Dust position reconstruction using the particle swarm optimization algorithm and string formation in a dusty plasma in microgravity conditions

Amund Westgård Fredriksen

Thesis submitted for the degree of
Master in Physics
(Space Physics and Space Technology)
60 credits

Department of Physics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Autumn 2021

Dust position reconstruction using the particle swarm optimization algorithm and string formation in a dusty plasma in microgravity conditions

Amund Westgård Fredriksen

Dust position reconstruction using the particle swarm optimization algorithm and string formation in a dusty plasma in microgravity conditions

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

The particle swarm optimization (PSO) algorithm was applied to reconstructing three-dimensional positions of dust particles from images taken during dusty plasma experiments in microgravity conditions. It was compared to the 'Shake the Box' (STB) algorithm, which is a state-of-the-art algorithm for three-dimensional single particle position reconstruction. The PSO and STB algorithms were tested on artificial particle sets with realistic-looking images. On average, the PSO algorithm found a moderately higher fraction of particles than the STB algorithm. On the other hand, STB returned a somewhat lower fraction of ghost particles. STB has a significantly faster runtime compared to PSO, though PSO is still practical to use. The conclusion is that, as the current versions of the algorithms are, the PSO algorithm is the better choice for finding a higher number of the particle positions, and the STB algorithm is the better choice for returning fewer ghost particles and for faster runtime.

Using the PSO algorithm, data from dusty plasma experiments in parabolic flight campaigns were analyzed. Three-dimensional positions of individual particles were reconstructed, and dust strings were observed. The one-dimensional pair correlation function was calculated for the string-containing regions, and based on the pair correlation function the interparticle distance in the strings was estimated to be $(250 \pm 40)\,\mu\text{m}$. The mean interparticle distance was found to remain approximately constant in time over the frame range of the analyzed data, corresponding to 2.5 seconds. The effective Debye length of the plasma was calculated to be $\lambda_\text{D} = 276\,\mu\text{m}$ for a subsonic ion stream with Mach number $M = 0.75$.

Various forces on a dust particle in a string in the dilute midplane region of the plasma chamber were estimated. The neutral drag force was estimated to be $|F_\text{n}| = 3.023 \cdot 10^{-14}\,\text{N}$. The longitudinal ion drag force was estimated to be $|F_\text{i,long}| = 1.765 \cdot 10^{-12}\,\text{N}$ for $M = 0.75$. The longitudinal ion drag force is concluded to be the main driver of the initial acceleration of the dust particles in the dilute region. The transverse ion drag force and the longitudinal electrostatic repulsive force from adjacent dust particles in a string were estimated for a dust particle in a string. These forces were found to be in the order of $10^{-13}\,\text{N}$ for a displacement of $s = 0.1d$ in the transverse and longitudinal directions, respectively. The electrostatic force is proposed as a possible second contributing force for the stability of interparticle spacing in dust strings, in addition to the already established transverse ion drag force.

Acknowledgements

I would like to thank my supervisor Wojciech Miloch and co-supervisor Michael Himpel for all of their knowledge and support, and for all the discussions and constructive feedback along the way. I also want to thank Stefan Schütt for providing helpful information and insightful comments.

I would also like to express my appreciation for the 4DSpace team, both to the lecturers who helped us learn the basics of plasma and space physics, and also for hosting the many digital meetings during the lockdown that helped preserve our sanity during those times.

And of course, my amazing fellow master students: Aksel, Lise, Michael, Saida and Sean. Thank you for adding so much more fun to these last two years!

Contents

1 Introduction

It is believed that the ordinary matter in the observable universe (that is, matter, excluding the theorized dark matter) at this point in time contains at least 99% plasma [57]. This claim seems reasonable when considering that the Sun, which mainly consists of plasma, comprises 99.8% of the mass of our solar system [3], as well as the fact that both the Sun and our solar system are average compared to other stars and solar systems [3, 55]. A plasma is an ionized gas [1], meaning a gas where some fraction of the gas molecules has been ionized, so that there are free electrons and ions present in the gas. The gas can be ionized for example by high temperatures, strong electric fields or photoionization [64, p. 29]. The moving free charges influence each other through electromagnetic forces. The motion and properties of a gas can be described using fluid mechanics and thermal physics [86, 73], and the electromagnetic interactions caused by the free charges are governed by Maxwell's equations [32]. Both of these domains of physics go together to construct a physical theory of plasmas [22].

A *dusty plasma*, also known as a complex plasma, is a plasma with dust particles in it [57]. Dust particles are small objects of solid matter that can be composed of a wide range of materials [64, 82]. The size of the dust particles in interstellar clouds is typically between 10 nm and 10 µm in diameter [57, p. 2], and in laboratory experiments, e.g. studying dusty plasma crystallization, the dust particle diameters are usually in the range between 1 and 10 µm [57, p. 3], though also nanometer particles are studied in experiments [26]. Real-life examples of dusty plasmas include planetary rings, noctilucent clouds and comet tails [57, 82]. Another example is the plasma processing techniques used in the manufacturing of microchips, where the *removal* of dust particles in the plasma discharge is what is of importance, since dust particles can accumulate on the etched structures and cause malfunctioning in the chips [57].

Because the electrons in the plasma move faster than the ions due to their comparably low mass, on average more electrons than ions hit the dust particles per unit time. This results in a net negative charge on the dust particles, provided that the collection currents of electrons and ions is the dominant dust charging mechanism, which most often is the case [57]. This leads to an electrostatic repulsion between the dust particles. The free charges also lead to the formation of *electrostatic sheaths* around charged objects, including dust particles and electrodes, which shield the surrounding plasma from the electrostatic forces of those charged objects [64]. Phenomena that can be observed in dusty plasma systems include dust crystals [88, 43], Yukawa balls [49], dust density waves [12], modified ion acoustic waves [57], dust voids [31] and string formation [11, 44].

In order to study phenomena in dusty plasmas in detail, it is necessary to perform experiments and collect data, often in the form of images of the dust particles. The three-dimensional positions of the dust particles must be determined from these images through computational methods. A newly developed method for determining particle positions in three dimensions is a stereoscopic method that is based on the *particle swarm optimization* (PSO) algorithm. This particle detection method was developed and programmed by Michael Himpel of the University of Greifswald, and in this project I have further developed and optimized this method. After the development and optimization of the PSO method was completed, it was compared to a currently used position determination algorithm for three dimensions, the 'Shake the Box' (STB) algorithm [71, 35].

Using the newly developed PSO-based method for particle detection, I have analyzed images taken during previously performed microgravity dusty plasma experiments on parabolic flights.

[1] For a proper definition of a plasma there are a few more nuances that need to be considered. A more formal definition is presented in section 2.

I have studied the phenomenon of string formation, which is a phenomenon in dusty plasmas that can be observed under certain physical conditions in microgravity [11, 47]. Particle strings in microgravity dusty plasmas have been studied or observed before, but mostly in two dimensions [11, 47]. Dust strings in three dimensions has received some, but not very much, attention [35, 48], due to limited diagnostics for three-dimensional position reconstruction in the past. The investigations performed in this thesis provide better understanding of the three-dimensional picture of dust strings in complex plasmas.

2 Theory

This section contains the most important relevant theory of plasmas and dusty plasmas. The theory introduced here will lay the groundwork for better understanding the experimental setup, which is presented in section 3, and the observations and results from the parabolic flight experiments, presented in section 5.

First, some of the fundamentals of plasma physics will be introduced. After a short discussion of the concepts of electrostatic shielding and quasi-neutrality, a more rigorous definition of a plasma will be given in section 2.1.5. Then, dust particles will be introduced into the plasma, and some of the basic theory of dusty plasmas will be presented, along with some notable phenomena that can be observed in dusty plasmas. Further, the current knowledge in the literature on dust strings will be reviewed, and the pair correlation function is presented as a tool for studying dust strings. Lastly, the mathematical model of the PSO algorithm is presented, along with a pseudocode for its implementation.

A note on word use: It is worth explaining the usage of the term *plasma discharge*. It is a term that refers to the plasma itself, usually within some confined space such as a plasma chamber, which often comes up in textbooks and scientific articles on plasma physics. Initially, the term 'gas discharge' originated in relation to an invention from 1745 called the *Leyden jar*, a high-voltage capacitor, sounded like a gun shot whenever it produced a spark (an electric discharge) in air [64]. The word 'discharge' has since become a word for the plasma itself, the link being that electric sparks are essentially 'made out of' plasma in the sense that the light from the sparks comes from the de-excitation of gas atoms that were excited or ionized due to the high voltage. While an exact definition of the term was hard to find despite its common use in the literature, it seems to be used simply as a more concrete term for referring to a plasma in a confined space as an 'object' instead of using the slightly more ambiguous word 'plasma'. As an example, let us say that there are two boxes, both of them containing the same amount of the same identical plasma. Then one could say that there are two discharges, but you probably would not say that there are 'two plasmas', since that could imply that the boxes contain two different *kinds* of plasmas (two plasmas with different plasma parameter values).

2.1 Introduction to plasma physics

2.1.1 Ionization of gas molecules

A plasma is an ionized or partially ionized gas (only a fraction of the gas molecules need to be ionized for it to be called a plasma), so it is useful to have an understanding of how the ionization of gas atoms or molecules can occur. Ionization occurs when an electron is ejected from or collected into an orbital around an atom or molecule [20]. In general, electronegative plasmas (plasmas with mainly negative ions rather than positive ions) can exist [74], but electropositive plasmas (plasmas with mainly positive ions) are much more common and more studied [64], and it is the type of plasma relevant for this thesis. So, the focus here will be on conventional electropositive plasmas, where the ionization occurs by releasing electrons from their orbitals around the gas molecules.

The question to answer, then, is how the ejection of electrons can occur. There are several ways this can happen. When an electron is bound to an atom, it is located in a potential well [29]. In order to eject an electron from its potential well (orbital), energy is required [10]. This energy can be from e.g. collisions (kinetic energy), thermal energy in the gas molecules, or the energy of individual photons (photoelectric emission) [82]. In the experiments relevant for this thesis, the ionization of the gas in the plasma chamber is caused mainly by field ionization from an rapidly alternating (radiofrequency) applied electric field. Each released electron will

then be accelerated and can collide with another gas molecule and release an electron from that one, and so on, giving an initial exponential growth of new electron-ion pairs until an equilibrium density of free electrons and ions is reached [64, p. 326]. In addition, the externally applied voltage alters the potential well of an electron, resulting in less energy being required to eject the electron from its orbital [2] [25, 79], which in turn leads to a higher ionization rate.

2.1.2 Electrostatic shielding

Electrostatic shielding, also known as *electrostatic screening*, *Debye shielding* or simply *shielding* or *screening*, is the phenomenon in which free charges will rearrange themselves to counteract an electric field set up by a charged object [22, 64, 17, 67]. Since the free charges also all have their own electric fields, the surroundings are effectively shielded from the electric field caused by the central charged particle. Electrostatic shielding can also be understood from Gauss' law,

$$\Phi_E \equiv \oint_S \mathbf{E} \cdot d\mathbf{a} = \frac{q_{\text{encl},S}}{\epsilon_0}, \tag{2.1}$$

which states that the total electric flux Φ_E coming from particles within a certain closed surface S in space is proportional to the net charge $q_{\text{encl},S}$ of the particles within the volume enclosed by S [32]. As the free charges move closer to the charged object, the net charge around the object decreases since charges of opposite sign are accelerated toward the object, and charges of equal sign are accelerated away from the object. From Gauss' law we see that the electric field in the volume surrounding the charged object will therefore be closer to zero as well, so the surroundings are *shielded* from the electric field. Electrostatic shielding is a general phenomenon, and the charged objects that are screened by the plasma environment can for example be a small rock, a spaceship, or, most relevant for this thesis, dust particles.

The *Debye length* of a particle species s in a plasma, $\lambda_{D,s}$, is defined as the distance it takes for the electric potential ϕ from a charged object to decrease by a factor of e due to the electrostatic screening performed by the species s, where e is Euler's number [18]. In mathematical terms, $\phi(r = r_0 + \lambda_D) \equiv \phi(r = r_0)/e$, where r_0 is an arbitrary reference point outside of the charged object. This definition aims to provide a concrete length scale for the distance at which the electrostatic influence of a charged object in a plasma is, to an approximation, negligible. The mathematical expression for the screening length of a species s is

$$\lambda_{\text{D},s} \equiv \sqrt{\frac{\varepsilon_0 k_{\text{B}} T_s}{n_{s0} q_s^2}}, \tag{2.2}$$

where ε_0 is the vacuum permittivity, k_{B} is the Boltzmann constant, T_s is the average temperature of the species s, n_{s0} is the number density of s infinitely far away from the charged particle, and q_s is the charge of the species s [22, 64]. The species s can for example be electrons or ions. Both a higher density n_{s0} and a higher charge q_s give a shorter screening length, since either a higher availability of charges to come closer to the charged particle or a stronger charge on each individual free charge will boost the screening effect of that species.

[2] More precisely, less energy is required to eject an electron *in one direction*, towards the positive electrode. In the other direction, toward the negative electrode, *more* energy will be required to eject the electron. The applied voltage decreases the potential well on one side and increases the potential well on the other side, due to one electrode having a net positive charge and the other electrode having a net negative charge. Still, because all the free electrons will be accelerated toward the positive electrode at all times (whichever of the electrodes is currently positive), a larger fraction of electrons will collide with gas molecules in the 'weakened' direction of the potential wells, resulting in an increased ionization rate per electron-molecule collision. In an alternating voltage, the direction of the asymmetry of the potential wells alternates with it. [25, 79]

The temperature increases the screening length since it is basically the average kinetic energy of the s-particles, so a higher temperature means that the particles will also move away from the charged particle due to their kinetic energy. The effective Debye length, or the *linearized* Debye length [64, p. 38], of a plasma environment containing multiple species is defined as [57, p. 59]

$$\frac{1}{\lambda_{\mathrm{D,eff}}^2} \equiv \sum_s \frac{1}{\lambda_{\mathrm{D},s}^2}. \tag{2.3}$$

From eq. 2.3 we see that for each additional species s that contributes to the sum, the effective Debye length $\lambda_{\mathrm{D,eff}}$ decreases. This is reasonable, since the more free charges that are available for screening per unit volume within the plasma, the less volume (distance) will be required for screening to be achieved.

Streaming ions. It is possible for the ions to have a non-zero drift velocity u_i (the average velocity of all ions) in the plasma, a situation also known as *streaming ions* [57]. One can have subsonic or supersonic ion streams, where the sonic speed, also known as acoustic speed or sound speed, of ions in the case of $T_\mathrm{e} \gg T_\mathrm{i}$ is [67]

$$C_\mathrm{s} = \sqrt{\frac{k_\mathrm{B} T_\mathrm{e}}{m_\mathrm{i}}}. \tag{2.4}$$

An ion drift velocity of $u_\mathrm{i} = C_\mathrm{s}$ is equivalent to Mach 1, or Mach number $M = 1$. In the case of streaming ions, the ions become less able to screen charged objects in the plasma. This is because the ions have a momentum which must be cancelled in order to settle into a shielding position around the charged object, and a certain number of ions (depending on the ion drift speed) will simply move past the object due to their momentum. For this reason, eq. (2.3) will often not give a realistic screening length in a plasma that contains streaming ions. An expression for the characteristic screening length in a plasma containing streaming ions is given by Ludwig *et al.* (2012) [54], as

$$\lambda_\mathrm{s} = \lambda_{\mathrm{D,e}} \left(\frac{k_\mathrm{B} T_\mathrm{i} + \eta m_\mathrm{i} u_\mathrm{i}^2}{k_\mathrm{B} T_\mathrm{e} + k_\mathrm{B} T_\mathrm{i} + m_\mathrm{i} u_\mathrm{i}^2} \right)^{1/2}, \tag{2.5}$$

where $\lambda_{\mathrm{D,e}}$ is the electron Debye length given by eq. (2.2), m_i is the ion mass, u_i is the ion drift velocity and η is a fitting parameter. Ludwig *et al.* found that $\eta = 1.27$ produced a better fit with the screening lengths obtained from PIC simulations for Mach numbers $0 \leq M \lesssim 2$ when compared to the expression for the effective screening length used by Hutchinson [42, 57], which, for reference, was

$$\lambda_\mathrm{s} = \left(\frac{\lambda_{\mathrm{D,e}}^2}{1 + 2k_\mathrm{B} T_\mathrm{e}/(m_\mathrm{i} v_\mathrm{s}^2)} + a^2 \right)^{1/2}, \tag{2.6}$$

where $v_\mathrm{s} = (u_\mathrm{i}^2 + v_{\mathrm{th,i}}^2)^{1/2}$.

Note that the Debye length for a charged species only gives the (approximate) shielding length for an inserted charged object *without* any externally applied electric potential. If a voltage is applied, the shielding length will either increase or decrease depending on the applied potential (more precisely, the direction of the electric field caused by the applied potential) [57].

Electrostatic sheaths. An *electrostatic sheath*, or simply *sheath*, is a region of higher positive [3] charge density around (initially uncharged) objects or surfaces that are exposed directly to the plasma that arises as a consequence of electrostatic shielding [22]. The reason for why electrostatic sheaths arise in a plasma, and why they have a net positive charge density, will be explained here.

Since the electron mass is less than one is several thousands compared to the mass of an ion, each unit of kinetic energy that an electron receives will translate into a much greater speed compared to if an ion received the same amount of kinetic energy [19]. For this reason, electrons usually have a much higher thermal velocity than ions. This results in an initially higher collision rate of electrons onto an object in the plasma compared to the collision rate of ions onto the object. The collision rates of the charged particles onto the object are equivalent to electric currents, so there will initially be a higher electron current I_e onto the object compared to the ion current I_i. This results in a negative potential ϕ_obj on the object. As the electron and ion currents continue to charge the object, the potential becomes more negative, which in turn increases the repulsive force on the electrons and the attractive force on the ions with respect to the object. This influences the electron and ion currents, so the currents can therefore be expressed as $I_\mathrm{e} = I_\mathrm{e}(\phi_\mathrm{obj})$ and $I_\mathrm{i} = I_\mathrm{i}(\phi_\mathrm{obj})$.

At some point, the potential reaches an equilibrium where the ion and electron currents become equal in magnitude, and the total current becomes zero. The potential at which the currents sum to zero is defined as the *floating potential*, ϕ_fl, so we have $I_\mathrm{e}(\phi_\mathrm{fl}) + I_\mathrm{i}(\phi_\mathrm{fl}) \equiv 0$ [64, p. 169]. In a region close to the object, the ion number density n_i will be higher than the electron number density n_e due to the relatively larger degree of electron depletion, as well as the object's attraction of positive charges, and this region is what is defined as the sheath. The thickness of a sheath can be estimated e.g. using the Child-Langmuir law (in the case of a collisionless sheath in low pressures) [50, 62]. The sheath thickness of an object with no applied voltage has a thickness in the order of a few Debye lengths [22, p. 270].

Sheaths form on the surfaces of walls as well as around objects. Though the sheath region around a wall (or any other object) has a more positive charge density compared to the bulk plasma, it will still act as a barrier for negative species in the plasma, such as free electrons or negatively charged dust particles. The surrounding positive ions in the sheath *shield* the bulk plasma from the negative potential of the wall, but they do not completely *cancel* it, and they do not result in a net positive potential in the sheath. The potential is most negative close to the walls, and the least negative in the bulk of the plasma [22]. This results in a positive gradient in the electric potential from the wall toward the bulk plasma, which equivalently gives rise to a non-zero electric field in the sheath region, directed toward the wall, due to the general electrostatic equation

$$\mathbf{E}(\mathbf{r}) = -\nabla\phi(\mathbf{r}), \tag{2.7}$$

where $\mathbf{r} = (x, y, z)$ is a position in space, ϕ is the electric potential and $\mathbf{E}$ is the electric field [32]. This electric field is what repels the negative species from the walls in e.g. a plasma chamber.

Coulomb interaction between particles in a plasma: The Debye-Hückel potential As mentioned previously, because of the shielding of charged particles in the plasma, the electric potential around charged particles are diminished exponentially. The *Debye-Hückel potential*, also known as the *screened Coulomb potential* or the *Yukawa potential*, is the electric

[3] The sheath layer *can* also have a surplus negative charge density, in the case that the object has a positive potential bias [40]. For objects without electric potential bias however, which is the case for most objects (dust particles, walls, etc.), the sheath will be a layer containing a greater density of positive ions.

potential around a point particle or outside of a charged spherical object (such as a dust particle) that takes into account the shielding of the plasma. The Debye-Hückel potential produced by a charged object in a plasma is given by the following expression:

$$\phi_{\mathrm{DH}}(r) = \frac{Q}{4\pi\varepsilon_0 r} \exp\left(-\frac{r}{\lambda_D}\right), \tag{2.8}$$

where Q is the charge of the object, r is the distance from the center of the object and λ_{D} is the effective Debye length of the plasma [57].

An important implication that follows from electrostatic shielding will be discussed, namely quasi-neutrality.

2.1.3 Quasi-neutrality

The electrostatic shielding of individual charged particles or objects in a plasma also has an effect on the large-scale plasma. If we use a dust particle as an example of a charged object in the plasma, a distance $r \gg \lambda_D$ from the dust particle the electric potential caused by it will be cancelled out by the surrounding plasma due to electrostatic shielding. In fact, *any* charge disturbances or deviations from equal ion and electron densities will be cancelled over spatial scales longer than the Debye length [64, p. 38]. This is why it is said that a plasma has a tendency to become *quasi-neutral*, meaning electrically neutral on average over large spatial scales (spatial scales much larger than the Debye length), but not electrically neutral on small spatial scales (spatial scales equal to the Debye length or less) [7].

2.1.4 Plasma frequency

If a displacement from a uniform distribution of ions and electrons occurs in the plasma, restoring electric forces will arise between the free charges in the plasma. Due to inertia, free charges that are accelerated by these electric forces will overshoot the equilibrium positions, which leads to the displacement appearing again, and therefore also the corresponding restoring forces of the free charges back towards the equilibrium positions [22, p. 78] [64, p. 39]. This leads to an oscillatory movement of the free charges in the plasma. Each species in the plasma has a characteristic response time or frequency of this oscillation, and this frequency is named the *plasma frequency*. The plasma frequency of a species s is given by

$$\omega_{\mathrm{p}s} = \sqrt{\frac{n_{s0} q_s^2}{\varepsilon_0 m_s}}, \tag{2.9}$$

where q_s is the charge of a particle of species s, m_s is the mass of a particle of species s and n_{s0} is the background number density of s [14]. The units of $\omega_{\mathrm{p}s}$ are radians per second. The symbol $f_{\mathrm{p}s} \equiv \omega_{\mathrm{p}s}/(2\pi)$ is usually used for expressing the plasma frequency in units *revolutions* per second, or hertz (Hz). Different plasma frequencies for different species is equivalent to different *timescales* for the different species. A higher plasma frequency is equivalent to a faster response time and therefore events happening at shorter timescales, and vice versa.

As will be calculated in the following example, when comparing the plasma frequencies of electrons, ions and typical dust particles, electrons have a much higher plasma frequency than ions, and ions have a much higher plasma frequency than dust particles. Equivalently, phenomena involving dust particles happen much slower than the movements of ions and electrons, and phenomena involving ions happen much slower than the movements of electrons.

