M. Galbraith, and P. S. Ross. Ingestion of microplastics by zooplankton in the northeast pacific ocean. *Archives of environmental contamination and toxicology*, 69(3):320–330, 2015.
- H. Dong, W. W. Carr, and J. F. Morris. An experimental study of drop-on-demand drop formation. *Physics of Fluids*, 18(7), 2006. ISSN 10706631. doi: 10.1063/1.2217929.
- Y.-p. Du, C. Wei, Z.-x. Wang, S. Li, H.-b. He, and G.-x. Jia. Liliun spp. pollen in china (liliaceae): taxonomic and phylogenetic implications and pollen evolution related to environmental conditions. *PLoS One*, 9(1):e87841, 2014.
- H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on information theory*, 29(4):551–559, 1983.
- S. Elghobashi. On predicting particle-laden turbulent flows. *Applied scientific research*, 52(4):309–329, 1994.
- P. Ern, F. Risso, D. Fabre, and J. Magnaudet. Wake-Induced Oscillatory Paths of Bodies Freely Rising or Falling in Fluids. *Annual Review of Fluid Mechanics*, 44(1):97–121, 2012. ISSN 0066-4189. doi: 10.1146/annurev-fluid-120710-101250. URL <http://www.annualreviews.org/doi/10.1146/annurev-fluid-120710-101250>.
- A. W. Fitzgibbon, M. Pilu, and R. B. Fisher. Direct least squares fitting of ellipses. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 1, pages 253–257. IEEE, 1996.
- A. Frohn and N. Roth. *Dynamics of droplets*. Springer Science & Business Media, 2000.
- R. Gieré and X. Querol. Solid particulate matter in the atmosphere. *Elements*, 6(4):215–222, 2010.
- G. Green. An essay on the application of mathematical analysis to the theories of electricity and magnetism, 1828.
- P. Gregory. Distribution of airborne pollen and spores and their long distance transport. *Pure and Applied Geophysics*, 116(2-3):309–315, 1978.
- R. Halir and J. Flusser. Numerically stable direct least squares fitting of ellipses. In *Proc. 6th International Conference in Central Europe on Computer Graphics and Visualization. WSCG*, volume 98, pages 125–132. Citeseer, 1998.
- Z. Han, W. Weng, and Q. Huang. Characterizations of particle size distribution of the droplets exhaled by sneeze. *Journal of the Royal Society Interface*, 10(88):20130560, 2013.

- J. Happel and H. Brenner. *Low Reynolds number hydrodynamics: with special applications to particulate media*, volume 1. Springer Science & Business Media, 1983.
- C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.
- D. M. Harris, T. Liu, and J. W. Bush. A low-cost, precise piezoelectric droplet-on-demand generator. *Experiments in Fluids*, 56(4):83, 2015.
- M. Hilchenbach, J. Kissel, Y. Langevin, C. Briois, H. Von Hoerner, A. Koch, R. Schulz, J. Silén, K. Altwegg, L. Colangeli, et al. Comet 67p/churyumov–gerasimenko: close-up on dust particle fragments. *The Astrophysical Journal Letters*, 816(2):L32, 2016.
- S. D. Hoath, editor. *Fundamentals of Inkjet Printing*. Wiley-VCH Verlag GmbH Co. KGaA, Weinheim, Germany, jan 2016a. ISBN 9783527684724. doi: 10.1002/9783527684724. URL <http://doi.wiley.com/10.1002/9783527684724>.
- S. D. Hoath. *Fundamentals of inkjet printing: the science of inkjet and droplets*. John Wiley & Sons, 2016b.
- A. Hölzer and M. Sommerfeld. New simple correlation formula for the drag coefficient of non-spherical particles. *Powder Technology*, 184(3):361–365, 2008.
- R. A. Houze Jr. *Cloud dynamics*. Academic press, 2014.
- J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.
- K. Jayaweera and B. Mason. The behaviour of freely falling cylinders and cones in a viscous fluid. *Journal of Fluid Mechanics*, 22(4):709–720, 1965.
- K.-H. Kim, E. Kabir, and S. Kabir. A review on the human health impact of airborne particulate matter. *Environment international*, 74:136–143, 2015.
- Y. J. Kim, S. Y. Kim, J. S. Lee, J. Hwang, and Y. J. Kim. On-demand electrohydrodynamic jetting with meniscus control by a piezoelectric actuator for ultra-fine patterns. *Journal of Micromechanics and Microengineering*, 19(10), 2009. ISSN 09601317. doi: 10.1088/0960-1317/19/10/107001.
- A. N. Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers. *Cr Acad. Sci. URSS*, 30:301–305, 1941.
- P. D. Komar and C. Reimers. Grain shape effects on settling rates. *The Journal of Geology*, 86(2):193–209, 1978.
- T. Koop, B. Luo, A. Tsias, and T. Peter. Water activity as the determinant for homogeneous ice nucleation in aqueous solutions. *Nature*, 406(6796):611–614, 2000.
- D. Kuhlman. *A python book: Beginning python, advanced python, and python exercises*. Dave Kuhlman Lutz, 2009.
- H. H. Lamb. Volcanic dust in the atmosphere; with a chronology and assessment of its meteorological significance. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 266(1178):425–533, 1970.
- J. Lee and C. Lee. Modification of particle-laden near-wall turbulence: Effect of stokes number. *Physics of Fluids*, 27(2):1–28, 2015. ISSN 10897666. doi: 10.1063/1.4908277.
- Y. Liu, O. Stein, J. H. Campbell, L. Jiang, N. Petta, and Y. Lu. Three-dimensional printing and deformation behavior of low-density target structures by two-photon polymerization. In *Nano-engineering: Fabrication, Properties, Optics, and Devices XIV*, volume 10354, page 103541U. International Society for Optics and Photonics, 2017.