Example: Plasma frequency values for different species in a plasma
Plasma parameter values from the radiofrequency dusty plasma system studied in [35] are used here. Electron and ion number densities: $n_{e0} = n_{i0} = 10^{15}\,\mathrm{m}^{-3}$. Electron mass: $m_e = 9.11 \cdot 10^{-31}\,\mathrm{kg}$. Ion mass (argon atoms): $m_i = 39.95u \approx 6.63 \cdot 10^{-26}\,\mathrm{kg}$ [34]. Electron and ion charges: $q_i^2 = q_e^2 = e^2$. Dust particle number density: $n_{d0} = 10^{11}\,\mathrm{m}^{-3}$. Assuming that the dust grains are spherical grains of melamine formaldehyde, with mass density $\rho_d = 1.5\,\mathrm{g/cm}^3 = 1500\,\mathrm{kg/m}^3$ [44] and radius $a = 5\,\mu\mathrm{m}$, the mass of a dust grain will be $m_d = \rho_d \cdot \frac{4}{3}\pi a^3 \approx 7.85 \cdot 10^{-13}\,\mathrm{kg}$. Dust charge: $q_d = 7000e$. Using these values and eq. (2.9), the plasma frequencies for electrons, ions and dust particles, respectively, are the following:

- $\omega_{pe} = 1.78 \cdot 10^9\,\mathrm{rad/s} \quad \Leftrightarrow \quad f_{pe} = 283 \cdot 10^6\,\mathrm{s}^{-1} = 283\,\mathrm{MHz}$,
- $\omega_{pi} = 6.60 \cdot 10^6\,\mathrm{rad/s} \quad \Leftrightarrow \quad f_{pi} = 1.05 \cdot 10^6\,\mathrm{s}^{-1} = 1.05\,\mathrm{MHz}$,
- $\omega_{pd} = 537\,\mathrm{rad/s} \quad \Leftrightarrow \quad f_{pd} = 85.5\,\mathrm{Hz}$.

Relative to the ion plasma frequency we see that this is equivalent to $\omega_{pe} \approx 270\omega_{pi}$ and $\omega_{pd} \approx 8 \cdot 10^{-5}\omega_{pi}$, so we have $\omega_{pe} \gg \omega_{pi} \gg \omega_{pd}$.

2.1.5 A more formal definition of a plasma

We have now established all the necessary concepts to construct a more formal definition of a plasma. As stated previously, a plasma is an ionized or partially ionized gas. In addition to this, the more formal definition of a plasma also contains the following requirements [6, 22]:

- The physical size of the ionized gas is much larger than the Debye screening length.
- The ionized gas is quasi-neutral.
- The collision frequency between charged particles (electrons and ions) and neutrals (neutral gas atoms/molecules) is small compared to their respective plasma frequencies.

Some more elaboration on these points follows. The physical size of the plasma must be much larger than the Debye screening length for two reasons: 1) It allows for quasi-neutrality, which by definition is electrical neutrality over spatial scales much larger than the Debye length, and 2) it makes the interactions in the bulk of the plasma more important than the boundary effects that can occur at the edges of the plasma [6].

The requirement of quasi-neutrality implies that there must be enough free charged particles in the ionized gas so that a charged object that is placed inside the gas will be shielded. This means that the total number density of charged particles in the gas, n, must be sufficiently high. More specifically, the number of particles N_D within a 'Debye sphere' must be large. A Debye sphere is the volume of a sphere with radius equal to the Debye length λ_D,

$$N_D = \frac{4}{3}\pi\lambda_D^3 n, \tag{2.10}$$

and one of the formal requirements to have proper shielding is that this number must be much greater than 1. N_D is sometimes referred to as *the plasma parameter* [22].

The requirement for the collision frequency between charged particles and neutrals compared to the plasma frequencies makes sure that plasma as a state of matter is noticeably influenced by electromagnetic forces. If the charged particles in an ionized gas undergo too frequent collision with neutral gas atoms, the motion of the charges are not mainly governed by electromagnetic forces, but by the forces of ordinary fluid dynamics. An example of an ionized

gas that is not considered a plasma for this exact reason is the weakly ionized gas that comes out of the exhaust of a running jet engine [22, p. 11].

Averaging over the various species in the plasma, if f is the typical frequency of plasma oscillations and τ is the average time between collision between charged particles and neutrals, then the average number of plasma oscillations between each charge-neutral collision is $f\tau$. A quantitative requirement we can choose for defining a plasma is $f\tau > 1$ or $f\tau \gg 1$, that is, that the charged species are on average able oscillate multiple times between each collision with a neutral [22, p. 11].

If all of the above conditions are met for an ionized gas, then it is called a plasma.

2.2 Radiofrequency plasmas

A radiofrequency (RF) plasma is a plasma that is created by placing a gas (e.g. argon gas) between two electrodes that are producing an alternating voltage that is alternating with a frequency in the radiofrequency range [21]. In other words, the gas inside the plasma chamber will be exposed to a rapidly alternating electric field, which is the mechanism that drives the ionization of the gas in an RF plasma, as explained in section 2.1.1. The plasma in the experiments analyzed in this thesis is an RF plasma with frequency $f_{\mathrm{RF}} = 13.56\,\mathrm{MHz}$, which is a much used alternating voltage frequency in RF plasma experiments for the past few decades [35, 11, 57, 76, 64].

The reason for having the voltage alternate with a frequency in the radiofrequency range is that this frequency range can [4] oscillate significantly faster than the plasma frequencies of the ions or other more massive species, while still oscillating slower than the electron plasma frequencies [21]. Forces and waves involving ions or dust particles experience only negligible accelerations from the alternating voltage. However, the electrons *will* be able to follow the varying electrode potential due to their high plasma frequency [69]. This claim is supported by comparing the RF plasma frequencies calculated in the example in section 2.1.4 with the RF voltage: $f_{\mathrm{pe}} = 283\,\mathrm{MHz} \gg f_{\mathrm{RF}} = 13.56\,\mathrm{MHz} \gg f_{\mathrm{pi}} = 1.05\,\mathrm{MHz} \gg f_{\mathrm{pd}}$. As the electrons are accelerated by the alternating electric field between the electrodes, they can collide with neutral gas atoms and ionize them, as described in section 2.1.1.

RF sheaths: Though the sheaths around the electrodes expand and contract with the alternating RF voltage, each electrode still has an average sheath region with a net positive charge density [78]. As previously mentioned, for typical RF plasmas the RF frequency is too high for the ions to respond to the alternating voltage but low enough so that electrons will respond. The details of the dynamics of RF sheaths requires kinetic theory or particle simulations. The results from such simulations show that the sheath thickness around electrode 1 expands at the same rate as the sheath thickness around electrode 2 shrinks, and vice versa when the alternating voltage switches direction, keeping the bulk plasma at a constant width oscillating back and forth between the two electrodes [64, p. 339]. Each of the electrodes' sheath edges oscillates sinusoidally around an equilibrium thickness, which is the time-averaged RF sheath thickness. Similarly to the regular electrostatic sheaths discussed in section 2.1.2, the time-averaged RF sheaths exert confining forces to keep negatively charged species in the bulk plasma, including negatively charged dust particles, away from the electrodes.

[4] Depending on the specific value of the alternation frequency and on the plasma frequencies of the different species in the plasma. The radiofrequency range is (roughly) defined to be between 3 kHz and 300 GHz [9]. However, usually the frequencies that are used in RF plasmas are in the megahertz range [21].

2.3 Dusty plasmas

When dust particles are added to a plasma environment, they will become charged due to collection of free charges in the plasma [57]. This makes the dust particles and the plasma environment interact through electromagnetic forces, giving rise to phenomena unique to dusty plasma systems, some of which will be discussed below. In this subsection, some of the basic theory of dusty plasmas will be presented, including the charging of dust particles and various forces that act on the dust particles in the plasma. Magnetic fields in dusty plasmas and ion wakefields will also be discussed.

2.3.1 Charging of dust particles

From the discussions in section 2.1.2, objects in a plasma will on average accumulate a negative charge due to the higher electron speed compared to the ion speed, and this is true for dust particles as well. Here, the charging of dust particles will be discussed in some more detail.

A simple model for particle charging, namely the 'orbital motion limit' (OML) model, will be presented. The OML model models the charging of a dust particle as the current of electrons and the current of ions onto the dust particle, and it assumes that the electrons and ions move from infinitely far away toward the dust particle on a collisionless path, each charge only affected by the electrostatic interaction between itself and the dust particle [57]. It is also assumed that the velocity distribution of the charges is given by an isotropic Maxwellian distribution function. Note that the OML assumptions more or less break down in real plasma discharges: The paths of the charges are not without collisions, and the velocity distribution is realistically more complex than an ideal isotropic Maxwellian. However, the model serves as a rough approximation for what charge we can expect dust particles to have, especially in many laboratory experiments where the currents from incoming electrons and ions are the main charging mechanisms.

An outline of the derivation of the equilibrium dust charge will be presented here. Many of the mathematical details of the derivation of the dust charge will be skipped here; more details can be found in Melzer (2019) [57]. The collisionlessness assumed in OML gives conservation of angular momentum between the incoming charges and the dust particle. There is also conservation of total energy for the charged particles (kinetic energy + electric potential energy). From this, the collection cross-sections of the electrons and ions can be obtained:

$$\sigma_{\mathrm{c,i}} = \cdots = \pi a^2 \left(1 - \frac{2e\phi_p}{m_{\mathrm{i}} v_{\mathrm{i},0}^2}\right) \quad \text{(ions)} \tag{2.11}$$

$$\sigma_{\mathrm{c,e}} = \cdots = \pi a^2 \left(1 + \frac{2e\phi_p}{m_{\mathrm{e}} v_{\mathrm{e},0}^2}\right) \quad \text{(electrons)}. \tag{2.12}$$

Since the (average) dust particle is negatively charged, it also has a negative electric potential, $\phi_p < 0$, compared to the surroundings infinitely far away (plasma potential), which we define here as zero potential. Since the potential around the dust particle is negative, we see from eqs. (2.11) and (2.12) that the ion collection cross sections will be larger than the dust particle cross section a, and the electron cross section is less than the dust particle cross section. This is because of the electric force; the positively charged ions will be attracted to the negatively charged dust particle, so ions that are just outside of collision course will be accelerated toward the dust particle and collide with it, and similarly some electrons just within collision course with the dust particle will be accelerated away and avoid collision. We also see that for both ions and electrons, the greater speed they have the smaller the second term is, so for

fast-moving ions/electrons the electrostatic potential of the dust particle does not affect the collection onto the dust particle as much.

The expressions for the collection cross sections are used for finding the ion and electron currents to the dust particle. The infinitesimal contribution to the current of a charged species s (either ions or electrons) dI_s from an infinitesimal velocity range (the speed of those charged particles) between v_s and $v_s + dv_s$, is given by

$$dI_s = \sigma_{\mathrm{c},s}(v_s)\, dj_s = \pm n_s e v_s f(v_s)\, dv_s, \tag{2.13}$$

where the plus sign is for ions with charge $+1e$ and the minus sign is for electrons (charge $-e$). The function $f(v_s)$ is the velocity distribution function, which gives the weighting of how much each velocity v_s contributes to the current dI_s. We will assume that this distribution function is an isotropic Maxwellian [57], which has the following form:

$$f_{\mathrm{M}}(v_s) = 4\pi v_s^2 \left(\frac{m_s}{2\pi k_B T_s}^{3/2}\right) \exp\left(-\frac{\frac{1}{2}m_s v_s^2}{k_B T_s}\right), \tag{2.14}$$

where we have $\int_0^\infty f_{\mathrm{M}}(v_s)\, dv_s = 1$. We can observe in the expression that $f(v \to 0) = f(v \to \infty) = 0$, meaning that very slow or very fast particles are unlikely.

The full current is then the sum of the contributions from all of the infinitesimal velocity intervals, which is the following integral:

$$I_s = \int dI_s = \int_0^\infty \pm n_s e v_s f_{\mathrm{M}}(v_s)\, dv_s. \tag{2.15}$$

Calculating the currents for the ions and for the electrons, the results are the following:

$$I_{\mathrm{i}} = \cdots = \pi a^2 n_{\mathrm{i}} e \sqrt{\frac{8k_B T_{\mathrm{i}}}{\pi m_{\mathrm{i}}}} \left(1 - \frac{e\phi_{\mathrm{p}}}{k_B T_{\mathrm{i}}}\right), \tag{2.16}$$

$$I_{\mathrm{e}} = \cdots = -\pi a^2 n_{\mathrm{e}} e \sqrt{\frac{8k_B T_{\mathrm{e}}}{\pi m_{\mathrm{e}}}} \exp\left(\frac{e\phi_{\mathrm{p}}}{k_B T_{\mathrm{e}}}\right). \tag{2.17}$$

Now we are in a position to determine the dust particle charge. The dust particle charge will be defined as the charge it will have when it has reached an equilibrium in the incoming currents. Mathematically speaking, this is when the sum of the ion and electron currents is zero, and the electric potential of the dust particle for that circumstance is, by definition, the *floating potential* ϕ_{fl}. By solving the equation $I_{\mathrm{i}}(\phi_{\mathrm{p}}) + I_{\mathrm{e}}(\phi_{\mathrm{p}}) = 0$ for the potential, the floating potential is obtained. The equation for the floating potential can be solved numerically [57]. To give an example (retrieved from [57, p. 18]): If we have the parameter values $a = 4.7\,\mu\mathrm{m}$, $n_{\mathrm{i}} = n_{\mathrm{e}} = 10^9\,\mathrm{cm}^{-3}$, $T_{\mathrm{e}} = 3\,\mathrm{eV}$ and $T_{\mathrm{i}} = 0.03\,\mathrm{eV}$, the floating potential is $\phi_{\mathrm{fl}} \approx -5\,\mathrm{V}$. A rule-of-thumb approximation for laboratory plasmas with $T_{\mathrm{e}} \gg T_{\mathrm{i}}$ is that

$$\phi_{\mathrm{fl}} = -\alpha k_{\mathrm{B}} T_{\mathrm{e}}/e \tag{2.18}$$

where α is a number between 2 and 2.5 that depends on the species of the plasma gas and the temperature ratio between the electrons and ions, $T_{\mathrm{e}}/T_{\mathrm{i}}$. For example, for an argon gas with $T_{\mathrm{e}}/T_{\mathrm{i}} = 100$ we have $\alpha = 2.414$ [57].

Finding the floating potential of a dust particle allows us to find an estimate for the charge of the dust particle. By modelling the dust particle as a spherical capacitor, the charge on the dust particle is given by

$$Q_{\mathrm{d}} = C\phi_{\mathrm{fl}}. \tag{2.19}$$

In a plasma environment with shielding length λ_{D}, the capacitance of a sphere is

$$C = 4\pi\varepsilon_0 a \left(1 + \frac{a}{\lambda_{\mathrm{D}}}\right), \tag{2.20}$$

which usually reduces to $C = 4\pi\varepsilon_0 a$ (the vacuum case) in the case of $a \ll \lambda_{\mathrm{D}}$ [57], which is the case in the experiments relevant for this thesis [35]. Using eqs. (2.19) and (2.20) and the rule-of-thumb expression for ϕ_{fl}, the final result for the equilibrium dust charge is

$$Q_{\mathrm{d}} = -(1400 a_{\mu\mathrm{m}} T_{\mathrm{e,eV}})e, \tag{2.21}$$

where $a_{\mu\mathrm{m}}$ is the radius of the particle expressed as the number of micrometers, and $T_{\mathrm{e,eV}}$ is the electron temperature expressed as the number of electron volts [57].

It bears repeating that the expression in eq. (2.21) is based on the OML case with the assumptions of collisionless particle paths and ideal isotropic Maxwellian velocity distributions, so it is only an approximation. With that in mind, here is an explicit example: For a dust particle with radius $a = 5\,\mu\mathrm{m}$ in a plasma with electron temperature $T_{\mathrm{e}} = 4\,\mathrm{eV}$ (these values are in the realistic range, see [11]), this model predicts a dust charge of $Q_{\mathrm{d}} = -28\,000e$.

2.3.2 Forces on the dust particles

In this subsection the most important forces acting on dust particles as well as their strengths under different physical conditions will be reviewed. The sum of these forces on any given dust particle at a given point in time and space will determine its acceleration, and therefore its movement. Therefore, knowledge about the forces acting on the dust particles gives important information for understanding the dynamics of the observed dust particles.

Gravity

The gravitational force on a dust particle is given by

$$\mathbf{F}_{\mathrm{g}} = m_{\mathrm{d}}\mathbf{g} = \frac{4}{3}\pi a^3 \rho_{\mathrm{d}}\mathbf{g} \tag{2.22}$$

for a spherical dust particle with radius a and mass density ρ_{d} [57, 17]. Since the gravitational force is dependent on the volume of the grain dust, it is one of the dominating forces for micrometer-radius dust particles when a gravitational acceleration is present, and it becomes negligible for nanometer-sized particles.

Microgravity conditions are conditions where the net acceleration vector due to gravity, $\mathbf{g}$, in the relevant frame of reference is approximately zero. This can be for instance in outer space far away from stars and planets, where there actually is no gravitational acceleration (or, it is negligible). The other way to achieve microgravity is for the reference system to accelerate in a manner that is equivalent to a free fall in the gravitational field without air drag. Drop towers, parabolic flights and the International Space Station (ISS) are examples where this type of microgravity is achieved. On the ISS there is currently an ongoing zero-gravity dusty plasma experiment [80, 47]. The experiments that were analyzed in this thesis were microgravity experiments performed on parabolic flight airplanes. On parabolic flights, the pilots of the airplane steer the plane to follow the path of a parabola, which is the equivalent of a free fall without air drag close to the surface of a planet, to achieve microgravity inside the airplane [1].

During parabolic flights experiments, there is usually some residual acceleration due to slight deviations from parabolic trajectories. For this reason, an initial inspection of the acceleration data is done in order to make sure that the data that is analyzed has a sufficiently good microgravity quality [53] (see also section 5.2).

Electric field forces

The electric field force or electrostatic force is perhaps the most notable of the forces acting on a dust grain in a plasma. The charged dust particles will experience an electric force if there is an electric field present. Electric fields can come from other charged dust particles, or they can be externally applied electric fields from electrodes. Although the surrounding ions in the plasma will shield the plasma from the electric field coming from the dust particle from the free charged particles in the surrounding plasma, if there exists an external electric field at the location of the dust particle then the dust particle will be accelerated by the electric field with the same force as if there was no surrounding plasma [33]. The electrostatic force on a dust particle with charge Q_d in an electric field $\mathbf{E}$ is given by [24, 17]

$$\mathbf{F}_\mathrm{es} = Q_\mathrm{d}\mathbf{E}. \tag{2.23}$$

This equation assumes a uniform plasma, meaning that the ion and electron densities are equal everywhere in the system. The electric field $\mathbf{E} = \mathbf{E}(\mathbf{r}, t)$ is the sum of all contributing electric fields present at the position $\mathbf{r}$ at the time t, which can for example be the electric field from a sheath region or from surrounding dust particles.

Although the electrostatic shielding by the free electrons and ions will diminish any external electric field imposed by e.g. an electrode, there will still be an electric field in the sheath region near the electrode which confines the dust particles to the bulk plasma, as mentioned in section 2.1.2.

The electrostatic force between dust particles, which is a repulsive force since the dust particles will typically be negatively charged, is the main force that creates the crystal patterns that can be observed in certain physical circumstances (along with some confining forces such as gravity and/or externally applied electric fields) [57]. For spherical dust particles, the electrostatic interaction force can be calculated by using the Debye-Hückel potential ϕ_DH (eq. (2.8)). From eq. (2.7), along with the general identity

$$\mathbf{F}_\mathrm{es} = q\mathbf{E} \tag{2.24}$$

for the electrostatic force on an object in an electric field $\mathbf{E}$, the Coulomb force acted on object 1 by object 2 (spherical, charged objects) with a distance $r \equiv |\mathbf{r}_1 - \mathbf{r}_2|$ between them is

$$\begin{aligned}
\mathbf{F}_\mathrm{es}(\mathbf{r}_1, \mathbf{r}_2) &= Q_1\mathbf{E}(\mathbf{r}_1, \mathbf{r}_2, Q_2) \\
&= -Q_1\,\nabla\phi_\mathrm{DH}(r, Q_2) \\
&= -Q_1\,\nabla\left(\frac{Q_2}{4\pi\varepsilon_0 r}\exp\left(-\frac{r}{\lambda_D}\right)\right) \\
&\;\;\vdots \\
\mathbf{F}_\mathrm{es}(\mathbf{r}_1, \mathbf{r}_2) &= \frac{Q_1Q_2}{4\pi\varepsilon_0 r}\left(\frac{1}{r} + \frac{1}{\lambda_\mathrm{D}}\right)\exp\left(-\frac{r}{\lambda_D}\right)\mathbf{e}_r,
\end{aligned} \tag{2.25}$$

where $\mathbf{e}_r \equiv (\mathbf{r}_1 - \mathbf{r}_2)/r$ is the unit vector in the radial direction for the distance between the two objects in spherical coordinates.

In some cases, the electric field $\mathbf{E}$ can be such that it causes non-uniform plasma densities. If $\mathbf{E}$ is an ambipolar field, or if the dust particle is in the pre-sheath region of an electrode or a wall, then the plasma densities will be non-uniform [17]. For a non-uniform plasma the Debye length can be different at different points in space, meaning that the gradient of the shielding length, $\nabla\lambda_D$, can be non-zero. This non-uniformity sets up another electric force, called the *polarization force*, which is given by

$$\mathbf{F}_{\text{pol}} = -\frac{Q_{\text{d}}^2}{8\pi\epsilon_0}\frac{\nabla\lambda_{\text{D}}}{(\lambda_{\text{D}}+a)^2}, \tag{2.26}$$

which is always in the direction of decreasing shielding length [33, 57]. The total electric force is then

$$\mathbf{F}_{\text{E}} = \mathbf{F}_{\text{es}} + \mathbf{F}_{\text{pol}}. \tag{2.27}$$

The relevant shielding length λ_{D} can be closer to the ion shielding length $\lambda_{\text{D,i}}$ or the electron shielding length $\lambda_{\text{D,e}}$ depending on the ion drift velocity u_{i}, as seen in eq. (2.5).

Neutral drag force

Another force that can act on the dust particles is the neutral drag force, which is a friction force caused by collisions between dust particles and neutral gas atoms and molecules in the plasma environment. The force comes from the momentum transfer to the dust particles from these collisions [57, 17].

A general drag force is given by the equation

$$\mathbf{F}_{\text{drag}} = \frac{dN}{dt}\Delta\mathbf{p} = \Delta p\, n\sigma\mathbf{v}_{\text{rel}}, \tag{2.28}$$

where dN is the number of particles interacting with the dust particle during an infinitesimal time dt, σ is the cross sectional area of interaction with the dust particle, n is the number density of the species causing the drag force, Δp is the average momentum transfer in a collision with the dust particle, and $\mathbf{v}_{\text{rel}}$ is the average relative velocity between the dust particle and the species causing the drag force [57, p. 35].

The neutral drag force was derived by Epstein in 1924 [28]. An expression for the neutral drag force can, based on eq. (2.28), be shown [57, 17] to be

$$\mathbf{F}_{\text{n}} = -\delta\frac{4}{3}\pi a^2 m_{\text{n}} v_{\text{th,n}} n_{\text{n}} \mathbf{v}_{\text{d}}. \tag{2.29}$$

Here, the cross section σ is simply the cross sectional area πa^2 of the dust particle, since the neutrals are free of charge and therefore experience no electric force from the dust particle. The parameter δ, sometimes called the (Epstein) friction coefficient, takes into account how the neutrals are reflected from the surface of the dust particle. For dust particles that are plastic microspheres made of melamine formaldehyde (MF), which are used in the experiments relevant for this thesis, a value of $\delta = 1.44$ has been found [45].

A second expression for the neutral drag force derived by Epstein is in the form of a friction force,

$$\mathbf{F}_{\text{n}} = -m_{\text{d}}\beta\mathbf{v}_{\text{d}}, \tag{2.30}$$

with

$$\beta \equiv \delta\frac{8}{\pi}\frac{p}{a\rho_{\text{d}}v_{\text{th,n}}}, \tag{2.31}$$

where ρ_{d} is the mass density of the material that the spherical dust particle is made of, δ is the same Epstein friction coefficient as above and p is the gas pressure. Make sure not to confuse the usage of the symbol p; it is the symbol used to represent both momentum and pressure in different contexts.

It turns out that eq. (2.31) uses the assumption of an ideal gas for the neutrals, that is, $p = n_\mathrm{n} k_\mathrm{B} T_\mathrm{n}$. I encountered this fact when calculating the neutral drag force with eq. (2.29) using the assumption of an ideal gas for the neutrals, and then using eq. (2.30), and ending up with identical answers. In the references that I read, I could not find an explicit mention of this assumption in the Epstein neutral drag equations. For this reason, a proof of the underlying ideal gas assumption in eq. (2.30) is included in appendix B. The ideal gas assumption for the neutrals is discussed briefly in section 5.5.

The momentum transfer Δp in a collision with neutrals is in the order of $m_\mathrm{n} v_\mathrm{th,n}$, where m_n is the mass of a neutral particle and $v_\mathrm{th,n}$ is the (mean) thermal velocity of the neutral species. The mean thermal velocity, or simply thermal velocity [5], for a species s in a gas in three dimensions is defined as the mean value of the speed distribution [64, p. 75] [57],

$$v_{\mathrm{th},s} \equiv \int_0^\infty f_\mathrm{M}(v_s)\, v_s \, dv_s = \cdots = \sqrt{\frac{8 k_\mathrm{B} T_s}{\pi m_s}}, \tag{2.32}$$

where $f_\mathrm{M}(v_s)$ is the Maxwellian velocity distribution as shown in eq. (2.14), T_s is the temperature of s and m_s is the mass of a particle of species s.

Ion drag force

The ion drag force is the force caused by momentum transfer between the dust particle and ions streaming into or past the dust particle [57, p. 35]. The momentum transfer between an ion and a dust particle can come from a direct physical collision or from the Coulomb force between them.

If there exists a stream of ions in the plasma discharge, then the influence of the ion drag force on the dust particles can be prominent. The ion drag force together with the electrostatic force is the main force balance that establishes voids inside radiofrequency dusty plasma discharges [31].

The force resulting from the collection of ions due to direct ion collisions is called the *collection force*, $\mathbf{F}_\mathrm{c}$, and the force resulting from deflected ions is called the *orbit force*, $\mathbf{F}_\mathrm{o}$. The ion drag force on a dust particle can then be written as

$$\mathbf{F}_\mathrm{i} = \mathbf{F}_\mathrm{c} + \mathbf{F}_\mathrm{o}. \tag{2.33}$$

There are several models for the ion drag force. A model that can be used to give a qualitative estimate of the ion drag force is the *Barnes model* [13]. The Barnes model uses the assumption that the collection force comes from the same ions that contribute to the charging of the dust grain. Combining the expression for the OML ion collection cross section in eq. (2.11) with the expression for the general drag force in eq. (2.28), the collection force is obtained:

$$\mathbf{F}_\mathrm{c} = \pi a^2 m_\mathrm{i} v_\mathrm{s} n_\mathrm{i} \mathbf{u}_\mathrm{i} \left(1 - \frac{2e\phi_\mathrm{fl}}{m_\mathrm{i} v_\mathrm{s}^2}\right), \tag{2.34}$$

where m_i is the ion mass, n_i is the ion number density, $\mathbf{u}_\mathrm{i}$ is the ion drift velocity, ϕ_fl is the floating potential of the dust particle, and v_s is a kind of 'mean velocity' defined as $v_\mathrm{s} \equiv (u_\mathrm{i}^2 + v_\mathrm{th,i}^2)^{1/2}$, with $u_\mathrm{i} \equiv |\mathbf{u}_\mathrm{i}|$ and the thermal ion velocity $v_\mathrm{th,i}$ [57].

[5] Not to be confused with another 'version' of the thermal velocity, where the thermal velocity is defined as the most *probable* speed, $v_{\mathrm{th},s} = \sqrt{2 k_\mathrm{B} T_s / m_s}$, which corresponds to the peak of the isotropic Maxwellian distribution [72].

The orbit force comes, as mentioned, from the ions that are deflected by the Coulomb interaction but do not collide directly with the dust grain. The cross section of Coulomb interactions for finite sized dust particles can be calculated to be [13]

$$\sigma = 4\pi b_{\pi/2}^2 \ln\left(\frac{\lambda_{\mathrm{D}}^2 + b_{\pi/2}^2}{b_c^2 + b_{\pi/2}^2}\right), \tag{2.35}$$

where λ_{D} is the screening length of the plasma, b_{c} is the impact parameter for direct collisions of ions onto the dust particle, and $b_{\pi/2}$ is the impact parameter for $\pi/2 = 90°$ deflections of ions around the dust particle [57],

$$b_{\pi/2} = \frac{Q_{\mathrm{d}} e}{4\pi\varepsilon_0 m_{\mathrm{i}} v_{\mathrm{s}}^2} = \frac{ae\phi_{\mathrm{fl}}}{m_{\mathrm{i}} v_{\mathrm{s}}^2}, \tag{2.36}$$

where the second equation comes from using the capacitor model for the dust charge, contained in eqs. (2.19) and (2.20). This cross section excludes all ions at impact parameters $b < b_{\mathrm{c}}$, because these correspond to direct collisions that are already accounted for in $\mathbf{F}_{\mathrm{c}}$. By substituting the Coulomb interaction cross section in eq. (2.35) into the general drag force (eq. (2.28)), the orbit force is obtained:

$$\mathbf{F}_{\mathrm{o}} = 2\pi \frac{a^2 e^2 \phi_{\mathrm{fl}}^2}{m_{\mathrm{i}} v_{\mathrm{s}}^3} n_{\mathrm{i}} \mathbf{u}_{\mathrm{i}} \ln\left(\frac{\lambda_{\mathrm{D}}^2 + b_{\pi/2}^2}{b_c^2 + b_{\pi/2}^2}\right). \tag{2.37}$$

We can see that both the collection force and the orbit force scale with a^2, so the total ion drag force, which is the sum of the two, also scales with a^2.

An example of a more sophisticated model of the ion drag force is the Hutchinson/Khrapak model [42, 57]. This model takes into account the ions that stream past the dust particle outside of the Debye sphere. In addition, the Hutchinson/Khrapak model uses eq. (2.6) for the effective Debye length, which interpolates between the linearized Debye length λ_{D} (eq. (2.3)) and the Debye length of the electrons, $\lambda_{\mathrm{D,e}}$, where the interpolation depends on the ion drift speed.

The effective screening length used in the Hutchinson/Khrapak model has a different expression than the screening length in eq. (2.5), but they both have the same purpose, which is to correct the screening length when ions have a drift velocity. As mentioned in section 2.1.2, Ludwig *et al.* found that eq. (2.5), agreed better with simulations than eq. (2.6) for the ion drift Mach numbers between 0 and 2, which encompasses the relevant ion streams for the plasma analyzed in this thesis. For this reason, the screening length in eq. (2.5) will be used for our estimates for the ion drag force, and it seems reasonable that using this screening length will provide a more accurate ion drag force when using the Barnes model compared to using the Barnes model with the linearized Debye length or the electron Debye length.

Thermophoretic force

The thermophoretic force is a force on the dust particles that exists when there is a temperature gradient in the neutral gas. This force is directed toward the colder area. A basic, but not complete, explanation of this force is that the dust particle experiences on average a greater momentum transfer per unit time from the collisions with the neutrals on the hot side compared to the cold side since the hotter neutrals by definition have a higher (thermal) velocity. An analytical expression for the thermophoretic force, derived from kinetic theory [83], is

$$\mathbf{F}_{\mathrm{th}} = -\frac{32}{15} \frac{a^2 \kappa_{\mathrm{n}}}{v_{\mathrm{th,n}}} \nabla T_{\mathrm{n}}, \tag{2.38}$$

where a is the dust particle radius, κ_n is the thermal conductivity of the neutral gas, $v_\mathrm{th,n}$ is the thermal velocity of the neutral gas and ∇T_n is the temperature gradient of the neutral gas [17, 57].

2.3.3 Magnetic fields

The effects of magnetic fields on the dust particles are not important in this thesis. However, they are mentioned here for completeness' sake. It is estimated that a very strong magnetic field of at least $\sim 5\,\mathrm{T}$ is required to noticeably affect the dust particle dynamics of individual sub-micrometer dust particles [16]. This is because of the relatively low charge-to-mass ratio of the dust particles compared to ions or electrons. Ions are magnetized at $B \sim 1\,\mathrm{T}$ and electrons are magnetized at a few milliteslas. It is worth noting that at a few milliteslas there can be observed a slow rotation of the dust cloud as a whole [16].

2.3.4 Dust voids in RF plasmas

In dusty RF plasmas in microgravity conditions, under certain conditions a region free of dust particles, called a *void*, can be observed [31, 41, 57]. The void is located in the middle of the discharge and has a complete absence of dust particles and a sharp boundary [31]. Examples of voids can be seen e.g. in figures 2.2 and 3.2. In a typical dusty plasma chamber of size in the order of 10 cm in each spatial dimension, a typical void will have a size in the order of several centimeters, which is a substantial fraction of the chamber [11]. For dust voids to occur, the dust cloud must be extended in all three spatial dimensions. In ground based laboratory experiments, the dust cloud is often compressed, because of the gravitational force, into a sheet that hovers above an electrode due to electrostatic repulsion. Dust clouds can be extended in three dimensions when the force of gravity is negligible, which is either in microgravity situations or for nanometer-sized dust particles. Alternatively, the gravitational force can be counteracted by the thermophoretic force by heating up the lower electrode [57, p. 45].

The basic mechanism that creates the dust void is a force balance between an outward directed ion drag force and an inward directed electric force [31]. The outward directed ion drag force comes from an enhanced ionization rate at some location inside the discharge [31]. For higher pressures and higher RF power, there can be an outward thermophoretic force of the same order of magnitude as the outward ion drag force [30]. For lower pressure and lower RF power (which are the conditions relevant for this thesis), the thermophoretic force is negligible compared to the ion drag force, in which case the main outward force creating the void will be the ion drag force.

These forces can be understood through the following explanation of how a void develops from scratch, which is based on the discussion in Goree *et al.* (1999) [31]:
Starting out, it is assumed that the dust particles are uniformly distributed throughout the box or chamber containing the dusty plasma. In the uniform situation, the ionization rate will be more or less equal everywhere in the bulk plasma, because the electron number density will be more or less equal everywhere, and the electrons cause the ionization. Then, at some point in time, a spontaneous fluctuation in the dust number density will happen somewhere in the chamber. This opens up a small area where the dust density is lower, and therefore fewer electrons will be absorbed by dust particles in that area, giving rise to a slightly higher density of free electrons. This local increase of electron density gives a local increase in the ionization rate, which in turn increases the ion density there. In the area of increased ion density, the ions will repel each other and move outwards. This creates the outward ion drag force on the surrounding dust particles, which further expands the area where the dust density is lower, resulting in a small void.

The increased volume of the dust-free region gives a larger space with lower dust density, further increasing the ionization rate and the outward ion drag force. However, the increased positive ion density inside the void also sets up an inward electrostatic force on the dust particles since the dust particles are negatively charged. At some point the growth of the void saturates due to the growth of the inward electrostatic force. If the ionization rate is approximately constant in time, which is the case for many dusty plasma experiments, then the void will reach a stable equilibrium size [31].

The ion drag force will only be strong enough to create a void if the ionization rate in the plasma is high enough [41]. Additionally, the dust particle size must exceed a critical size in order for the ion drag force to overpower the electrostatic inward pull.

As mentioned in section 2.2, the time-averaged sheath regions around the electrodes have a net negative electric potential compared to the center of the discharge. Therefore, in the sheath regions the negatively charged dust particles will be repelled from the electrodes and confined to the bulk plasma, and ions will be attracted toward the electrodes. This can also affect the shape of the void.

2.3.5 Wakefield potentials and ion focuses

If a stream of ions is present in a dusty plasma discharge, the ion trajectories will be affected when passing the negatively charged dust particles. The ion trajectories are bent and converge into an area of higher ion concentration downstream of the dust particle which is called an *ion focus* or *ion wake* [57, 64, 51, 84]. The focus region of increased ion density sets up an electric force on other dust particles which can be described using two different models: a macroscopic wakefield model or a microscopic particle model [57, 64].

The wakefield model considers the structure of the electric potential surrounding the dust particle in the ion flow. The region behind the dust particle where the ion focus is present will have a positive electric potential relative to its surroundings, which sets up an electric field that will attract nearby negatively charged dust particles [60]. This enhanced electric potential or electric field in the wake of the upstream particle is what is referred to as the wakefield. Wakefields are found both in subsonic and supersonic ion flows [57, p. 65].

The ion focus model considers the trajectories of all of the individual ions streaming past the dust particle, which collectively form the downstream ion focus. The ions undergo a momentum exchange with any nearby dust particles due to the attractive electrostatic force, and the increased ion density in the ion focus results in that the negatively charged dust particles will experience a net attraction toward the ion focus. In the ion focus model, the attractive force on dust particles in the downstream region is essentially an ion drag force since it is due to the momentum transfer from the ions flowing through the focus [52].

The wakefield model and the ion focus model both describe the same phenomenon of an attractive force downstream of a dust particle in an ion flow. An illustration of these forces can be seen in figure 2.1. The attractive force on the downstream dust particle arises as a consequence of the presence of an upstream particle in the ion flow. This force is, however, not reciprocal with respect to the upstream particle; the upstream particle is not subjected to an equal but oppositely directed force. This is because the attractive force is due to the ions in (or rather, passing through) the ion focus, so it is the *ions* that receive the reciprocal forces from the downstream dust particle.

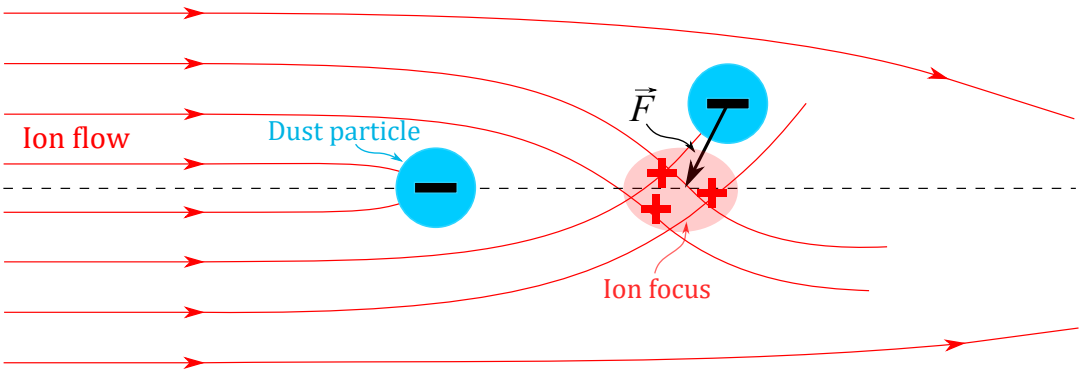


Figure 2.1: A cartoon illustrating a downstream dust particle being attracted toward the ion focus of an upstream dust particle in an ion flow. The force $\vec{F}$ is the sum of the ion drag and wakefield forces arising due to the ions flowing through the ion focus.

Ludwig *et al.* (2012) [54] showed with particle-in-cell simulations that for a dust grain with a potential of $\phi_\mathrm{p} = \phi_\mathrm{fl} = -2.0\,\mathrm{V}$ and an electron-ion temperature ratio of $T_\mathrm{e}/T_\mathrm{i} = 100$, the position of the first (positive) peak of the wakefield was located at one electron Debye length $\lambda_\mathrm{D,e}$ downstream from the dust particle for an ion drift speed of $M = 0.5$, with a potential value of 0.2 V at the peak. For higher ion drift speeds, between $M = 0.75$ and $M = 1.5$, the magnitude of the wakefield potential peaks increased, and the peak locations were pushed further downstream, to 1.1 - 1.7 $\lambda_\mathrm{D,e}$.

The ion drag force from the ions passing through the focus in subsonic ion flows, and specifically its role in the alignment of dust particles, has been studied by Piel (2011) [66]. Piel showed that the transverse ion drag force is the major component in the transverse or horizontal restoring force that results in the alignment of dust particles due to the ion stream. An estimate for the transverse restoring ion drag force was provided:

$$F_\mathrm{i,tr} \approx 2\frac{Q_\mathrm{d}^2}{4\pi\varepsilon_0 d^2}\frac{s}{d}, \tag{2.39}$$

where Q_d is the dust charge, d is the distance between two dust grains in the string, and $s \ll d$ is a small displacement distance from the equilibrium position of the downstream grain.

2.4 String formation

In certain dusty plasma experiments, strings of dust particles, also known as lanes or chains, can be observed [11, 35, 57]. Some examples of different kinds of dust strings that can occur in dusty plasmas are: Particle lanes in binary complex plasmas [44], strings in electrorheological plasmas [47], strings in three-dimensional dust clusters levitated by the thermophoretic force [48], and strings observed in microgravity RF plasmas [11, 35].

A particle string will here be defined as a set of particles that form an approximately straight line in three dimensions and contains a minimum of three particles. The dust strings observed in the experiments in this thesis also have approximately equidistant particles within a string. Another attribute of a string is that its constituent particles should be at least weakly bound to their relative positions in the string, so that the string is to a certain degree stable over some finite time interval. The string can still at some point break or dissolve, split into two strings, or merge with another string, however.

Figure 2.2 shows particle strings highlighted in one of the images taken during a parabolic flight experiment, in a low-pressure RF plasma. It is, however, important to note that what appears to be a particle string in a 2D image is not necessarily a connected string in three dimensions. This is because the particles can be at different depths with respect to the camera viewing angle, so two particles that appear to be short distance from each other in a 2D image can in actuality be quite far away from each other when taking depth into account. An actual three-dimensional string, on the other hand, will always look like a string when projected onto a 2D image, with the exception of when the string is perfectly aligned with the viewing direction, in which case the string will appear as a dense cluster of particles or a single particle in the image. But, in general, if strings are seen in a 2D image then it *can* be a string in three dimensions as well. For this reason, it is essential to use 3D diagnostics when studying dust strings in a three-dimensional dust cloud. A visual comparison between strings in two and three dimensions can be seen in section 5.3.1 (figure 5.11).

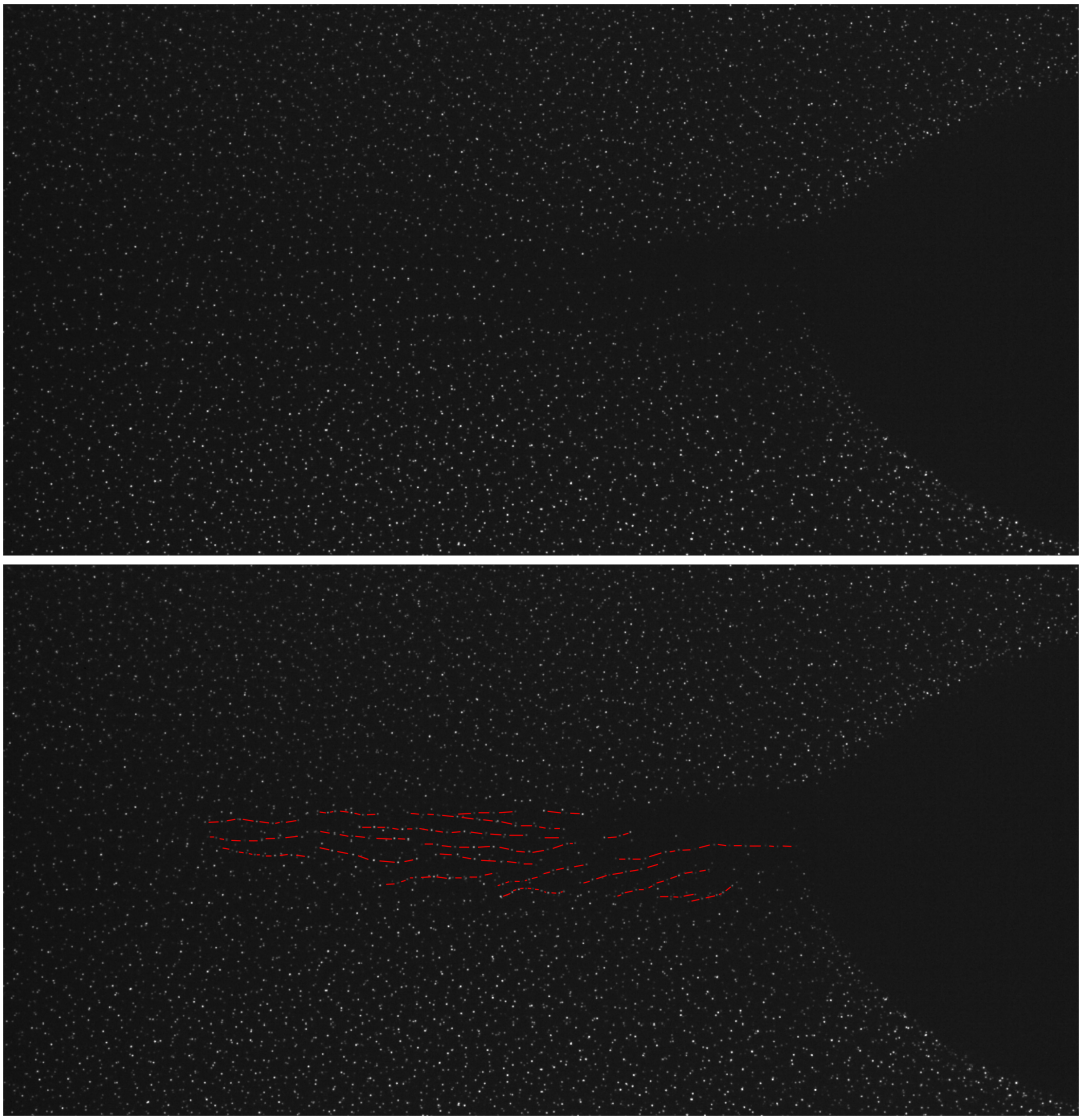


Figure 2.2: String formation of dust particles in an RF discharge in microgravity at low pressure. Top: A raw image of the dust cloud. Bottom: Same as the top image, but with some of the particle strings highlighted with red lines.

The main forces behind the string formation phenomenon is believed to be a combination of the transverse components of the ion drag force and the wakefield attraction force, as discussed in section 2.3.5 and illustrated in figure 2.1 [66, 11]. Both of these forces are the result of an ion flow and the electrostatic lensing effect of a negatively charged dust particle on the ion flow. The wakefield attraction downstream of a dust particle in a stream of ions can attract another dust particle into this ion focus. Then, another ion focus can be created behind that particle, attracting another dust particle to the new ion focus, and so on, which can result in the formation of longer particle strings. However, the string formation phenomenon is not fully understood, and detailed knowledge of how various forces or physical parameters affect things such as the shape, stability and interparticle distances of the strings is still lacking.

Dust particle charges in a string: Determining the charges on dust grains in the wakes downstream of other grains in an ion flow is a complex physical problem. For this reason, researchers have performed simulations in order to gain insight into the charging dynamics of such systems [61, 60, 15].

Miloch and Block (2012) [60] performed PIC simulations in order to investigate the dust particle charge at sonic ion speeds (Mach numbers between 1 and 1.5). They simulated five particle strings, with a length of three particles each, placed next to each other in such a way that the strings were aligned with the ion stream. The strings were separated by a distance of $1\lambda_{\mathrm{D,e}}$, and the distance between the particles in each string was $1.1\lambda_{\mathrm{D,e}}$. They investigated the charge on each of the three particles in one of the strings. The dust particles were fixed at stationary positions. Investigating the charge on each of the three particles in one of the strings for ion drift speed $M = 1$, they found that relative to the charge Q_1 on the first particle in the string (the particle furthest upstream), the second particle in the string had a charge of $Q_2 \approx 0.4Q_1$, and the third particle in the string had a charge of $Q_3 \approx 0.7Q_1$.