- E. Loth. Drag of non-spherical solid particles of regular and irregular shape. *Powder Technology*, 182(3):342–353, 2008.
- G. V. Madhav and R. Chhabra. Drag on non-spherical particles in viscous fluids. *International Journal of Mineral Processing*, 43(1-2):15–29, 1995.
- T. Mannel, M. S. Bentley, R. Schmied, H. Jeszenszky, A. C. Levasseur-Regourd, and J. R. K. Torkar. Fractal cometary dust—a window into the early solar system. *Monthly Notices of the Royal Astronomical Society*, page stw2898, 2016.
- E. Marchildon, A. Clamen, and W. Gauvin. Drag and oscillatory motion of freely falling cylindrical particles. *The Canadian Journal of Chemical Engineering*, 42(4):178–182, 1964.
- D. C. Meisel, M. Diem, M. Deubel, F. Pérez-Willard, S. Linden, D. Gerthsen, K. Busch, and M. Wegener. Shrinkage Precompensation of Holographic Three-Dimensional Photonic-Crystal Templates. *Advanced Materials*, 18(22):2964–2968, nov 2006. ISSN 09359648. doi: 10.1002/adma.200600412. URL <http://doi.wiley.com/10.1002/adma.200600412>.
- S. Mishra, K. L. Barton, A. G. Alleyne, P. M. Ferreira, and J. A. Rogers. High-speed and drop-on-demand printing with a pulsed electrohydrodynamic jet. *Journal of Micromechanics and Microengineering*, 20(9), 2010. ISSN 09601317. doi: 10.1088/0960-1317/20/9/095026.
- J. A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.
- G. Nelson, R. A. Kirian, U. Weierstall, N. A. Zatsepin, T. Faragó, T. Baumbach, F. Wilde, F. B. Niesler, B. Zimmer, I. Ishigami, et al. Three-dimensional-printed gas dynamic virtual nozzles for x-ray laser sample delivery. *Optics express*, 24(11):11515–11530, 2016.
- J. Nocedal and S. J. Wright. *Numerical optimization*. 2006. ISBN 0387303030.
- D. Nurfiani and C. Bouvet de Maisonneuve. Furthering the investigation of eruption styles through quantitative shape analyses of volcanic ash particles. *Journal of Volcanology and Geothermal Research*, 354:102–114, 2018. ISSN 0377-0273. doi: <https://doi.org/10.1016/j.jvolgeores.2017.12.001>. URL <https://www.sciencedirect.com/science/article/pii/S0377027317303797>.
- J. O’Rourke. *Computational geometry in c*, cambridge univ. Press, Cambridge, 1994.
- J. M. Prospero, E. Bonatti, C. Schubert, and T. N. Carlson. Dust in the caribbean atmosphere traced to an african dust storm. *Earth and Planetary Science Letters*, 9(3):287–293, 1970.
- H. R. Pruppacher and J. D. Klett. Microphysics of clouds and precipitation. *Nature*, 284(5751): 88–88, 1980.
- A. Pumir and M. Wilkinson. Collisional Aggregation Due to Turbulence. *Annual Review of Condensed Matter Physics*, 7(1):141–170, 2016. ISSN 1947-5454. doi: 10.1146/annurev-conmatphys-031115-011538. URL <http://www.annualreviews.org/doi/10.1146/annurev-conmatphys-031115-011538>.
- J. Purto, A. Verch, P. Rogin, and R. Hensel. Improved development procedure to enhance the stability of microstructures created by two-photon polymerization. *Microelectronic Engineering*, 194(February):45–50, jul 2018. ISSN 01679317. doi: 10.1016/j.mee.2018.03.009. URL <https://doi.org/10.1016/j.mee.2018.03.009><https://linkinghub.elsevier.com/retrieve/pii/S0167931718301205>.
- S. Raschka and V. Mirjalili. *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd, 2019.
- L. Rayleigh. On the instability of jets. *Proceedings of the London mathematical society*, 1(1):4–13, 1878.