Another set of simulations performed by Block and Miloch (2015) [15] of a similar 5 string setup also including subsonic ion flows showed a smaller difference in charge between the particles in the string for lower ion drift speeds. At $M = 0$ (no ion flow) the particles had equal charge, and at $M = 0.5$ the charge distribution was $Q_2 \approx 0.67Q_1$ and $Q_3 \approx 0.90Q_1$.

They state that the relatively smaller degree of decharging in the third particle compared to the second particle is a consequence of the decharging of the second grain. The decharging of the second grain weakens its lensing ability, so that the third grain will receive a smaller influx of ions and will therefore be decharged to a lesser degree. The conclusion was that the charge distribution of the dust particles in a string can be different for the different particles in the string, and that the charge variation can be non-monotonous for consecutive grains. Note that these simulations did not consider electric fields or collisions with neutrals.

Due to the different charges on the different dust grains in a particle string, it can also be expected that the potential of the ion focus further downstream will be different at different points in the string, and therefore also that the aligning forces on the downstream dust particles will have different magnitudes at different points.

Circumstances where dust strings have been observed: Arp *et al.* (2012) observed particle strings in a microgravity RF plasma with a subsonic ion flow during parabolic flight experiments [11]. The experiments were performed in the IMPF-K2 plasma chamber, as discussed in section 3.1, as well as in the IMPF-K chamber, the predecessor of IMPF-K2 [65]. The only difference between the IMPF-K and the IMPF-K2 chambers is their electrode configuration. The IMPF-K chamber has its electrodes segmented into concentric rings such that the voltages are produced by two different RF power generators, and therefore two different peak-to-peak voltages can be used on the electrode segments [65]. The plasma was based on an argon gas.

In the IMPF-K2 chamber, with dust particles with diameter $2a = 9.55\,\mu\mathrm{m}$, pressure 50 Pa, and RF peak-to-peak voltage amplitude $V_{\mathrm{pp}} = 50\,\mathrm{V}$, strings were observed in the midplane region, at a similar location relative to the void compared to the strings seen in figure 2.2. A second experiment with the IMPF-K2 chamber with $2a = 9.55\,\mu\mathrm{m}$, argon pressure 30 Pa, peak-to-peak amplitude 45 V resulted in *no* string formation, and also no notable dilution in the midplane region. In the experiment performed using the IMPF-K chamber, the experiment parameters were $2a = 6.8\,\mu\mathrm{m}$, argon pressure 15 Pa, peak-to-peak amplitude $V_{\mathrm{pp}} = 70\,\mathrm{V}$ in the center electrode and $V_{\mathrm{pp}} = 50\,\mathrm{V}$ in the outer ring electrode. This resulted in a very dilute midplane region, with some particle strings present.

Himpel *et al.* (2018) performed microgravity dusty plasma experiments on parabolic flights

using the IMPF-K2 chamber [35]. For particles of diameter $2a = 7.01\,\mu\text{m}$, argon pressures between 15 and 30 Pa, peak-to-peak voltage between 70 and 100 V and RF power between 3 and 3.5 W, particle strings were observed in the dilute midplane region of the chamber.

2.4.1 The pair correlation function

For analyzing structures in dusty plasma experiments, often the pair correlation function $g(r)$, also known as the radial distribution function, is used. It is a function that gives the distribution of the particles as a function of the distance r of a reference point, providing information about how many particles can be found at different distances from either a specified reference point or from other particles. Alternatively, it be described as the probability that there exists a particle at a distance r from a chosen reference particle [57, p. 87].

The pair correlation function can be used as a quantitative measure of string formation by calculating $g(r)$ in the direction parallel to the observed strings, which will also be discussed in the following paragraphs. Since strings consist of several approximately equally spaced particles, $g(r)$ calculated along one dimension should show equally spaced peaks if strings are present. This is demonstrated in section 5.3.2. Pair correlation functions in two and three dimensions can for example be used as indicators of lattice structures in two and three dimensions, respectively [57, 77, 68].

The pair correlation function for a set of particles in a volume is calculated by going outward radially in a stepwise manner and counting the number of particles in each discrete distance interval of some width Δr. Essentially, this results in a histogram with data bins of width Δr. Then, the number of particles in each distance interval is normalized by dividing by the volume corresponding to that distance interval or search space. For a radial, three-dimensional distance search, the volume of a distance interval at some distance r corresponds to a spherical shell of radius r with a thickness equal to Δr. In addition to normalizing with respect to the size of the search spaces, $g(r)$ is also normalized by dividing by the total number of registered distances. In the three-dimensional case, the normalization involves dividing the number of particles N_i found in each increment volume $\Delta V_i \equiv V_{i+1} - V_i = \frac{4}{3}\pi(r_{i+1}^3 - r_i^3)$ in space (the volume difference of the spheres at r_{i+1} and r_i) by ΔV_i, giving the particle density in each increment. The implementation of the pair correlation function in three dimensions is shown in algorithm 1.

It is possible to calculate $g(r)$ for distances in two dimensions or one dimension by projecting the 3D positions onto a chosen plane or a chosen axis, respectively. In the case of two dimensions, the normalization divisors would be the area differences between concentric circles, $\Delta A_i \equiv A_{i+1} - A_i = \pi(r_{i+1}^2 - r_i^2)$. In the one-dimensional case, no spatial normalization is required because the size of the search space does not increase as r increases in one dimension; the search space is equal for all distances, $\Delta r_i \equiv r_{i+1} - r_i = \Delta r = \text{constant}$.

It is useful to establish how we can expect $g(r)$ to look like 'normally', meaning when there is no particular structure in the collection of particles. Figure 2.3 shows pair correlation functions for a uniformly distributed set of particles. Figure 2.3a shows the radial (3D) pair correlation function calculated with respect to one point, the center position of the volume, rather than including all distances between all particles. This is done in order to mimic $g(r)$ for an infinitely extending system. The particles at the edges of the box will have nearby particles at one side (inside the box), but no particles at the other side (outside the box). This reduces the total number of particles within a certain distance range for particles closer to the edges as r increases, which in turn reduces $g(r)$. By calculating $g(r)$ with respect to the center point only, we see approximately how $g(r)$ should look like for a uniformly distributed and infinitely extending set of particles up to a certain value of r, determined by the size of the box. In this case, the box is a cube of dimensions 8 mm $\times$ 8 mm $\times$ 8 mm centered at the

Algorithm 1: The pair correlation function in three dimensions.

```
rMax ← a value entered by the user
binWidth ← a value entered by the user
distances ← an empty array
counter ← 1
for j = 1 : nParticles do
    r_j ← position of particle j
    for k = 1 : nParticles do
        r_k ← position of particle k
        if k ≠ j then
            distances(counter) ← |r_k − r_j|
            counter ← counter + 1
N_distances ← length(distances)
binEdges ← 0 : binWidth : rMax    /* MATLAB syntax
N ← histcounts(distances, binEdges)    /* Built-in MATLAB function
/* Normalization in three dimensions (MATLAB syntax):  */
rMaxSphere ← (4/3)*π*(rMaxˆ3)    /* volume of a sphere with radius rMax
edgeSpheres ← (4/3)*π*(binEdges(:).ˆ3)  /* sphere volumes for radii equal to
 the bin edges
gr ← (N./N_distances) ./ ((edgeSpheres(2 : end) - edgeSpheres(1 : end-1)) /
 rMaxSphere)    /* the normalized pair correlation function
/* Plot g(r):  */
xValues ← the midpoints between the values of binEdges
plot(xValues, gr)
```

origin, so at a radial distance of $r = 4$ mm the sides of the box will have been reached. This is why $g(r)$ is seen to decrease for $r > 4\,\text{mm}$.

Because the particle positions were generated with a uniform random distribution, the particle density is on average equal in all parts of space. Figure 2.3a shows that $g(r)$ has a lot of variation between $r = 0$ and $r \approx 0.8$, and then it starts to converge towards a constant $g(r) = 1$ as r increases. A constant value of $g(r)$ as $r \to \infty$ is exactly what we expect in a uniformly distributed set of particles since the average particle density does not vary in space. The initial spikes and variations in $g(r)$ can be explained by the discrete implementation of the pair correlation function: The search volumes (spherical shells with a set thickness) are small for small r, and $g(r)$ is normalized by dividing on these volumes. The number of N_i found inside each incremental search volume is a whole number, so if N_i is equal to 1 or 2 and the search volume has a numerical value of much less than 1, then $g(r)$ will have a spike at that value of r. This can make $g(r)$ prone to having large spikes for small r.

Figure 2.3c shows the radial $g(r)$ for the same set of uniformly distributed particles, but now for all distances between all particles inside the cubical box. This plot shows that $g(r)$ starts decreasing immediately as the distance increases. This is due to the fact that no particles are found outside of the box, in combination with the fact that the normalization divisor (the volumes of the spherical shells) increases as r increases. For these reasons, $g(r)$ decreases to zero for all finite systems.

An observation that can be made is that the plot in figure 2.3c is smoother and less erratic compared to the plot in figure 2.3a. This is because there are a lot more registered distances in 2.3c than in 2.3a, since 2.3a only looks at the distances distances to all particles with respect to one point, and 2.3c looks at the distances to all particles with respect to all other particles. So, if there are N_p particles in total, there will be $\sim N_\text{p}$ registered distances in 2.3a and $\sim N_\text{p}^2$ registered distances in 2.3c.

Figure 2.3b shows a one-dimensional $g(r)$ for the same uniformly distributed particle set as above, with respect to one point, the origin. The one-dimensional version gives the distribution of distances along one axis with a specified, but arbitrary, direction $\vec{d}$. The axis that was chosen in this case was the y-axis, using $\vec{d} = [0, 1, 0]$. Note that in this case r is not the radial distance to a particle in three dimensions, but the distance to a particle along the y-axis. As expected for a uniform particle distribution, we see that $g(r)$ fluctuates around a constant value, in this case ≈ 0.0125, until it reaches the end of the box at $r = 4$ mm. At that point $g(r)$ decreases instantly to zero because it is the end of the cubical box; for $|y| > 4$ mm there are no more particles to be found.

The plot in figure 2.3d shows the one-dimensional pair correlation function along the y-axis, as in figure 2.3b, but where all distances between all particles are counted, as in figure 2.3c. Similar to the three-dimensional situation, when including all distances between all particles, $g(r)$ starts decreasing immediately with increasing r. Again, this is because now also the particles closer to the edges contribute to $g(r)$. So one edge, beyond which there are no particles, will be reached at $r < 4$ mm for these particles.

In section 5.3.2, $g(r)$ is calculated for a single string and for regions containing strings that are observed in experiments.

2.5 Particle swarm optimization

In this thesis, the particle swarm optimization (PSO) algorithm is applied to determining the positions of the dust particles from the images taking during the experiments. The PSO algorithm is a computational method for solving optimization problems [46, 63]. In this subsection, an explanation of the PSO algorithm will be presented along with the mathematics

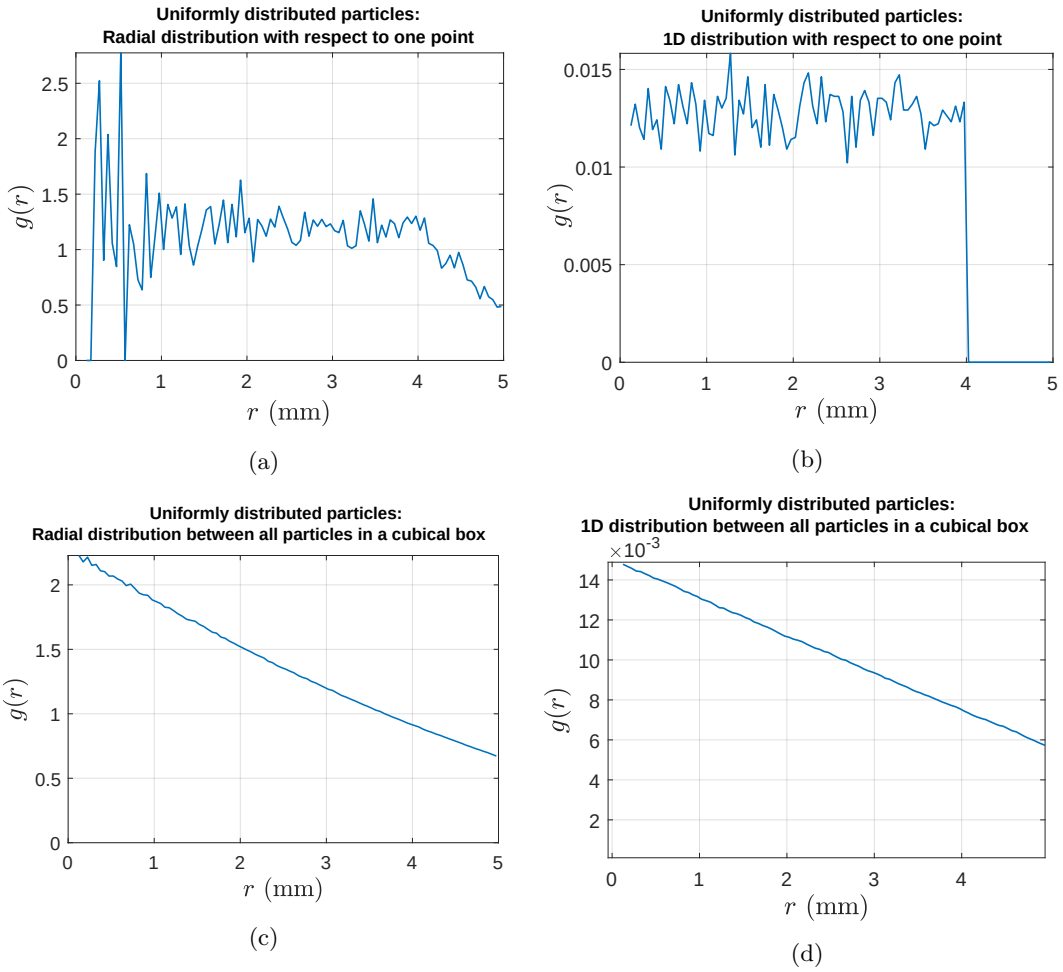


Figure 2.3: Pair correlation functions $g(r)$ for a randomly generated set of 10 240 positions with a uniform random distribution in a 8 mm $\times$ 8 mm $\times$ 8 mm box.

defining the algorithm. How the PSO algorithm is applied to finding dust particle positions is shown in section 4.1.

The PSO algorithm is in part inspired by swarm behavior in nature, such as in bird flocks or schools of fish [27, 75]. The term 'particle' in 'particle swarm optimization' refers to a candidate for the solution of the optimization problem at hand. The PSO particles are *not* the physical dust particles which are the physical subjects of this thesis. These two usages of the term 'particle' refer to two completely unrelated things, and the reader should be careful not to mix up the two meanings.

The algorithm is designed such that each particle has *memory* in the sense that the particle stores its own best position in the search space, where the 'best position' is the position where the function value is the most close to the optimal value, according to the relevant optimization problem. In addition to this, the global best solution, which is the best solution among all particles, is stored and updated each iteration. These terms are described in more detail in section 2.5.1. At the end of the algorithm, the best global position is returned as the solution of the optimization problem.

2.5.1 Mathematical model of PSO

Here, the mathematical definition of the PSO algorithm is presented. Similar descriptions can be found e.g. in references [85] and [46]. Consider a cost function $f : X \subset \mathbb{R}^n \to \mathbb{R}$. This is a function of n variables that we want to find the global minimum of. Let the total number of particles in the swarm be N_{swarm}. Each particle moves through a specified search space $X \ni \vec{x}_i(t)$ as the algorithm iterates, where $i \in \{1, 2, \ldots, N_{\text{swarm}}\}$ is the particle number and $\vec{x}_i(t)$ is the position of particle number i in the search space at timestep t. The search space is the collection of points in which the particles will search for the global minimum of f. The 'time' $t = 0, 1, 2, \ldots$ is here a dimensionless number.

At every iteration (timestep) of the algorithm, the position $\vec{x}_i$ and velocity $\vec{v}_i$ of the particle is updated. The velocity $\vec{v}_i \in X$ is a direction in the same space as the position $\vec{x}_i$, and it serves the same purpose as 'velocity' as it is used in physics, where the velocity of an object gives approximately the position at the next timestep, though in this algorithm there is no actual dimension of time, only iteration numbers.

The updated position $\vec{x}_i(t+1)$ is defined as

$$\vec{x}_i(t+1) \equiv \vec{x}_i(t) + \vec{v}_i(t+1). \tag{2.40}$$

The mathematical expression for the change in position at a given timestep, which defines the PSO algorithm, is

$$\vec{v}_i(t+1) = w\vec{v}_i(t) + c_{\text{P}}\left[\vec{p}_i(t) - \vec{x}_i(t)\right] + c_{\text{G}}[\vec{g}(t) - \vec{x}_i(t)], \tag{2.41}$$

where $w, c_{\text{P}}, c_{\text{G}} \in \mathbb{R}$. The coefficient w is the called the 'inertia coefficient' and $c_{\text{P}}, c_{\text{G}}$ are 'acceleration coefficients' that may or may not be different from each other. The coefficient c_{P} weights how much the *personal* best position $\vec{p}_i$ affects the velocity $\vec{v}_i$, and c_{G} weights how much the *global* best position $\vec{g}_i$ affects $\vec{v}_i$. The vector $w\vec{v}_i(t)$ is called the 'inertia component' of $\vec{v}_i(t+1)$, $c_{\text{P}}\left[\vec{p}_i(t) - \vec{x}_i(t)\right]$ is called the 'cognitive component', and $c_{\text{G}}[\vec{g}(t) - \vec{x}_i(t)]$ is called the 'social component' [81]. Figure 2.4 shows an illustration of how the position at the next timestep $\vec{x}_i(t+1)$ is calculated from the current position $\vec{x}_i(t)$.

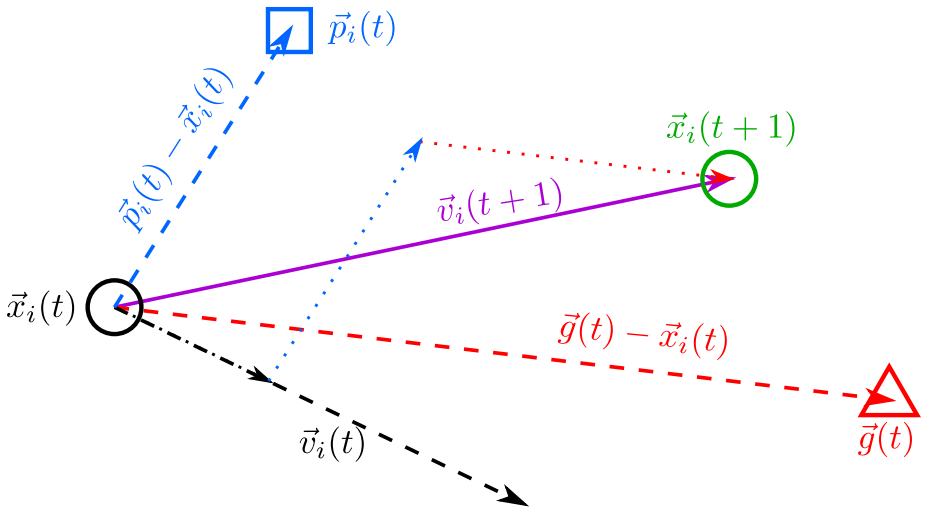


Figure 2.4: A vector illustration of how the updated position $\vec{x}_i(t+1)$ is calculated in the PSO algorithm in each iteration. The particle changes its position according to both an 'inertia component' (along the current velocity $\vec{v}_i(t)$), a 'cognitive component' (toward its personal best position $\vec{p}_i(t)$) and a 'social component' (toward the global best position $\vec{g}(t)$). Based on the weights w, c_{P} and c_{G}, the velocity $\vec{v}_i(t)$ is calculated from the personal best position $\vec{p}_i(t)$ and the global best position $\vec{g}(t)$ relative to the current position $\vec{x}_i(t)$. The vectors with dotted lines are scaled (weighted) versions of the vectors of the same colors with dashed lines, and $\vec{v}_i(t+1)$ is the sum of these weighted vectors.

Concrete examples of values of the weighting parameters are given in section 4.2.3.

Pseudocode for the general PSO algorithm: Algorithm 2 shows the basic steps of the PSO algorithm for solving the optimization problem of finding the global minimum of a cost function with n dimensions, based on the mathematical definition in eqs. (2.40) and (2.41). In this implementation, vectors with n elements of uniformly distributed random numbers between 0 and 1, generated by a function `rand(`n`)`, are multiplied element-wise with with the cognitive and social components. The symbol $\odot$ is used to signify element-wise multiplication for vectors or matrices (arrays), also known as the Hadamard product [59].

Algorithm 2: PARTICLE SWARM OPTIMIZATION: Finds the global minimum solution of a cost function.

Input: A cost function $f(x_1, x_2, \ldots, x_n)$, number of variables n
Output: The global minimum position $\mathbf{x}_{\min}$ of the cost function f
`/* Main PSO loop: */`
for it $= 1 :$ maxIt **do**
  for $i = 1 :$ swarmSize **do**
    $\mathbf{v} \leftarrow w\mathbf{v} + c_1 \cdot \text{rand(n)} \odot (\mathbf{x}_{\text{PB}} - \mathbf{x}) + c_2 \cdot \text{rand(n)} \odot (\mathbf{x}_{\text{GB}} - \mathbf{x})$
    $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v}$
    cost $\leftarrow f(\mathbf{x})$
    if cost $<$ pBestCost **then**
      pBestCost $\leftarrow$ cost
      $\mathbf{x}_{\text{PB}} \leftarrow \mathbf{x}$
      if cost $<$ gBestCost **then**
        gBestCost $\leftarrow$ cost
        $\mathbf{x}_{\text{GB}} \leftarrow \mathbf{x}$
  $w \leftarrow w_{\text{damp}} \cdot w$
return $\mathbf{x}_{\text{GB}}$

3 Experimental setup

The experiments that provided the data analyzed in this thesis were done in microgravity conditions during parabolic flights. The data analyzed in this thesis was collected during a flight campaign organized by the German Space Agency at DLR and hosted by Novespace in 2018. The dusty plasma experiments were performed by scientists from the University of Greifswald and the University of Oslo. The setup of these experiments are presented and explained in this section.

3.1 The IMPF-K2 plasma chamber

The plasma chamber used to produce the dusty plasma is called IMPF-K2. Descriptions of this plasma chamber can also be found in [35], [11], and [12]. A sketch of the IMPF-K2 chamber is shown in figure 3.1. The chamber is filled with an argon gas, which is ionized by applying an alternating voltage between two disk-shaped electrodes with diameter 80 mm, resulting in a so-called capacitively coupled plasma. The gap between the two electrodes is 30 mm. The alternating frequency of the voltage between the electrodes is 13.56 MHz and typically has a peak-to-peak voltage between 45 and 100 V [11, 35]. The electrodes in the IMPF-K2 chamber apply the same voltage everywhere on the surfaces of each electrode, and the two electrodes are operating in push-pull mode, meaning that the applied voltages of the two electrodes are phase-shifted by 180° with respect to each other. The pressure of the argon gas is typically 15 to 50 Pa.

Particles are injected into the plasma chamber during a parabolic flight using electromagnetically driven dispensers. The particles are illuminated by a laser sheet that is created by using a laser and a cylindrical lens. The dust particles used in this experiment were spherical particles of melamine formaldehyde (MF), a type of plastic resin [2]. Two species of dust were injected into the plasma chamber during the experiment; one species of pure MF with a diameter of 7.01 µm, and a second species of MF particles that were dyed with Rhodamine-B (RhB) with a diameter of 6.38 µm. Rhodamine-B is a fluorescent dye that can be used to trace selected dust particles [38]. An approximately equal number of pure MF particles and RhB particles

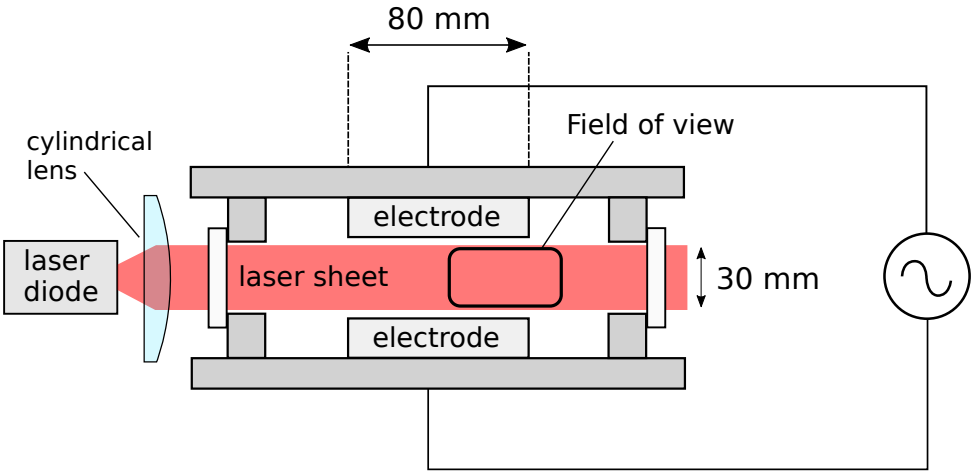


Figure 3.1: Sketch of the IMPF-K2 dusty plasma chamber (side view). When dust particles are injected into the chamber, they are illuminated by the laser sheet. The 'field of view' region is the region imaged by the cameras. Based on [11] and [58].

were injected into the plasma chamber. Figure 3.2 shows images of a large part of the dust cloud taken during the same flight parabola as the one that is analyzed in this thesis. The images are taken with two additional cameras, where one camera captures all particles (both species), and the other camera captures only the fluorescent RhB particles.

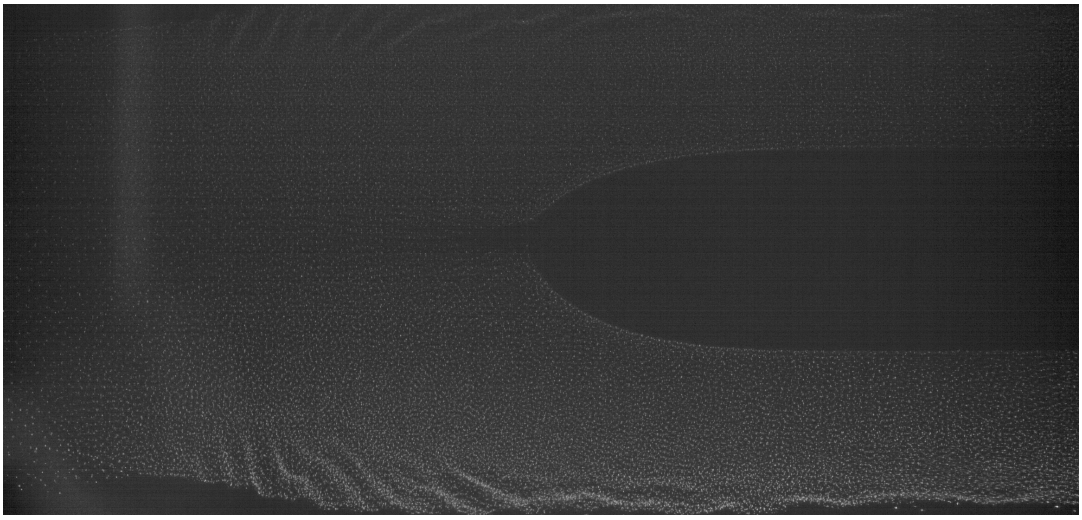
Because the outward directed ion drag force scaling as a^2 (see eqs. (2.34) and (2.37)), and the inward directed electric field force which is responsible for the void boundary formation scales linearly with a (see eqs. (2.21) and (2.23)), the larger 7.01 µm particles will be pushed further out than the smaller 6.38 µm particles. Therefore, the particles in the field of view of the images analyzed in this thesis are almost exclusively the slightly smaller RhB-coated MF particles with diameter 6.38 µm.

On the parabolic flights, around 20 seconds of microgravity conditions are realized during each parabola, and around 30 parabolas are performed each flight day during a parabolic flight campaign [35]. The dynamics of the dust particles are sensitive to even small amounts of residual acceleration, so usually only a few seconds of data from each parabola is usable for detailed analysis.

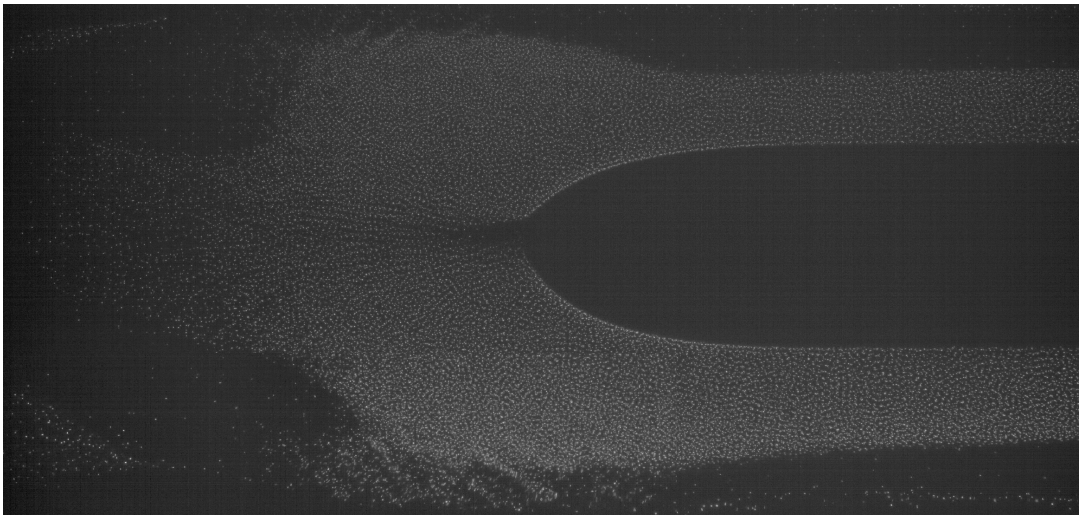
3.2 Stereoscopic imaging

In the parabolic flight experiments, four cameras were used to image the IMPF-K2 chamber, for the purpose of determining particle positions in three dimensions using *stereoscopic imaging*, or stereoscopy. The cameras used in these experiments had a frame rate of 200 frames per second (FPS). Stereoscopy is one of several different methods for reconstructing particle positions in three dimensions [57], and it is the method that was used for the work in this thesis. It is a method based on the principle that visual input from multiple different viewing angles gives depth of vision [56]. It is realized by taking images simultaneously using multiple cameras from different angles, and then using geometric optics to find the 3D position of the dust particles.

Using two cameras is sufficient to determine the three-dimensional position of an object [39], but since there are thousands of particles inside the chamber, some particles can be occluded by other particles in multiple cameras. Having more cameras allows for detection of



(a) Both species



(b) Fluorescent particles only

Figure 3.2

several more of these particles that would otherwise be hidden by occlusion.

Each camera is calibrated by the help of a reference image or calibration image in order accurately determine the so-called *projection matrix* of each camera [37]. The projection matrices are required for both of the stereoscopy-based algorithms that are described in section 4, the STB algorithm and the PSO algorithm. The mathematical details of the optics of stereoscopic imaging, including the projection matrices, can be found in [57, p. 181] and [37].

4 Methods

To allow for a detailed study of the structures and physics of a dusty plasma system, the three-dimensional positions of the dust particles must be determined. The particle positions are found by the use of image analysis tools developed for this purpose.

In this section I will explain the two aforementioned methods for determining particle positions in three dimensions from images: the 'Shake the box' algorithm and the particle swarm optimization algorithm applied to dust particle detection. In addition, the process that was used for optimizing the PSO algorithm will be described.

4.1 Determining the particle positions in three dimensions

When an image of the collection of particles is taken, the computer is not automatically able to distinguish between a particle and the background. The image is simply a two-dimensional array containing numbers that represent color intensities. However, the computer can perform mathematical operations on arrays, including element-wise addition, subtraction, multiplication and division. Subtraction is especially useful for comparing two images. Using methods that will be explained below, positions of particles in 2D images can be determined. This, in conjunction with the optics of stereoscopic imaging, can then be used to determine 3D positions of particles. In this subsection, the 'Shake the Box' algorithm and the particle swarm optimization algorithm applied to dust particle detection are described.

4.1.1 The 'Shake the Box' algorithm

The 'Shake the Box' (STB) algorithm is one of the current established algorithms for determining dust particle positions in three dimensions [39, 35]. The STB algorithm is an iterative reconstruction algorithm which is based on predicting the next position of the particles based on their trajectories, which reduces the search space of the algorithm and therefore saves a significant amount of computing time [70, 71] [57, p. 187]. A set of initial particle trajectories is required to initialize the STB algorithm, so another algorithm has to be used in order to get these initial trajectories. One method that has been used for this purpose is the 'Iterative Particle Reconstruction' (IPR) method [70]. IPR searches the entire reconstruction volume in each iteration and therefore it has a longer computation time compared to STB, which in each iteration only searches a small targeted sub-volume.

Based on the provided initial trajectories, the STB algorithm 'guesses' a 3D position for the particle inside this smaller search volume. Then, one small synthetic image is created for each camera angle, where each image contains one particle, corresponding to the guessed 3D position. The 2D positions in the images corresponding to the guessed 3D position are determined by using the projection matrices of the respective cameras. These synthetic sub-images are then subtracted from the corresponding sections in the input images, and the resulting difference images can be used to quantify how well the guessed 3D position matches the position of a real particle.

The small volume that STB searches is what is called the 'box', and it is 'shaked' (slightly moved) iteratively until a sufficiently good match between synthetic images and real images is found, hence the name 'Shake the Box' [70]. More details on the STB algorithm can be found in [70, 71, 57].

4.1.2 The particle swarm optimization algorithm

While the details of the inner workings of the PSO algorithm were explained in section 2.5, here, the application of PSO in the context of dust particle position detection will be detailed.

The PSO program/script that runs the PSO algorithm is written in MATLAB. The built-in MATLAB function `particleswarm` is used for performing the actual particle swarm optimization [4]. The various parameters that are used in the PSO program are described in more detail in section 4.2.3.

The main idea behind the usage of PSO for dust particle position detection is that the determination of dust particle positions can be turned into an optimization problem. A three-dimensional position, $\vec{r} = (x, y, z)$, is the input that we want to be optimized to match a 3D position of a dust particle observed in the images. Using the projection matrices, $\vec{r}$ will correspond to a 2D position $\vec{r}_{2\mathrm{D},i}$ in each camera, where i is the camera number. PSO requires a cost function to minimize, and in our case this cost function will be named the *residual function*, or simply the residual [6]. The residual between two images `I_sub` and `I_art` is defined in the following manner:

```
diff = I_sub-I_art; % Difference between real image and artificial image
diff(diff<0) = 0;   % Discard negative values
residual = (sum(diff(:))-sum(I_sub(:)))/sum(I_art(:));
```

where `I_art` is a small artificial image containing one artificially generated dust particle at the image position $\vec{r}_{2\mathrm{D},i}$ and `I_sub` is a sub-image of the same size copied from the corresponding position in the real image. Typically a 4×4 pixel sub-image of `I_art` and `I_sub` have been used in this project. From this expression we see that if `I_art` = `I_sub`, meaning that we have a perfect match, then the residual equals -1. If there is no particle in the sub-image of the real image at the guessed position then we will have `I_sub` = $\mathbf{0}$ (an array of zeros, excluding image noise), which leaves the `diff` array with only zeroes after all negative values are set to zero, resulting in a residual of 0. Initial testing that I did using two artificial particles in two separate images showed that the residual between the two images had a monotonous decrease toward -1 as the two particle positions converged toward each other. The optimization problem is thus to get the residual as low as possible. The above definition is for one image, or one camera angle. Using N cameras, the total residual corresponding to one three-dimensional position $\vec{r}$ is the sum of the residuals of the images corresponding to the different camera angles. In our case we use $N = 4$ cameras, so a perfect match would correspond to a total residual of -4.

After each time `particleswarm` finds a particle position, the program subtracts the found particle from the images of all camera angles (by setting, `I_sub` = `I_sub` − `I_art`), essentially deleting the found particle from the images in order to avoid detecting the same particle position multiple times.

The PSO program divides the reconstruction volume into voxels. The volume of the voxels can be specified by the user. Typically the entire volume is divided into anywhere between ten to a few hundred voxels, but in any case the program loops through all voxels so that the entire reconstruction volume is searched.

Each time the `particleswarm` function is called, it can find at most one 3D dust particle position. This is because it searches for one optimal 3D position with regard to minimizing the residual cost function. For this reason, the total number of times `particleswarm` is called (the number of iterations) should be greater than the (estimated) number of dust particles. This total number of iterations is spread out equally across all voxels. For example, with a voxel volume of $2\,\mathrm{mm}^3$ and a dust particle number density of $20\,\mathrm{mm}^{-3}$, the number of iterations of `particleswarm` in each voxel should be greater than $2\,\mathrm{mm}^3 \cdot 20\,\mathrm{mm}^{-3} = 40$ in order to make sure that dust particles are not missed. Each returned position by the `particleswarm` function is stored in an array. The PSO program continues to call the `particleswarm` function

[6] The definition of the residual function was developed by Michael Himpel.

until all of the voxels have been searched for the specified number of iterations. At the end, all of the found particle positions are returned by the program.

4.2 Optimization of the PSO parameters

In order to determine how well the PSO algorithm can work for the purpose of finding particle positions in three dimensions, the ideal values of the various parameters used in the program should be found. This subsection goes through the process of how the PSO algorithm has been optimized, including a description of the benchmarking process that was used in order to find more ideal values for the various parameters.

In order to test the accuracy of the PSO algorithm, it is necessary to have the ground truth positions of the particles, and also synthetic images based on these positions are also required. Then, the PSO algorithm is run on these images, and it returns a set of found positions. The accuracy of these positions can then be checked by comparing with the ground truth positions. In this benchmarking process, a candidate position returned by PSO is accepted as accurate if it is closer than 50 µm to one of the ground truth particles. From this acceptance or accuracy condition, some performance parameters have been defined [71]:

- The fraction of ground truth particles that are found accurately by at least one of the candidate positions: `foundFraction`.
  `foundFraction` = 0 means that no particles were accurately found, and `foundFraction` = 1 means that all particles were found.
- The fraction of candidate positions that were not accurately close to any of the ground truth positions (also known as ghost particles): `ghostFraction`.
  `ghostFraction` = 0 means that none of the candidate particles were inaccurate, and `ghostFraction` = 1 means that all of the candidate particles were inaccurate.
- The total runtime of the PSO program: `runTime`.

The ideal values of these performance parameters would be `foundFraction` = 1 and `ghostFraction` = 0. Both of these results parameters are important for the accuracy of the found positions. If the results have a high value of `foundFraction` but also a high value of `ghostFraction`, the program has found many particles but also many wrong positions. Similarly, if the results have a low ghost fraction but also a low found fraction, the returned positions will be accurate, but a lot of particles will be missing.

The runtime of the algorithm should of course ideally be as low as possible. However, as long as the runtime is not unreasonably long, it can be a good idea to prioritize the accuracy of the algorithm in order to get the most accurate representation of the dust particle system, and therefore also a higher quality analysis of the physics of the system.

4.2.1 Creating synthetic images for benchmarking

The synthetic images that were used for benchmarking were created with a MATLAB script that was developed as a part of this thesis. The main steps of the script are explained below.

The process of creating the synthetic particle sets has two main steps: 1) Generate the particle positions inside the specified volume, and 2) generate synthetic images for each camera angle based on the particle positions.

The particle positions were generated using a uniform random distribution, so each particle is assigned a random position inside the specified reconstruction volume (with volume V). The particle number density ρ_N in the box is specified in the script, which then calculates the total number of particles by multiplying the particle number density with the reconstruction volume,

rounded off to the nearest whole number: $N = \text{round}(\rho_N V)$. One restriction was added to the randomly generated particle positions: Each new position is only accepted if it is at minimum a specified distance `minParticleDistance` away from its nearest neighbor. The distance to its nearest neighbor is found with the built-in MATLAB function `knnsearch`. For the synthetic particle sets used in this project, values between `minParticleDistance` = 0.050 mm = 50 µm and `minParticleDistance` = 100 µm have been used. This is to emulate the realistic situation that dust particles are not arbitrarily close to each other due to electrostatic repulsion between the particles. A randomly generated position that does not fulfill this requirement is discarded, and a new random position is generated, until it lands on a position where the nearest neighbor is sufficiently far away. At the end of this process the program has produced an $N \times 3$ array, containing the three-dimensional positions of N particles inside the reconstruction volume.

With the particle positions obtained, the synthetic images are created by projecting the positions onto 2D images using the projection matrices [7] for the different camera viewing angles [87]. As in the experiments, we use four cameras for the synthetic images. The reconstruction volumes and the projection matrices used for the synthetic particle sets in this project have been slightly different from the ones used in the real experiments. For each camera, particles are added one at a time. Using the projection matrix for the current camera, each position is projected onto the plane of the camera viewing angle, resulting in a 2D image containing all of the particles. The images are in black and white, with the background being black and the particles being white. Each particle is added to the images as a white dot of exponentially decaying intensity from the center of the particle,

$$I_{\text{particle}}(x, y) = I_{\text{c}} \exp\left(-\frac{(x - x_{\text{c}})^2 + (y - y_{\text{c}})^2}{w}\right), \tag{4.1}$$

where (x, y) is a position in the 2D image, $(x_{\text{c}}, y_{\text{c}})$ is the particle center position in the image, I_{c} is the intensity at the particle center position, and w is the width of the particle (although the particle border is continuous and not discrete). The intensity of any position/pixel in the image can range between 0, which is completely black, and 1, which is completely white.

In the real images, there is some noise in the images; the background is not completely black, but has a slight tint of dark gray. Therefore, in order to make the images more realistic, some noise is added to the synthetic images. The noise is generated by adding a small random intensity to each pixel. The random distribution of the noise is a Gaussian distribution, with a specified mean noise intensity and standard deviation, capped between specified lower and upper boundaries. The values for the mean intensity and standard deviation of the noise were determined by trial and error by generating new particle sets and comparing them with the real images until a realistic-looking image noise level was achieved.

The first version of the synthetic images initially used a constant value of I_{c} for all particles in the images for all four cameras. This version of the synthetic images was used for the majority of the benchmarking of the PSO algorithm. An example of such a synthetic image is shown in figure 4.1b. A second version of the image-generating script was created at a later point in time, which provides more realistic-looking particles in the synthetic images. One of these images is shown in figure 4.1c. For comparison, a real image taken during one of the experiments can be seen in figure 4.1a. In the more realistic synthetic images, the particle width w was reduced, and the particle intensities were also reduced. In addition, the intensity of each particle was randomly generated with a Gaussian distribution, in a similar manner as with the image noise, in order to emulate the brightness variation of the particles in the real images. As with the image noise parameters, the values for the mean intensity and standard deviation that gave the most realistic-looking particles were determined by trial and error by

[7] The projection matrices used throughout this project were provided by Michael Himpel.

generating new particle sets until realistic-looking particles and images were achieved. The realistic-looking synthetic images allow us to get a more accurate idea of the performance of a particle reconstruction algorithm with regard to accuracy. It also enables us to generate images from real-image analysis in order to visually get a rough idea of the accuracy of the particle reconstruction. Such a comparison is shown in section 5.2 (figure 5.8).

4.2.2 The benchmarking process

The process of benchmarking and optimizing the parameters of the PSO algorithm for accuracy and runtime is done by running the algorithm for multiple specified parameter values in a systematic way. The goal is to get `foundFraction` as high as possible and to get `ghostFraction` and `runTime` as low as possible. The parameter values that accomplish a higher accuracy and/or a faster runtime will be considered as the ideal values.

However, given that there were seven main parameters that were included in the benchmarking, and the number of values for each parameter, and that each run took on average around 1.25 hours for the synthetic particle sets, a ballpark estimate of the total runtime of this N-tuple for-loop is roughly 5000 years. So instead of doing that, I benchmarked parameters one or two at a time, pairing parameters together that to the best of my judgement considerably affected each other in terms of the resulting accuracy and/or runtime of the algorithm. The consequence of this method is that the optimization most likely will not be perfect, but it was still able provide significant improvements in terms of both accuracy and runtime (see section 5.1).

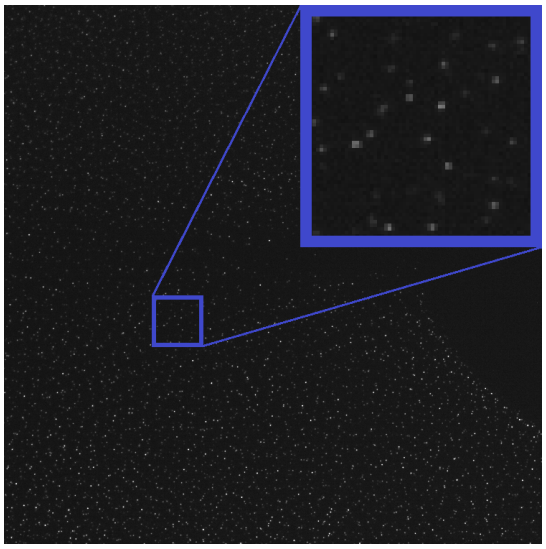
After each completed run of PSO, the results parameters for the current parameter value(s) are calculated/registered (`foundFraction`, `ghostFraction` and `runTime`). These results are saved in a `.mat` file. Each results file contains the relevant parameter value(s) in its file name. After this script has finished, another script goes through all of these files and imports the results so that the accuracy and runtime can be plotted against the parameter values, revealing the most optimal parameter values with respect to accuracy and runtime. After each parameter was optimized, the PSO program was updated to use the discovered optimal parameter values. This process was repeated for each of the PSO parameters until all parameters had been optimized.

4.2.3 PSO parameters

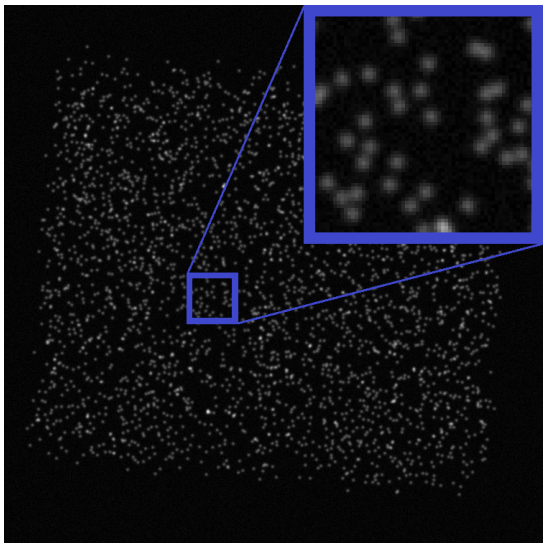
This subsection contains a list of the PSO parameters that were optimized, along with explanations for how their values affect the algorithm. For the parameters that go into the built-in `particleswarm` MATLAB function, descriptions of the parameters can also be found on the MathWorks® website, as well as more information about PSO in a MATLAB context: [4, 5].