- A. Roy, R. J. Hamati, L. Tierney, D. L. Koch, and G. A. Voth. Inertial torques and a symmetry breaking orientational transition in the sedimentation of slender fibres. *Journal of Fluid Mechanics*, 875:576–596, 2019a.
- A. Roy, R. J. Hamati, L. Tierney, D. L. Koch, and G. A. Voth. Inertial torques and a symmetry breaking orientational transition in the sedimentation of slender fibres. *Journal of Fluid Mechanics*, 875:576–596, 2019b.
- O. Schlenczek. *Airborne and ground-based holographic measurement of hydrometeors in liquid-phase, mixed-phase and ice clouds*. PhD thesis, Max Planck Institute for Chemistry, Max Planck Society, 2018.
- J. Schneider and C. Hendricks. Source of uniform-sized liquid droplets. *Review of scientific Instruments*, 35(10):1349–1350, 1964.
- D. Sinclair and V. K. La Mer. Light scattering as a measure of particle size in aerosols. the production of monodisperse aerosols. *Chemical reviews*, 44(2):245–267, 1949.
- R. J. Sornek, R. Dobashi, and T. Hirano. Effect of turbulence on vaporization, mixing, and combustion of liquid-fuel sprays. *Combustion and flame*, 120(4):479–491, 2000.
- C. Steger. An unbiased detector of curvilinear structures. *IEEE Transactions on pattern analysis and machine intelligence*, 20(2):113–125, 1998.
- I. Steinke. Liquid Jets for Experiments on Complex Fluids. *Dissertation*. 2015.
- G. G. Stokes. *On the effect of the internal friction of fluids on the motion of pendulums*, volume 9. Pitt Press Cambridge, 1851.
- D. Terwagne. *Bouncing droplets, the role of deformations*. PhD thesis, Université de Liège, Liège, Belgique, 2011.
- D. Terwagne, F. Ludewig, N. Vandewalle, and S. Dorbolo. The role of the droplet deformations in the bouncing droplet dynamics. *Physics of Fluids*, 25(12):1–15, 2013. ISSN 10706631. doi: 10.1063/1.4832975.
- S. Thiele, K. Arzenbacher, T. Gissibl, H. Giessen, and A. M. Herkommer. 3d-printed eagle eye: Compound microlens system for foveated imaging. *Science advances*, 3(2):e1602655, 2017.
- F. Toschi and E. Bodenschatz. Lagrangian properties of particles in turbulence. *Annual review of fluid mechanics*, 41:375–404, 2009.
- G. A. Voth and A. Soldati. Anisotropic Particles in Turbulence. *Annual Review of Fluid Mechanics*, 49(1):249–276, 2016. ISSN 0066-4189. doi: 10.1146/annurev-fluid-010816-060135.
- J. B. Will, V. Mathai, S. G. Huisman, D. Lohse, C. Sun, and D. Krug. Kinematics and dynamics of freely rising ellipsoids at high Reynolds numbers. *arXiv*, pages 1–46, 2020. ISSN 23318422. doi: 10.1017/jfm.2020.1104.
- V. Wolf, T. Kuhn, M. Milz, P. Voelger, M. Krämer, and C. Rolf. Arctic ice clouds over northern sweden: microphysical properties studied with the balloon-borne ice cloud particle imager b-ici. *Atmospheric Chemistry and Physics*, 18(23):17371–17386, 2018. doi: 10.5194/acp-18-17371-2018. URL <https://acp.copernicus.org/articles/18/17371/2018/>.
- M. Wright. Direct search methods: Once scorned, now respectable. In *Numerical Analysis: Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis*, pages 191–208. Addison-Wesley, 1996.
- J. C. Yang, W. Chien, M. King, and W. L. Grosshandler. A simple piezoelectric droplet generator. *Experiments in fluids*, 23(5):445–447, 1997.
- X. Zhou, Y. Hou, and J. Lin. A review on the processing accuracy of two-photon polymerization. *AIP Advances*, 5(3):030701, 2015.

C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-BFGS-B. *ACM Transactions on Mathematical Software*, 23(4):550–560, dec 1997. ISSN 0098-3500. doi: 10.1145/279232.279236. URL <https://dl.acm.org/doi/10.1145/279232.279236>.

Appendix A

Data sheets

For more information, please visit the [product page](#).

	Part No: CEB-20D64	Date: 7/28/2006
	Description: piezo electric diaphragm	Unit: mm Page No: 1 of 4

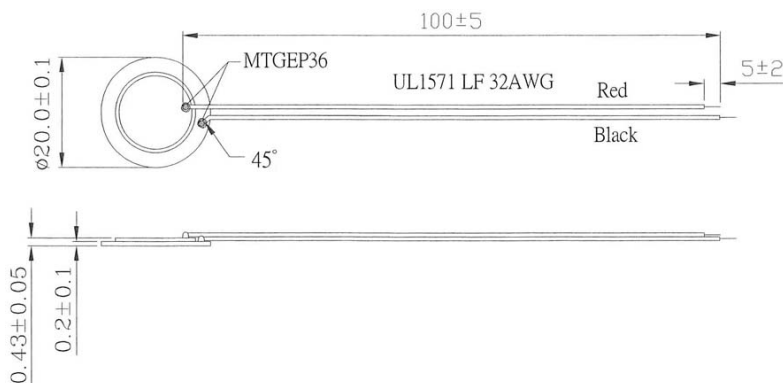


Specifications

Maximum input voltage	30 Vp-p	
Resonant frequency	6.5 ± 0.5 KHz	see Measurement Methods
Resonant impedance	350 Ω max.	see Measurement Methods
Electrostatic capacitance	13,000 ±30% pF	at 120 Hz / 1 V
Operating temperature	-20 ~ +70° C	
Storage temperature	-30 ~ +80° C	
Dimensions	Ø20.0 x H0.43 mm	
Weight	1.50 g max.	
Material	Brass	
Terminal	Wire type	
DC resistance	20 M Ω min.	Fluke 45 rate: Fast Measurement time: 1 second (only for ≤ 20 mm test)
RoHS	yes	

Appearance Drawing

Tolerance: ±0.5



For more information, please visit the [product page](#).



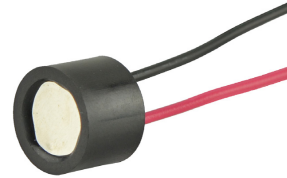
date 03/30/2016

page 1 of 4

MODEL: CPT-1495C-300 | **DESCRIPTION:** PIEZO BUZZER TRANSDUCER

FEATURES

- externally driven
- wire leads
- IP67 rated



SPECIFICATIONS

parameter	conditions/description	min	typ	max	units
operating voltage				30	Vp-p
current consumption	at 10 Vp-p, 4,250 Hz square wave			12	mA
rated frequency			4,250		Hz
sound pressure level	at 10 cm, 10 Vp-p, 4,250 Hz square wave	84			dB
electrostatic capacity	at 1,000 Hz/1 V	14,000	20,000	26,000	pF
dimensions	Ø14.0 x 10.0				mm
weight				2.60	g
material	PBT+15% Glass				
terminal	wire type				
operating temperature		-30		85	°C
storage temperature		-40		95	°C
RoHS	2011/65/EU				
IP level	IP67 (IEC standard 529 edition 2.0 (1989))				

Notes: 1. All specifications measured at 5~35°C, humidity at 45~85%, under 86~106 kPa pressure, unless otherwise noted.