`SelfAdjustmentWeight:` The `SelfAdjustmentWeight` parameter is the weighting of how much the PSO algorithm will adjust each PSO particle toward its current personal best position in the next iteration. It is the same as the coefficient c_{P} in eq. 2.41. Default value: `SelfAdjustmentWeight` = 1.49.

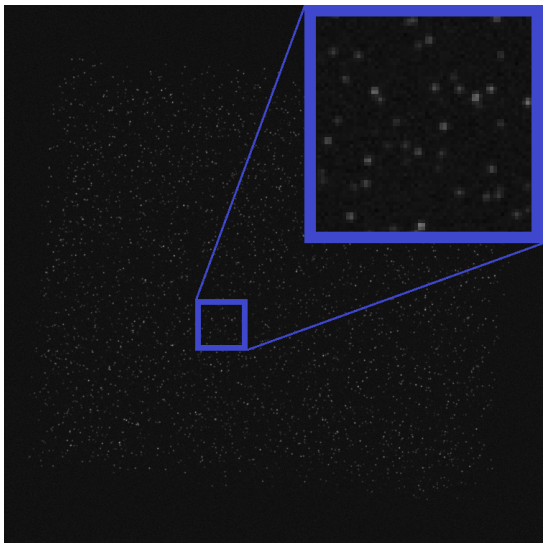
`SocialAdjustmentWeight:` The `SocialAdjustmentWeight` parameter is the weighting of how much the PSO algorithm will adjust each PSO particle toward the current global best position among all PSO particles in the next iteration. It is the same as the coefficient c_{G} in eq. 2.41. Default value: `SocialAdjustmentWeight` = 1.49.



(a) A real image



(b) A synthetic image, version 1



(c) A synthetic image, version 2

Figure 4.1: Comparison between synthetic and real images. The blue squares highlight a 70×70 pixel area in each image.

`FunctionTolerance` **and** `MaxStallIterations`**:** These two parameters together define a stopping criterion. If the relative change in the best function value from one iteration to the next is less than `FunctionTolerance`, the algorithm considers this to be 'stalling'. If the best function value changes with less than `FunctionTolerance` in each iteration for `MaxStallIterations` iterations (or in other words, if the algorithm stalls for `MaxStallIterations` iterations), the `particleswarm` function stops. Default values: `FunctionTolerance` $= 10^{-6}$ and `MaxStallIterations` $= 20$.

`MaxIterations`**:** This parameter defines a stopping criterion. If the number of iterations in the PSO algorithm reaches `MaxIterations`, the iterations stop. Default value: `MaxIterations` $= 200 \times$ (number of variables) $= 600$ in the case of three unknown variables (the three spatial coordinates of a dust particle).

`SwarmSize`**:** The number of particles in the PSO 'swarm'. Note that the PSO particles are not the physical dust particles, as mentioned in section 2.5. A larger swarm size gives more individual searching particles, which gives a higher possibility that the `particleswarm` function will more reliably successfully find a dust particle in the allotted number of iterations. Default value: `SwarmSize` $= 50$.

`iterationMultiplier` **(C_{it}):** The iteration multiplier parameter C_{it} determines the number of times N_{it} the `particleswarm` function will be run on each voxel in the reconstruction volume. This number is given by $N_{\mathrm{it}} = \mathrm{ceil}(C_{\mathrm{it}}\rho_N V_{\mathrm{vox}})$, where ρ_N is the average particle number density in the box and V_{vox} is the volume of the cubical voxels in the reconstruction volume. Note that the 'iterations' here are not the same iterations that were mentioned regarding the parameters `FunctionTolerance`, `MaxStallIterations` and `MaxIterations`. Those iterations were the number of low-level iterations performed inside the `particleswarm` function. In this context, the word 'iterations' refers to the number of times `particleswarm` itself is called. The `particleswarm` function can find at most one particle position each time it is called, which is why it must be called multiple times.

The reason why the number of iterations N_{it} is not set directly is because it is most practical to scale the iterations with the particle number density, regardless of the total volume or the voxel volume. If for example the voxel volume is increased, then the number of particles inside each voxel will increase correspondingly, which in turn makes it necessary to increase the number of iterations in order to find all of the particles inside each voxel. The above formula for N_{it} takes care of these issues by controlling the number of iterations on each voxel through the iteration multiplier C_{it}. C_{it} must be ≥ 1 in order for it to be possible to find all particles in the reconstruction volume, and realistically it should be a bit higher (e.g. $C_{\mathrm{it}} \geq 1.4$) in order to compensate for the fact that some voxels will contain more particles than others due to the random distribution of the dust particles, as well as the possibility that `particleswarm` may sometimes fail to find a particle position. When the real dust particle number density is not exactly known, as is the case for real images, an estimate for the particle number density must be made.

Note that increasing the number of iterations directly increases the total runtime of the PSO algorithm program as well, so C_{it} should only be as large as is necessary.

`residualLimit`**:** This parameter deals with the 'residual', which was described in section 4.1.2, but it is briefly recapped here. The residual is essentially the difference between a synthetically generated particle at a guessed position and the real image at that position. For each of the camera angles, a residual between 0 (zero match) and -1 (perfect match) is obtained. Summing up those for all N_{cams} cameras, we get a total residual between 0 (zero

match) and $-N_{\text{cams}}$ (perfect match). In our case we have $N_{\text{cams}} = 4$, so a residual of -4 represents a perfect match for a 3D position. Since a residual of -4 is basically impossible given the random fluctuations of dust particle intensities in real images, `residualLimit` is employed as the threshold value for which residual values are accepted as a successfully found particle position. If `residualLimit` $= 0$, every single position suggested by PSO, including the completely wrong ones, will be accepted. If `residualLimit` $= -4$, most likely not a single suggested particle position will be accepted. An appropriate balance seems to be found for `residualLimit` values somewhere in the range between -2.4 and -3.0.

`voxelVolume`: The volume of one of the cubic voxels that make up the reconstruction volume. Some voxels may be rectangular instead of cubical in order to not unnecessarily search outside of the reconstruction volume, but the cubic voxels are the ones with the largest volume. The voxels of a reconstruction volume are illustrated for two different voxel volumes in figure 4.2. The PSO program searches each voxel in turn, one at a time. It is not immediately clear whether a large voxel volume or a small voxel volume is better for accuracy or runtime, which is why the benchmarking of this parameter is necessary. Default value: `voxelVolume` $= 1$ mm^3.

4.3 PSO: Personal contributions

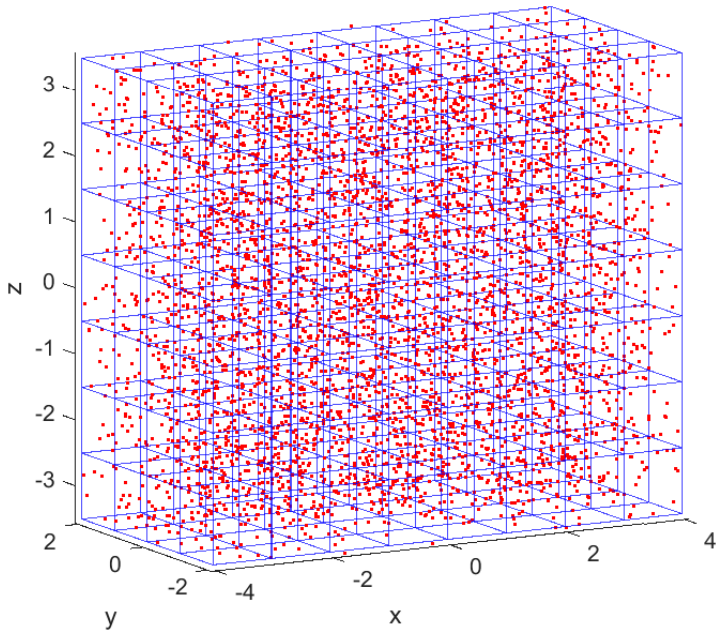
Before moving on to the results section, I would like to establish which parts of the program was already developed when I started the work on this project, and which parts I have developed.

The core of the code of the PSO program was written by Michael Himpel (University of Greifswald). This includes, but is not limited to, the following:

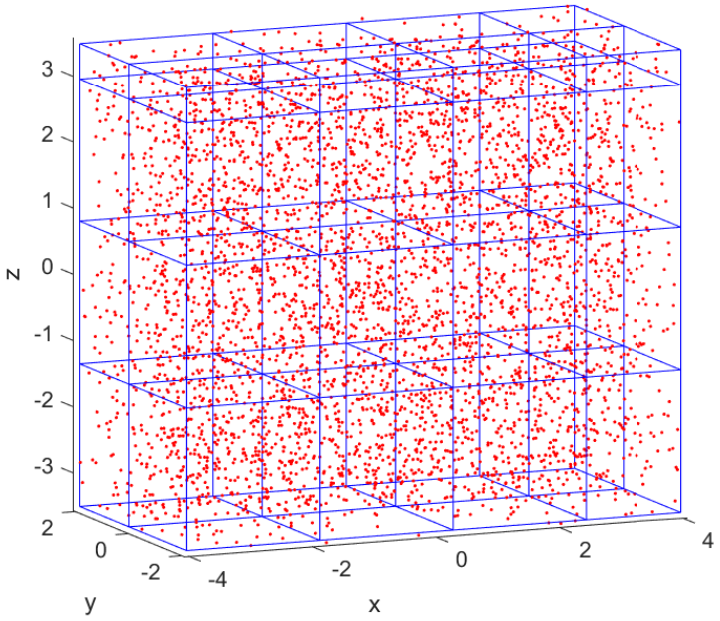
- Implementing the mathematics behind stereoscopic optics, including the creation of the projection matrices.
- Created the function that translates a voxel in 3D space to the corresponding regions in the images of the four different cameras.
- Created the definition of the 'residual' cost function.
- Created the function `PSO_findParticle()`, which uses the built-in MATLAB function `particleswarm()` to find the global minimum of the residual function within a given voxel of the reconstruction volume, and then returning a three-dimensional candidate position. optimize the given a set of positions to find them best suiting the recorded images.
- In general, created the foundational framework of the code and a working program for detecting three-dimensional particle positions using the PSO algorithm. There are many more functions and scripts that were not mentioned here that make up the initial framework of the PSO program.

My contributions and further developments include the following:

- Implementing the `residualLimit` criterion, which sets a requirement for the maximal allowed value of the residual cost function for the corresponding returned position to be accepted by the PSO program.
- Applied the MATLAB function `knnsearch()` to determining the distance of a particle in the returned set of positions to the closest ground truth particle (using synthetic particle sets).



(a) `voxelVolume` = 1 mm^3



(b) `voxelVolume` = 10 mm^3

Figure 4.2: The voxels of a reconstruction volume illustrated for `voxelVolume` = 1 mm^3 and `voxelVolume` = 10 mm^3. The reconstruction volume is equal to 4 mm × 7 mm × 8 mm = 224 mm^3. The red dots are particles from a computer-generated particle set.

- Introduced the `errorLimit` parameter, which specifies how large of an error distance is allowed from a returned position to the closest ground truth position for it to be considered as an accurate position. Using this parameter, I implemented the accuracy results parameters, namely `foundFraction` and `ghostFraction`.
- Implemented automatic calculation of a suitable number of `particleswarm()` iterations in the PSO program depending on the number of voxels that the reconstruction volume is divided into. The `iterationMultiplier` or C_{it} parameter, which is the scaling factor that determines the number of iterations based on the voxel volume, was introduced for this purpose.
- Implemented the support of rectangular voxels, which made sure that the entire reconstruction volume was covered, and that the voxels did not overextend beyond the reconstruction volume, which was the case for most reconstruction volumes when using only cubic voxels.
- Implemented 'shaking' of the voxels, where each voxel is systematically displaced slightly in a spatial direction in each iteration. This was done in order to find any undetected particles along the edges of the voxels. Additionally, I wrote a function that draws the generated voxels, as demonstrated in figure 4.2, for the purpose of visualizing the voxel sizes.
- Wrote the scripts that created the synthetic particle sets, as described in section 4.2. Randomly generated particle positions were projected onto the 2D images corresponding to the four camera angles by using pre-existing code and the projection matrices.
- As an additional required step for benchmarking the STB algorithm, I simulated simple particle trajectories for the artificial particle sets, since STB requires initial trajectories. The trajectories were based on a simple constant acceleration term for each particle and an additional inverse square repulsive force for particles that were close to each other. Each particle was assigned a random 3D position and velocity within specified borders, attempting to mimic the speeds observed in the real images. The trajectories were not created to be realistic in the physical sense, just realistic enough that the STB tracking could detect the trajectories, without making it too easy for the tracking algorithm (e.g. using only constant velocities).
- Created the benchmarking scripts as described in section 4.2, along with the 'infrastructure' that was required for it, including folder structures, results file names for different parameter values, value reading from specific results files for different parameters, etc.

5 Results and discussion

This section contains the results of the benchmarking process for the PSO parameters, as well as its application for reconstruction of dust particle positions from real data and the analysis of the obtained positions.

The optimal parameter values are determined and presented, and the finalized version of PSO is then tested against the STB algorithm on a realistic artificial particle set in section 5.1. In section 5.2, the PSO algorithm is used to obtain the three-dimensional particle positions from images taken during the parabolic flight experiments performed during a flight campaign organized by DLR in 2018.

The reconstructed dust particle positions will then be analyzed. Pair correlation functions will be calculated first for a single isolated string, and then for string-containing regions in the plasma chamber in section 5.3. Various forces on a dust particle in a string in the dilute region will be estimated in section 5.5. Before calculating the forces, estimates for the different required plasma parameters are calculated in section 5.4. The forces acting on a dust particle in a string with constant velocity will be considered and discussed.

5.1 PSO benchmarking

The benchmarking results (`accurateFraction`, `ghostFraction` and runtime) for the different parameters are shown in plots in this subsection. The mean values and standard deviations of the results for each of the parameter values, as well as the results from the individual runs, are indicated. Each of the parameter values were run on ten slightly different randomly generated synthetic particle sets. The number of particles inside the reconstruction volume in these particle sets ranges between 2240 and 4032.

For each parameter or pair of parameters, optimal parameter value choices will be suggested in three categories: Highest accuracy (prioritizing a higher accuracy, potentially higher runtime), faster runtime (prioritizing a faster runtime, potentially lower accuracy) and balanced settings (a balance between accuracy and speed). The settings that were used for the analysis of the real images from the experiments in this project were high accuracy settings and balanced settings.

`SocialAdjustmentWeight` and `SelfAdjustmentWeight`: Benchmarking results are shown in figure 5.1. `SocialAdjustmentWeight` and `SelfAdjustmentWeight` were benchmarked together in order to see how the results were affected by higher/lower emphasis on the swarm particles going toward the global best solution (`SocialAdjustmentWeight`) vs. higher/lower emphasis on the swarm particles going toward their personal best solutions (`SelfAdjustmentWeight`). The results do not show a very big difference for the different chosen parameter values, though the fraction of ghost particles is slightly lower for `SocialAdjustmentWeight` $= 1.3$ compared to the other values. Although the difference in runtime is miniscule, the runtime seems to have been a few minutes shorter for `SelfAdjustmentWeight` $= 1.3$ compared to the other values.

- **Recommended settings:** `SocialAdjustmentWeight` $= 1.3$ and `SelfAdjustmentWeight` $= 1.3$.

`FunctionTolerance` and `MaxIterations`: Benchmarking results are shown in figure 5.2. Here, the default value `MaxStallIterations` $= 20$ was used. The most notable results here are for the runtime, which shows a clear dependence on `FunctionTolerance`. The smaller `FunctionTolerance` is, the more iterations are required until the `MaxStallIteration` counter

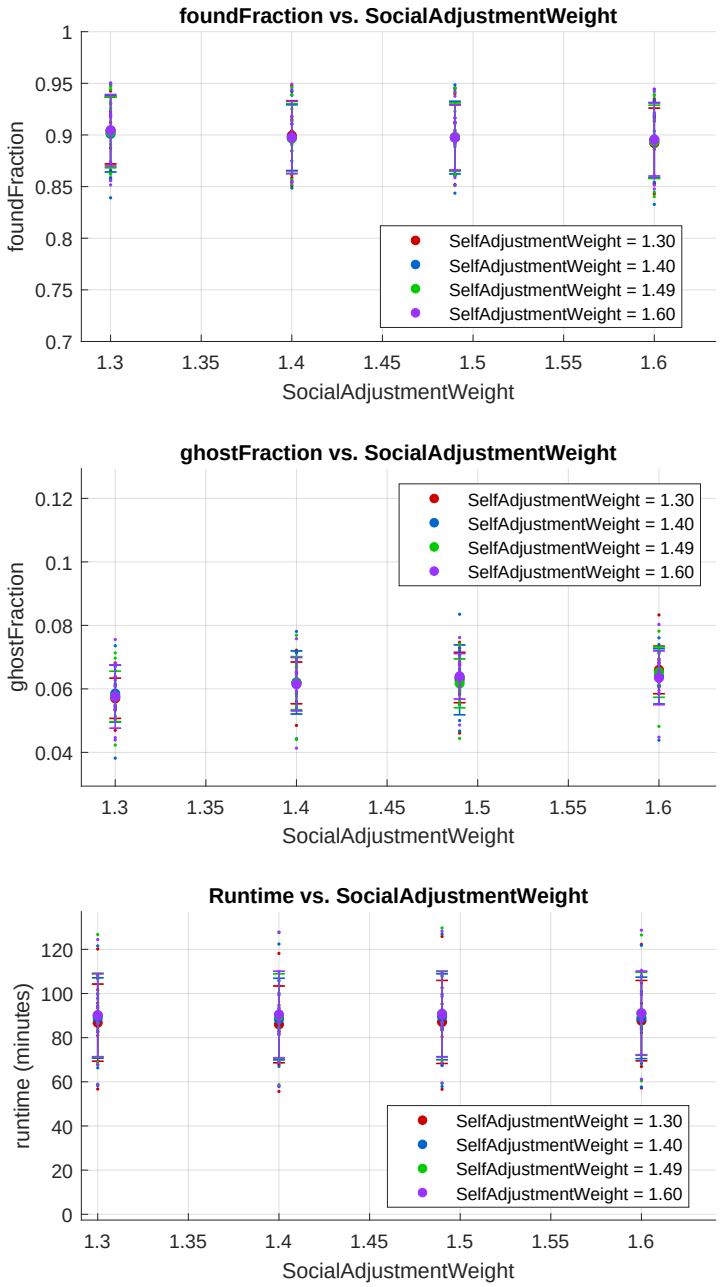


Figure 5.1: Benchmarking results for the `SocialAdjustmentWeight` and `SelfAdjustmentWeight` parameters in the PSO algorithm. The benchmarking is performed on 10 slightly different synthetic particle sets. The smaller dots represent the results from the individual particle sets, the larger dots represent the average results of all sets, and the error bars show one standard deviation of uncertainty around each average value.

starts, which means that more total iterations and therefore more time will be required for the `particleswarm` function to finish each time it is called.

There can be seen a slightly higher accuracy for smaller values of `FunctionTolerance` compared to the larger values. The results do not show a notable dependence between the two parameters in the chosen parameter value ranges. A possible reason for this is that the `particleswarm` function might find the solution many iterations before `MaxIterations` is reached for these values of `MaxIterations`.

- **High accuracy settings:** `FunctionTolerance` $= 10^{-7}$ and `MaxIterations` $= 500$.
- **Faster runtime settings:** `FunctionTolerance` $= 10^{-3}$ and `MaxIterations` $= 400$.
- **Balanced settings:** `FunctionTolerance` $= 10^{-6}$ and `MaxIterations` $= 400$.

`MaxStallIterations`**:** Benchmarking results are shown in figure 5.3. Here, the default value `FunctionTolerance` $= 10^{-6}$ was used. The value of `MaxStallIterations` is seen to significantly affect both accuracy and runtime. The values of `foundFraction` and `ghostFraction` reach a limit around `MaxStallIterations` $= 13$, where the accuracy improves no more with increasing `MaxStallIterations`. The runtime is seen to continue increasing with increasing `MaxStallIterations`. These results are reasonable, since a larger number of iterations at the end of the algorithm, even after the position is accurately found, will increase the runtime of each call of the `particleswarm` function. Furthermore, if the iterations terminate too soon, the `particleswarm` function may not have come sufficiently close to the correct particle position to be accepted as an accurately found particle. This means that that candidate position will be counted as a ghost particle instead of an accurately found particle, resulting in a higher `ghostFraction` and a lower `foundFraction`.

- **High accuracy settings:** `MaxStallIterations` $= 20$.
- **Faster runtime settings:** `MaxStallIterations` $= 8$.
- **Balanced settings:** `MaxStallIterations` $= 13$.

`SwarmSize` **and** `iterationMultiplier` **(C_{it}):** The `SwarmSize` and iteration multiplier parameters have been optimized together because if the PSO program performs few iterations of the `particleswarm` function, then the swarm size must be larger in order to thoroughly search the entire volume of the current voxel and find all of the particles there (or as many as possible). In the same way, if the program performs many iterations of the `particleswarm` function, then a smaller swarm size can be sufficient in order to search the volume thoroughly, and a larger swarm size would be unnecessarily time-consuming.

The benchmarking results are shown in figure 5.4. It is seen that the interdependence between the iteration number and the swarm size comes the most into play for the accuracy for `SwarmSize` $= 30$, where the value of C_{it} is seen to affect the `foundFraction` and `ghostFraction` results the most. Regardless, the highest accuracies are seen for higher values of `SwarmSize`. The runtime grows in a linear fashion with `SwarmSize`, and is further increased with higher values of C_{it}.

- **High accuracy settings:** `SwarmSize` $= 90$ and $C_{\mathrm{it}} = 2.0$.

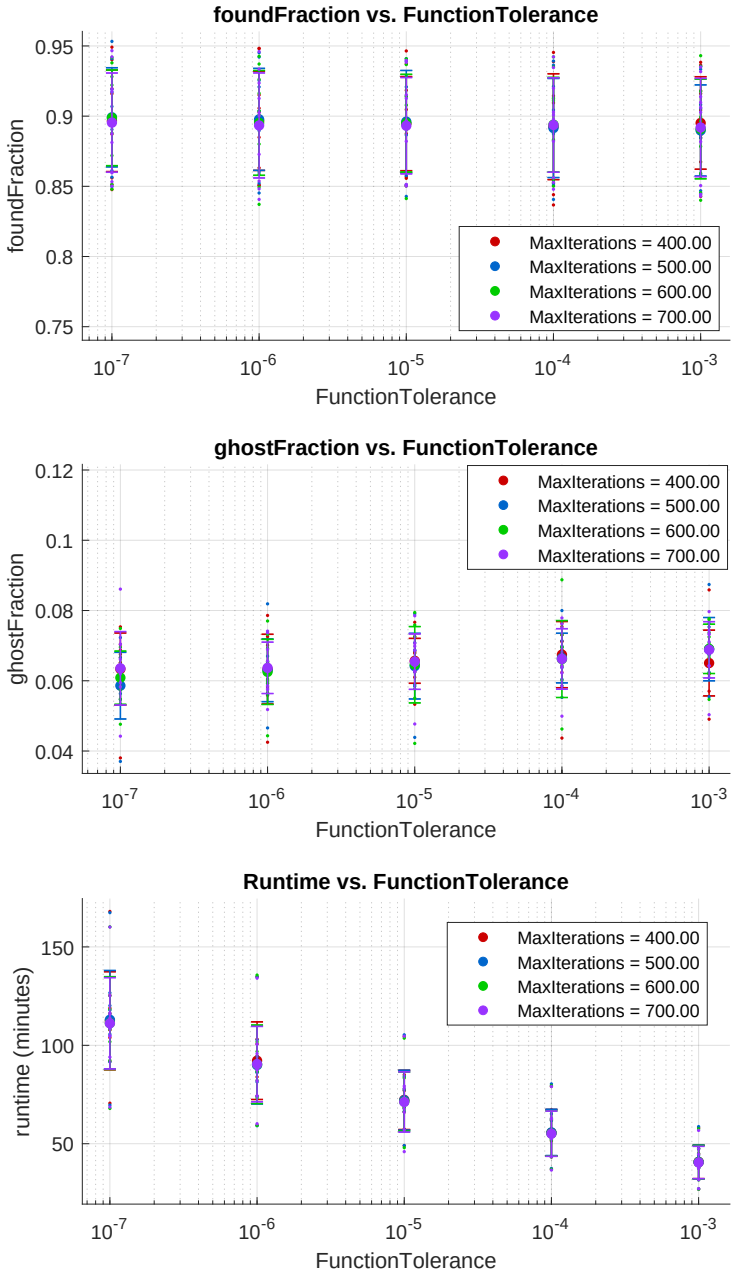


Figure 5.2: Benchmarking results for the `FunctionTolerance` and `MaxIterations` parameters in the PSO algorithm. The benchmarking is performed on 10 slightly different synthetic particle sets. The smaller dots represent the results from the individual particle sets, the larger dots represent the average results of all sets, and the error bars show one standard deviation of uncertainty around each average value.