SOLDERABILITY

parameter	conditions/description	min	typ	max	units
hand soldering	for maximum 5 seconds	330		380	°C

Appendix B

Insertion test 3

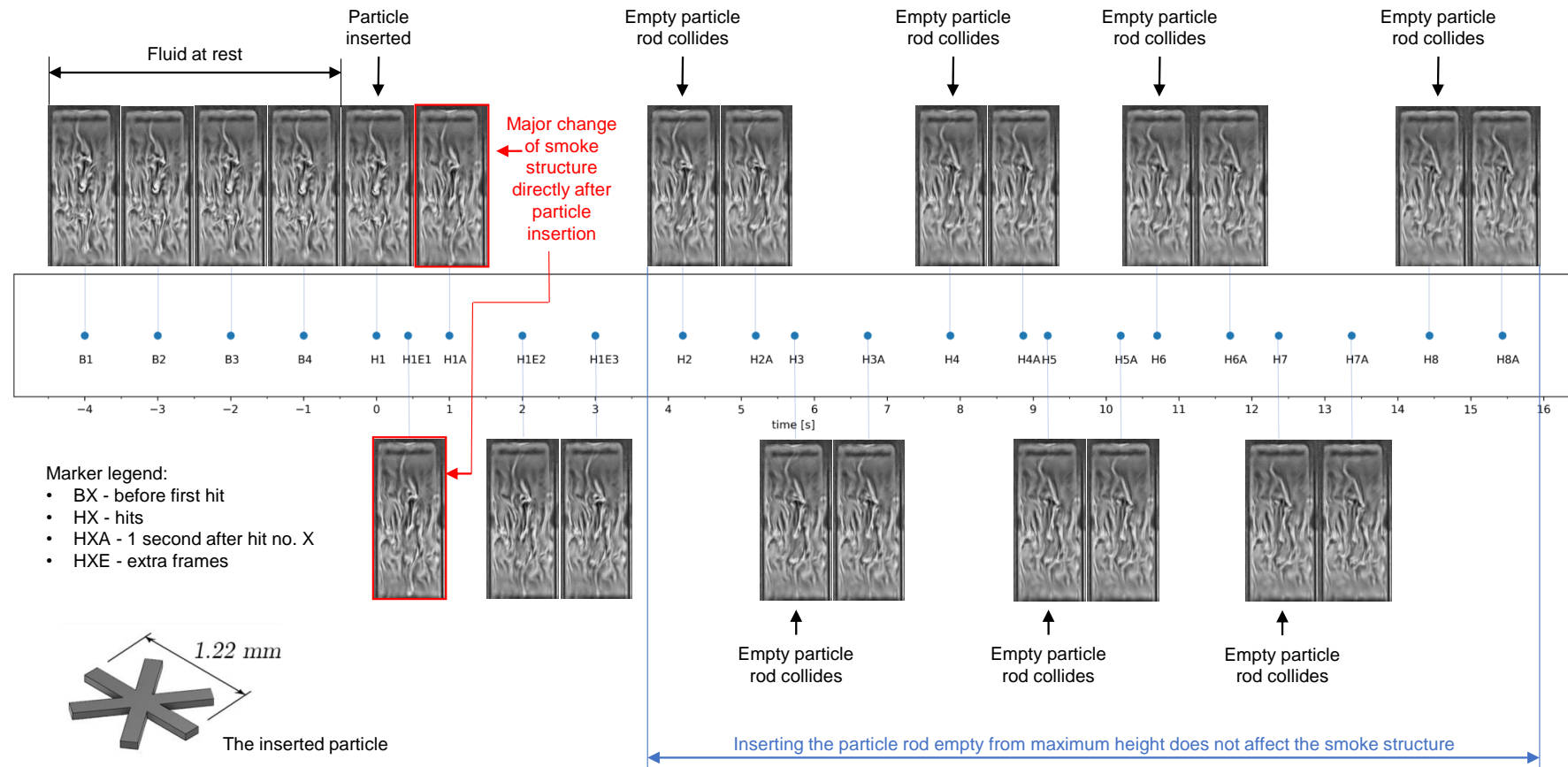


Figure B.1: Results of insertion test 3: Sealed drawer, big particle. Here, a large particle was released with the insertion mechanism. The particle had the shape of a six-legged cross with an outer diameter of 1.22 mm. Markers B1-B4 are time stamps 4, 3, 2 and 1 seconds before the first time the particle rod collides inside the insertion mechanism after falling freely from maximum height, which is marked with H1. The markers "H" (hit) mark collisions. The flow structure of the smoke is at rest during markers B1-B4. A large change of the flow structure is visible at markers H1E1 (433 ms after H1) and H1A (1 s after H1) compared to marker H1. This was caused by the particle passing through the smoke and affecting it. Subsequent collisions of the empty particle rod falling from maximum height (markers H2-H8 and markers H2A-H8A), do not display any change in the flow structure. This means that the insertion mechanism is working as intended and no relevant amount of flow is inserted into the observation volume during its operation, even when dropping the particle rod from the maximum height possible.