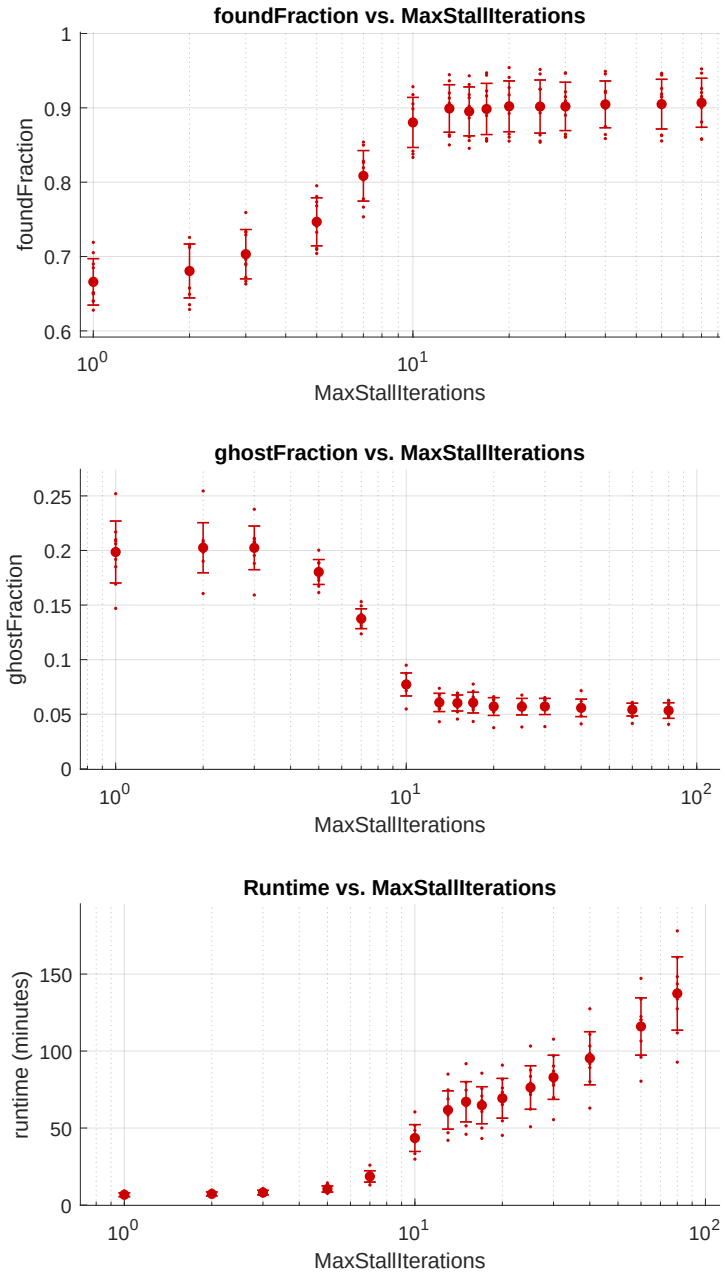


Figure 5.3: Benchmarking results for the `MaxStallIterations` parameter in the PSO algorithm, for values of `MaxStallIterations` between 1 and 80. Note that the x-axis is logarithmic. The benchmarking is performed on 10 slightly different synthetic particle sets. The smaller dots represent the results from the individual particle sets, the larger dots represent the average results of all sets, and the error bars show one standard deviation of uncertainty around each average value.

- **Faster runtime settings:** `SwarmSize` $= 40$ and $C_{\text{it}} = 1.4$.
- **Balanced settings:** `SwarmSize` $= 90$ and $C_{\text{it}} = 1.4$.

`residualLimit`: The benchmarking results are shown in figure 5.5. The results for `foundFraction` is highest for `residualLimit` values closer to -2, dropping as the value moves toward -3.6. The `ghostFraction` results decrease between `residualLimit` $= -2$ and `residualLimit` $= -3$. These are the expected results from the discussions of this parameter in section 4.2.3.

The increase in ghost fraction for `residualLimit` ≥ -3.4 is unexpected. It could possibly be an artefact from the fact that very few positions are accepted for more negative values of `residualLimit`. Regardless, these values of `residualLimit` are not really useful for dust particle detection of real data because of the low `foundFraction` results at these values.

There can also be seen a slightly higher runtime for `residualLimit` closer to -3.6 compared to values closer to -2. The reason for this is unknown, but the dependence is small. Choosing the value for `residualLimit` doesn't have much to do with the runtime of the PSO program, but rather it has to do with how many ghost particles the user wants to allow in the returned set of positions.

- **Higher detection threshold settings (lower `foundFraction` and lower `ghostFraction`):** `residualLimit` $= -3.0$.
- **Lower detection threshold settings (higher `foundFraction` and higher `ghostFraction`):** `residualLimit` $= -2.5$.
- **Balanced settings:** `residualLimit` $= -2.8$.

`voxelVolume`: The benchmarking results are shown in figure 5.6. Note that for some values of `voxelVolume`, the program threw an error and did not run. The reason for why this happened is not known, but it seems to be an issue only for certain values.

The highest average `foundFraction` results are at `voxelVolume` $= 2.5$ and 3. However, this average is based on only three particle sets each rather than ten, possibly due to errors for these values of `voxelVolume`. The lowest values for `ghostFraction` are at `voxelVolume` $= 8$ and 10, along with high `foundFraction` results. These values of `voxelVolume` ran successfully for all ten particle sets. The default value `voxelVolume` $= 1$ shows relatively high `foundFraction` results and reasonably low `ghostFraction` results, and is perhaps the safest choice with regards to avoiding voxel-related errors. The runtime does not seem to have a notable dependence on voxel volume, other than a couple of anomalous slightly higher runtimes for `voxelVolume` $= 4$ and 5.

It is likely that the ideal choice of the voxel volume depends on the total volume and/or the particle number density inside the total volume. This should be accounted for when using the PSO program on a particle set or reconstruction volume that is very different in size or particle density compared to the reconstruction volume used here. The reconstruction volume used for these artificial particle sets is 4 mm $\times$ 7 mm $\times$ 8 mm $= 224$ mm^3, and the particle number density is between 10 and 18 particles per cubic millimeter.

- **Recommended settings:** `voxelVolume` $= 1$ (safe choice), or `voxelVolume` $= 8$ or `voxelVolume` $= 10$ (possibly slightly lower `ghostFraction`).

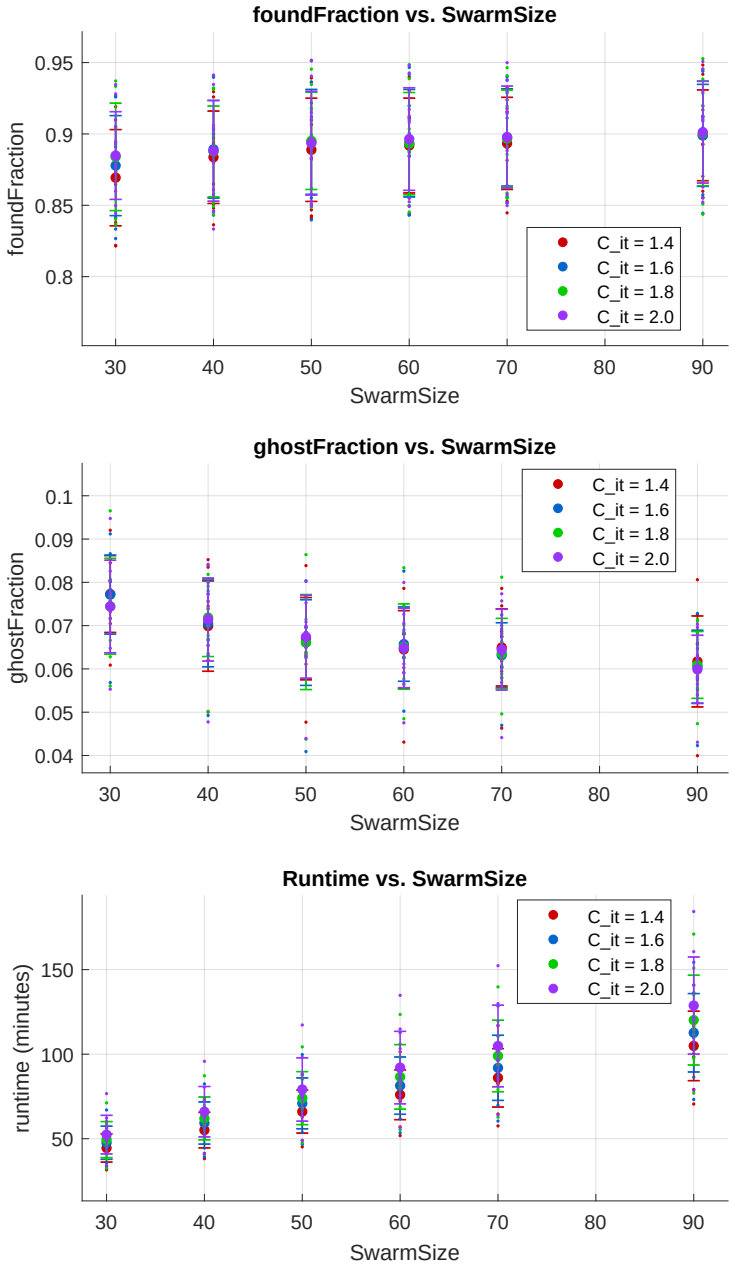


Figure 5.4: Benchmarking results for the `SwarmSize` and iteration multiplier (`C_it`) parameters in the PSO algorithm. The benchmarking is performed on 10 slightly different synthetic particle sets. The smaller dots represent the results from the individual particle sets, the larger dots represent the average results of all sets, and the error bars show one standard deviation of uncertainty around each average value.

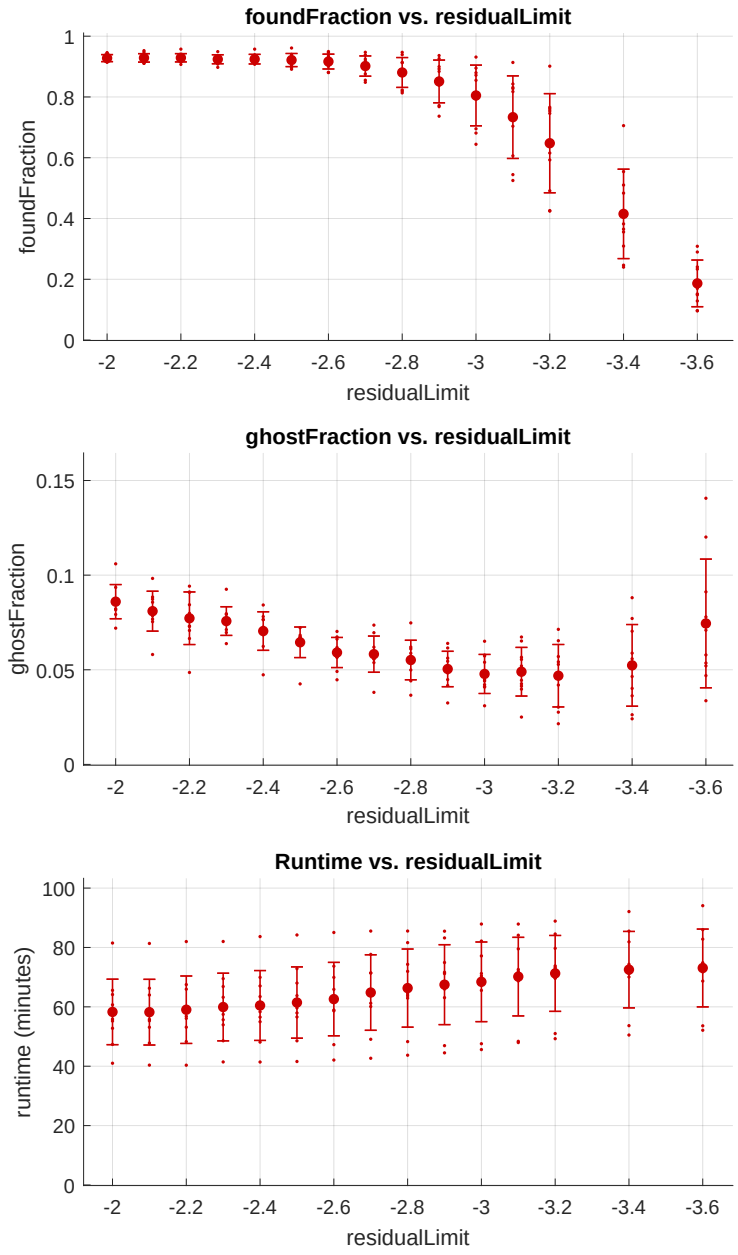


Figure 5.5: Benchmarking results for the `residualLimit` parameter in the PSO algorithm. Note that the x-axis is inverted. The benchmarking is performed on 10 slightly different synthetic particle sets. The smaller dots represent the results from the individual particle sets, the larger dots represent the average results of all sets, and the error bars show one standard deviation of uncertainty around each average value.

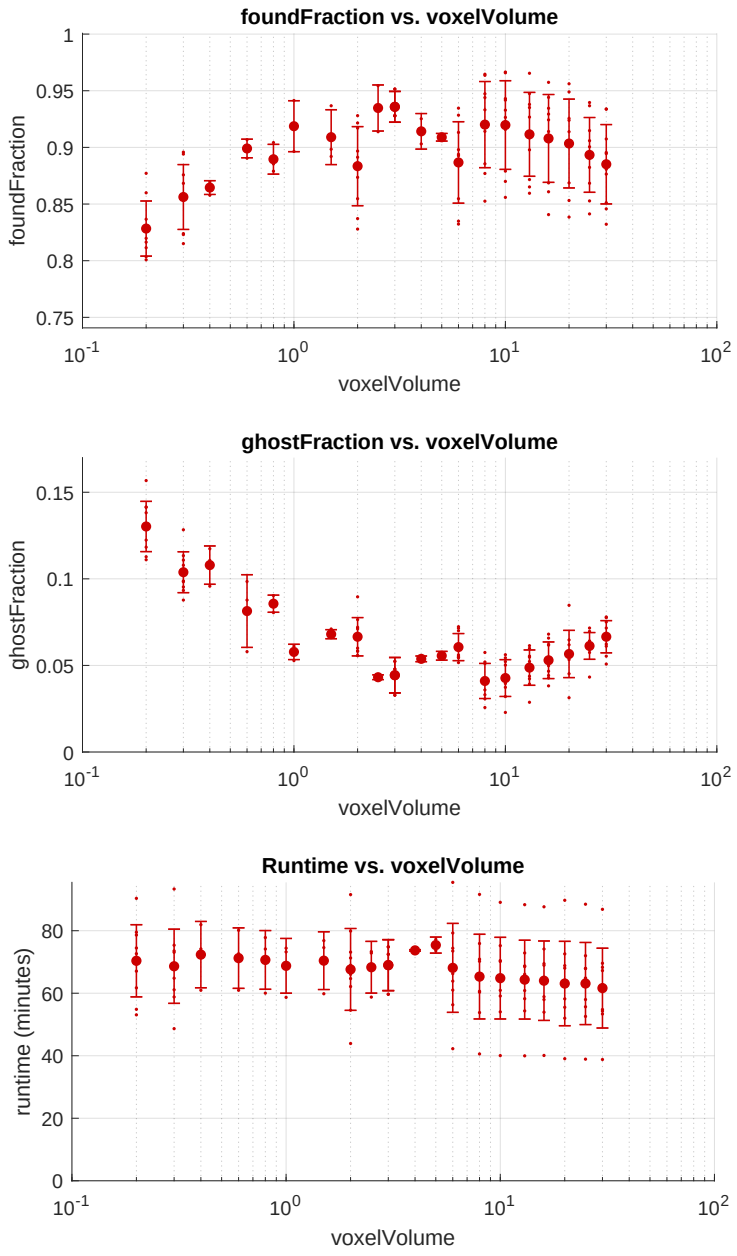


Figure 5.6: Benchmarking results for the `voxelVolume` parameter in the PSO algorithm. Note that the x-axis is logarithmic. The unit along the x-axis is mm^3. The benchmarking is performed on 10 slightly different synthetic particle sets. The smaller dots represent the results from the individual particle sets, the larger dots represent the average results of all sets, and the error bars show one standard deviation of uncertainty around each average value. Note that for some values of `voxelVolume`, the program threw an error for some of the particle sets, so not all values of `voxelVolume` have 10 points of results data.

5.1.1 Summary: Optimal PSO parameter values

As mentioned previously, the suggested parameter values are not necessarily *the* most optimal set of parameters, because of the runtime limitations which prevent the testing of all combinations of all parameter values. The suggested parameters found here are the most optimal parameter values based on the benchmarking that was done.

High accuracy settings:
`SocialAdjustmentWeight` = 1.3
`SelfAdjustmentWeight` = 1.3
`FunctionTolerance` = 10^{-7}
`MaxIterations` = 500
`MaxStallIterations` = 20
`SwarmSize` = 90
`iterationMultiplier` = 2.0
`residualLimit` = −2.8 *
`voxelVolume` = 1 **

Faster runtime settings:
`SocialAdjustmentWeight` = 1.3
`SelfAdjustmentWeight` = 1.3
`FunctionTolerance` = 10^{-3}
`MaxIterations` = 400
`MaxStallIterations` = 8
`SwarmSize` = 40
`iterationMultiplier` = 1.4
`residualLimit` = −2.8 *
`voxelVolume` = 1 **

Balanced settings:
`SocialAdjustmentWeight` = 1.3
`SelfAdjustmentWeight` = 1.3
`FunctionTolerance` = 10^{-6}
`MaxIterations` = 400
`MaxStallIterations` = 13
`SwarmSize` = 90
`iterationMultiplier` = 1.4
`residualLimit` = −2.8 *
`voxelVolume` = 1 **

* Alternatively: Use `residualLimit` = −3.0 or `residualLimit` = −2.5. See the '`residualLimit`' paragraph in the first part of section 5.1.

** Alternatively: Use `voxelVolume` = 8 or `voxelVolume` = 10. See the '`voxelVolume`' paragraph in the first part of section 5.1.

5.1.2 Final results: PSO vs. STB

In order to test and compare the accuracies of the PSO algorithm and the STB algorithm, an artificial particle set including particle trajectories over 20 frames was created. The artificial images were similar to the ones in figure 4.1c. Frames 1-10 were set aside as initial trajectories to initialize the STB algorithm; the PSO algorithm does not require initial trajectories. Frames 11-20 were analyzed by the PSO and STB algorithms. For this test, the 'error limit', which is the limit for what counts as an accurate particle position or a ghost particle position, was decreased from 50 µm to 30 µm.

For the STB algorithm, the mean results of these frames were `foundFraction` = 0.508 and `ghostFraction` = 0.089. The ghost fraction result for frame 11 was as low as 0.040, for frame 12 it was 0.063, and it increased until the ghost fraction reached around 0.10 (peak value 0.113).

The very low ghost fraction for the first frames is likely due to STB being supplied with perfect initial trajectories for frames 1-10. When analyzing real images, STB would require a separate algorithm, such as IPR (see section 4.1.1), in order to get the required initial trajectories. The initial trajectories obtained this way will not be perfect, which will reduce the accuracy of STB, depending on the quality of the initial trajectories. The main consequence of poor initial trajectories for STB seems to be a reduction in `foundFraction` rather than `ghostFraction`, since STB in general does a good job of filtering out ghost particles.

For PSO, the average accuracy results for the analyzed frames were `foundFraction` = 0.621 and `ghostFraction` = 0.193. The results were close to equal for all of the analyzed frames.

From these results we see that PSO has a higher fraction of found particles and a higher fraction of ghost particles than STB, so it is able to find more particles than STB, but it returns more ghost particles as well. Since the PSO algorithm does not require initial trajectories, the accuracy performance of PSO will most likely not be reduced as much as STB when comparing the accuracy of the results for synthetic particle sets and real particle sets. Based on these considerations, PSO is deemed to be a reasonable choice for analyzing the experimental dataset.

5.2 Particle position reconstruction for real images

With the optimized parameter values, the PSO algorithm is run on the real images from the parabolic flight experiments.

The acceleration data for the parabola during which the data analyzed in this thesis was collected is shown in figure 5.7. In the data analysis, frame numbers spanning over a range of just over 500 frames from the middle of the parabola, frames 2000 to 2504, were analyzed. At the frame rate of 200 FPS this corresponds to about 2.5 seconds of data. The exact synchronization between the time axis in figure 5.7 and the frame numbers could not be obtained. However, it was possible to make a rough estimate of the positions of the chosen frame range on the time axis based on when the parabola started and when particles began to appear weightless, along with the frame rate of the cameras. It was estimated that the 2.5 seconds of analyzed data lie roughly somewhere in the range between 9 and 13 seconds on the time axis in figure 5.7, with some room for error. In this time range, $|a_z|$ stays below $0.01g$, $|a_x|$ stays below $0.015g$ and $|a_y|$ is negligible. The maximum residual acceleration magnitude in the analyzed data is then $|\mathbf{a}_{\max}| \equiv a_{\max} = (a_x^2 + a_y^2 + a_z^2)^{1/2} \approx 0.018g = 0.177\,\mathrm{m/s^2}$. The force corresponding to this acceleration will be considered in section 5.5.

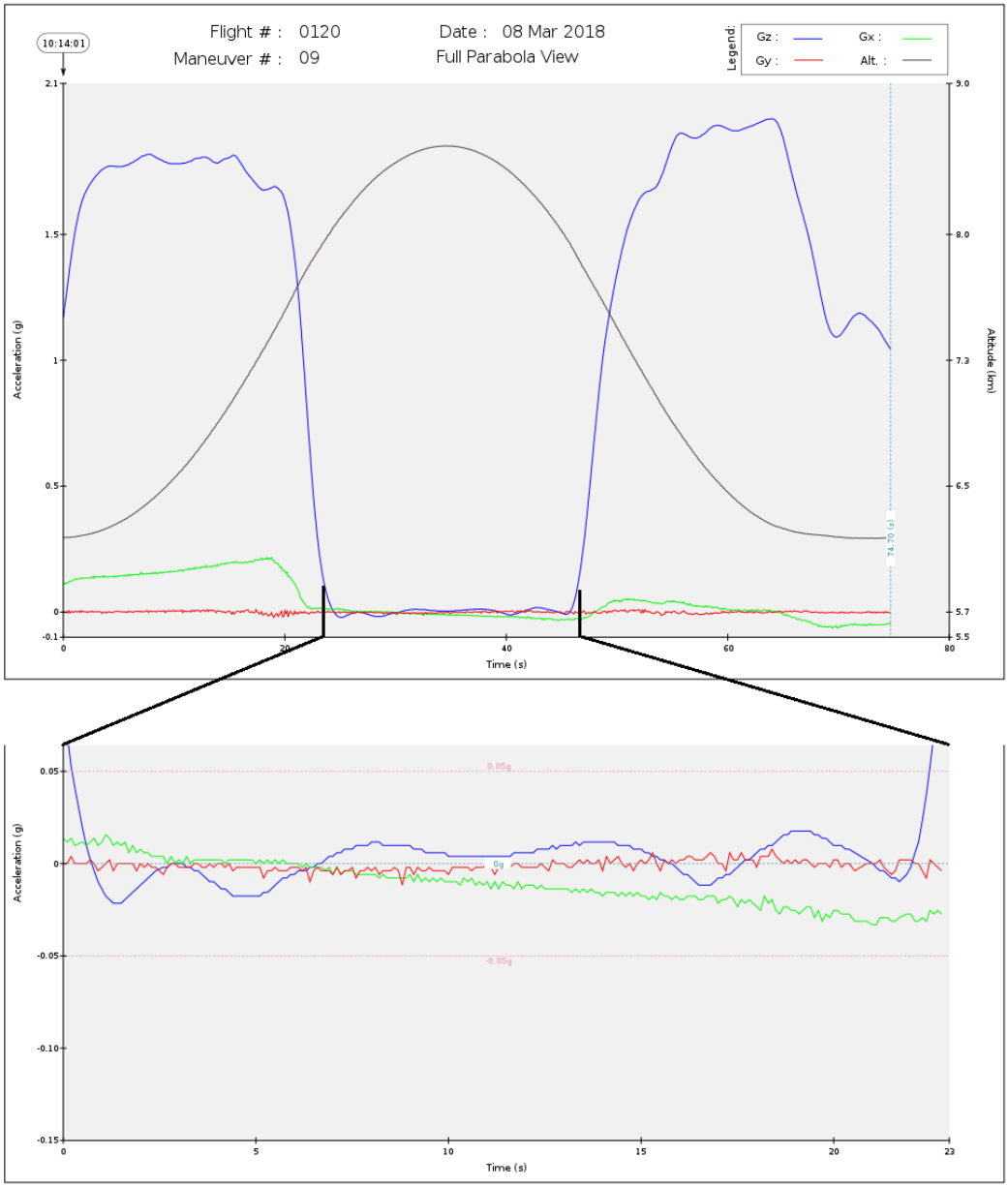


Figure 5.7: Acceleration data for the parabola during which the data analyzed in this thesis was collected. Top: Full cycle of the parabola, including the upward acceleration before and after the parabola. Bottom: Zoomed-in region containing only the acceleration data collected during the parabola. The acceleration a is plotted in units of $g = 9.81\,\mathrm{m/s^2}$. $a = 1g$ corresponds to the gravitational acceleration at Earth's surface and $a \approx 0g$ corresponds to microgravity. The altitude of the airplane is also indicated in the plot (black curve).

Note that the x, y and z-axes when speaking of the acceleration data (the frame of the airplane) are defined differently from the coordinate axes used for the coordinate system of the reconstructed particle positions. We can distinguish these coordinate systems by unit vectors $\hat{\mathbf{x}}_\mathrm{A}, \hat{\mathbf{y}}_\mathrm{A}, \hat{\mathbf{z}}_\mathrm{A}$ for the 'airplane coordinates', and $\hat{\mathbf{x}}_\mathrm{C}, \hat{\mathbf{y}}_\mathrm{C}, \hat{\mathbf{z}}_\mathrm{C}$ for the 'plasma chamber coordinates'

(the coordinates for the reconstructed particle positions). In the airplane coordinates, the positive z-direction $\hat{\mathbf{z}}_\mathrm{A}$ is defined to be the opposite direction of the gravitational pull $\mathbf{g}$ toward Earth. The laser sheet in the plasma chamber which illuminates the observed dust particles is a vertically extending plane in the airplane coordinates which extends in the directions $\hat{\mathbf{x}}_\mathrm{A}$ and $\hat{\mathbf{z}}_\mathrm{A}$. Comparing these directions with the coordinate system used for the reconstructed particles, we have $\hat{\mathbf{y}}_\mathrm{C} = \hat{\mathbf{z}}_\mathrm{A}$, $\hat{\mathbf{x}}_\mathrm{C} = \hat{\mathbf{x}}_\mathrm{A}$ and $\hat{\mathbf{z}}_\mathrm{C} = \hat{\mathbf{y}}_\mathrm{A}$.

A comparison between the PSO reconstruction results and the original image can be seen in figure 5.8. Features such as the void, the dilute region in the midplane region and strings are seen in both images. Note that the particles in the reconstructed image have random intensities, as described in section 4.2.1.

Since ground truth positions are not available for real data, we cannot determine the accuracy of the returned positions directly, but the tests on realistic-looking artificial particle sets gave accuracy values of `foundFraction` between 0.60 and 0.65, and `ghostFraction` of around 0.20. Because of the similarities between the artificial images and the real images, as seen in figure 4.1, the accuracy of the returned positions for the real images is likely to be in the same ranges.

In the following subsections, the positions returned by PSO from processing the real images from the experiments will be used to analyze the system of dust particles, with special emphasis on dust strings.

(a) Original, real image

(b) Reconstructed image based on the particle positions returned by PSO

Figure 5.8: Comparison between the synthetic image based on the reconstruction done by PSO (b) and the original image (a). The red rectangles highlight the overlapping regions in the two images where the PSO reconstruction has been performed.

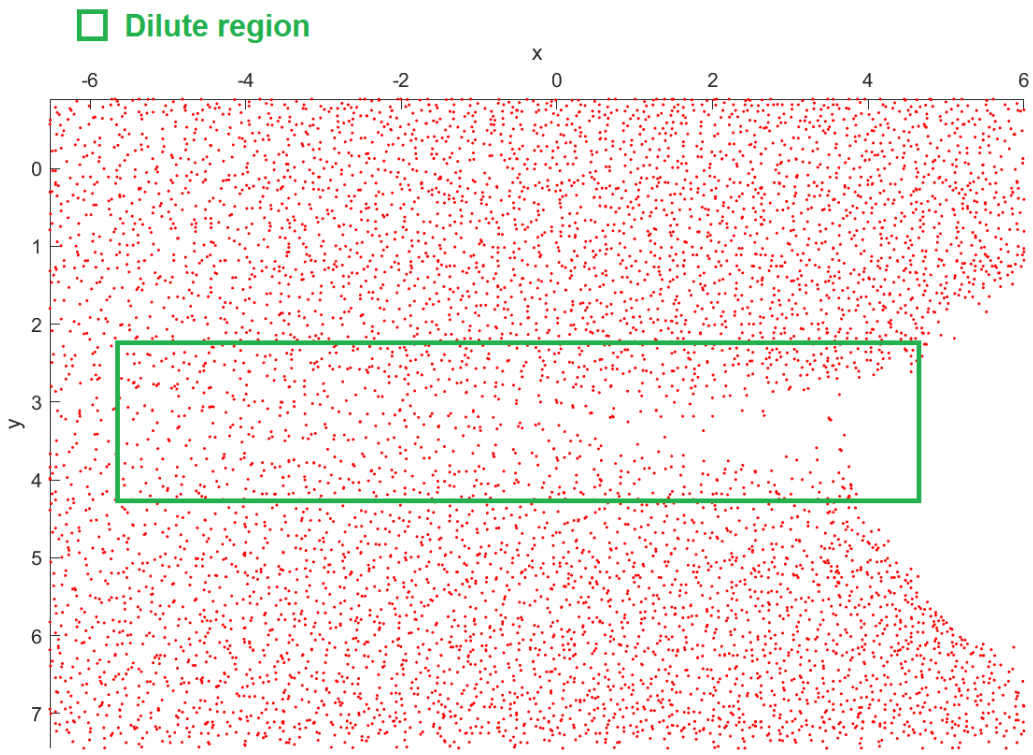


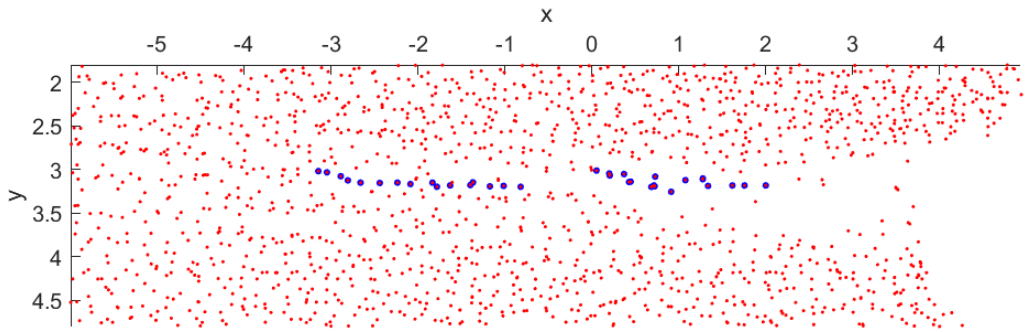
Figure 5.9: Three-dimensional particle reconstruction of frame 2150 from the experiment data, performed by the PSO algorithm. The green rectangle highlights the dilute midplane region of the plasma chamber.

5.3 String formation

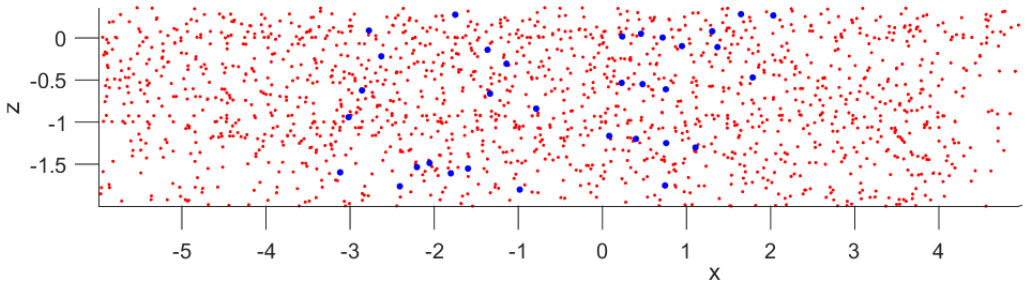
The positions found by the PSO algorithm in one of the analyzed frames is shown in figure 5.9. The dilute region, where the strings are located, are highlighted. It bears repeating that the set of reconstructed particle positions is not completely accurate compared to the real positions. Some particles will be missing, and some of the suggested positions will be slightly displaced from their actual positions. However, the reconstruction is deemed to be sufficiently accurate for the purpose of studying the observed string structures.

5.3.1 Particle strings in three dimensions

As mentioned in section 2.4, a string of particles seen in a two-dimensional image is not necessarily a string in three dimensions. This is demonstrated in figures 5.10 and 5.11. Figure 5.10 shows highlights of two groups of particles that look like particle strings when viewed from above. Seen from the side, in figure 5.10b, the particle group between $x \approx -3$ mm and $x \approx -1$ mm is seen to be a largely disconnected set of particles with much variation in the z-components of their positions. The particle group between $x \approx 0$ mm and $x \approx 2$ mm, on the other hand, is revealed to contain several three-dimensional strings, separated in the z-direction, along with some other separate particles. The strings at $z \approx -1\,\text{mm}$, $z \approx -0.5\,\text{mm}$ and $z \approx 0\,\text{mm}$ are seen to consist of 4 particles, 3 particles and at least 4 particles, respectively. This illustrates the necessity and usefulness of having the 3D positions of the dust particles



(a)



(b)

Figure 5.10: Manually highlighted apparent particle strings when seen from a two-dimensional view. Frame number: 2150. a) Top view (xy-plane). b) Side view (xz-plane).

for studying particle strings.

Figure 5.11 shows a handful of manually highlighted three-dimensional particle strings in the dilute region in frame 2150. This is the same volume that is highlighted in figure 5.9. The strings are viewed from different angles in order to show clearly that they are in fact strings in three dimensions. The strings are directed mainly along the x-axis, with a slight slope in the z-direction. When viewing the image sequence frame-by-frame, the dust particles, including the strings, are observed to drift in the negative x-direction. The string directions are seen to have slight variations in their y-components, but overall the directions show a symmetry about the y-axis. The string directions, along with the dust velocity (see section 5.4), could imply slight differences in the directions of the ion drift velocity at different points in the dilute region if we assume that the particle alignment axis coincides with the ion stream direction.

The slight slope of the strings in the z-direction is somewhat unexpected. From figure 5.11b, an estimate of the average angle of this slope is calculated to be 7.9°. Similar slopes in the z-direction were observed for the other analyzed frames as well. In the experimental setup of the IMPF-K2 chamber, the laser is placed so that it illuminates a plane that passes through the center of the plasma chamber. Though, a shift of the order of 0.5 to 1 mm cannot be fully excluded. Since the three-dimensional center of the void is illuminated, we would expect to see the same symmetry for the z-component of the string direction as is seen in the y-component in figure 5.11a.

One possibility is that the laser was slightly shifted at some point before or during the flight

of the experiment. In that case, assuming that the streaming ions from the center of the void have approximately azimuthal (cylindrical) symmetry about the vertical axis (the $\hat{\mathbf{z}}_{\mathrm{A}} = \hat{\mathbf{y}}_{\mathrm{C}}$ axis), the cross section illuminated by the laser sheet would contain an ion stream that has a slight bias toward either the positive or negative z-direction in the plasma chamber coordinates. It could also be that the void had not settled in the exact center of the plasma chamber, which also would end up illuminating a cross section of the dust cloud that does not pass through the center of the void.

Additionally, the positions of the dust dispensers may have a slight influence on the initial dust distribution. It is not very likely to be an effect of residual acceleration, since the acceleration component in the y-direction of the aircraft frame, which is equivalent to the z-direction of the plasma chamber frame, was negligibly low for the entirety of the parabola. However, the possibility can not be entirely excluded.

One or several of these effects could be the reason for the observed slopes of the strings.

5.3.2 Pair correlation functions

The pair correlation function or radial distribution function $g(r)$, as described in section 2.4, has been calculated for the found 3D positions of the dust particles from images taken during parabolic flight experiments performed in 2018. All of the pair correlation functions shown this section are calculated for all distances between all particles, as in figures 2.3c and 2.3d in section 2.4.

$g(r)$ for a single string

A single isolated dust string among the found positions is shown in figure 5.12. It consists of seven dust particles. The string was isolated by running a script that takes a set of found positions and discards all particles that are outside of a specified volume.

The actual average distances to the first, second and third nearest neighbors in this particle string was calculated using a simple script that sorted the particles in ascending x-values and then calculated the three-dimensional distances between the different particles. These average distances were 0.245 mm, 0.490 mm and 0.738 mm, respectively. This method will not work well for an entire particle set with multiple strings and other particles present, but in the simple case of one string it works, and it provides ground-truth values for the interparticle distances to compare $g(r)$ with.

Pair correlation functions in one and three dimensions have been calculated for the isolated dust string and are shown in figure 5.13. Figure 5.13a shows a non-normalized version of the radial (three-dimensional) pair correlation function. In the non-normalized version, the y-axis of the plot simply shows the number of registered distances at each value of r. The distance r is divided into data bins with a specified width `binWidth`. Additionally, a parameter `initialBinWidthIgnore` specifies an initial data bin width that is to be skipped by $g(r)$, that is, no distances $r <$ `initialBinWidthIgnore` will be counted by $g(r)$.

The normalized version of the radial $g(r)$ is shown in figure 5.13b. Due to the fact that other particles are only found along one axis and in no other direction, the three-dimensional normalization quickly decreases $g(r)$ as r increases.

Figure 5.13c shows a one-dimensional $g(r)$ which is calculated for the approximate direction of the string. The string direction is found manually from the difference of the position vectors of two particles in the string. In this case, the one-dimensional $g(r)$ is exactly equal in its shape to the un-normalized radial $g(r)$ in figure 5.13a, but with a normalization for one dimension. In actuality, in the one-dimensional $g(r)$ some of the radial distance is 'lost' due to projecting

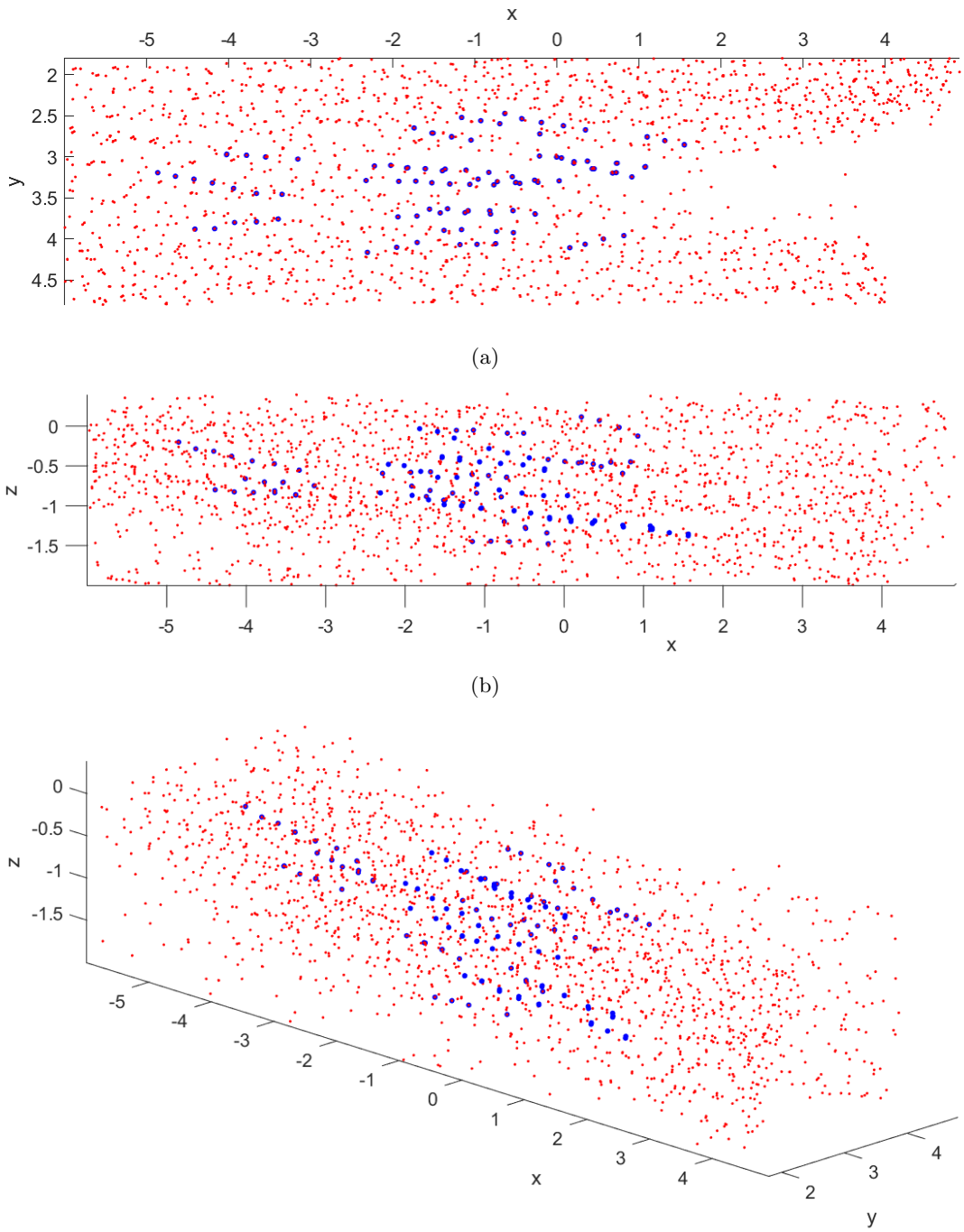


Figure 5.11: Manually highlighted three-dimensional particle strings in the dilute region of the plasma chamber. Frame number: 2150. a) Top view (xy-plane). b) Side view (xz-plane). c) Perspective view at an angle from 'behind' the strings.

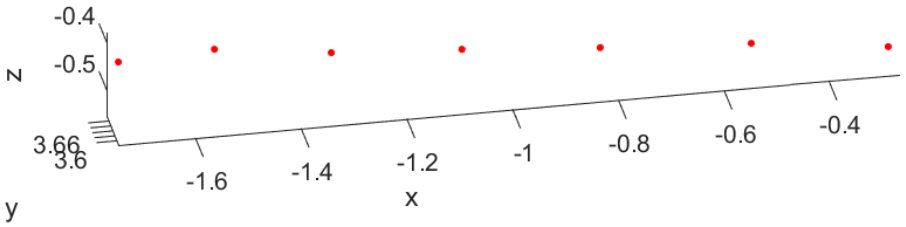


Figure 5.12: An isolated particle string in the found positions obtained from the PSO algorithm. The axis unit is millimeters.

distances onto the string direction axis, but in this case this projection did not affect $g(r)$ since the difference was too small to make any distances change bins.

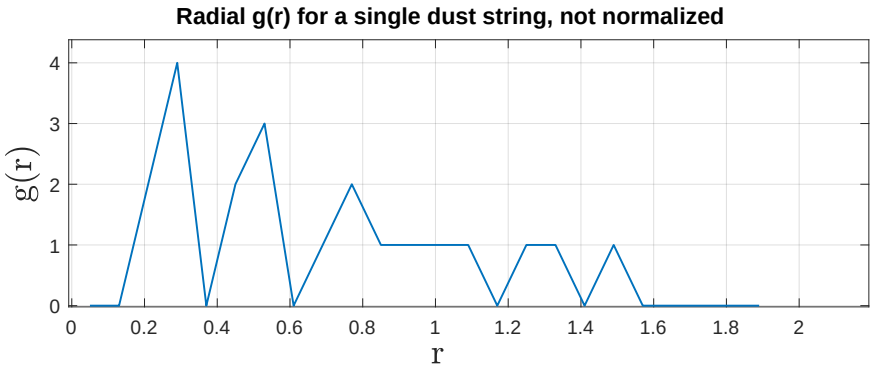
Examining the shape of $g(r)$ in figure 5.13a, it is seen that the first peak is located at $r = 0.290\,\text{mm}$, the second peak is at $r = 0.530\,\text{mm}$, and the third peak is at $r = 0.770\,\text{mm}$. In this case, the second peak is actually distributed almost evenly between the two points $r = 0.450\,\text{mm}$ with $y = 2$ registered distances and $r = 0.530\,\text{mm}$ with $y = 3$ registered distances. The average of all these distances is $r \approx 0.498\,\text{mm}$, which is significantly closer to the approximate ground truth value of $r = 0.490\,\text{mm}$. The same thing is true for the first peak as well, with $y = 2$ at $r = 0.210\,\text{mm}$ and $y = 4$ at $r = 0.290\,\text{mm}$, giving a weighted average of $r \approx 0.263\,\text{mm}$, which again is closer to the ground truth value of $0.245\,\text{mm}$.

This leads to some uncertainty in using the peaks in $g(r)$ as indicators of the interparticle distances within particle strings. In order to reduce this uncertainty as much as possible, the two parameters `binWidth` and `initialBinWidthIgnore` should be adjusted so as to most accurately match the real interparticle distances. The bin width should be small enough that peaks and valleys of $g(r)$ are captured, and also be wide enough to avoid a too high resolution, which will result in a large number of peaks which can make features of the plot difficult to interpret. The `initialBinWidthIgnore` parameter is used to shift all of the bins by a small distance (less than one bin width) in order to find more pronounced peaks and valleys in $g(r)$. If the peaks are more well-defined it means that the peak positions are more likely to more closely represent the actual interparticle distances in the string(s).

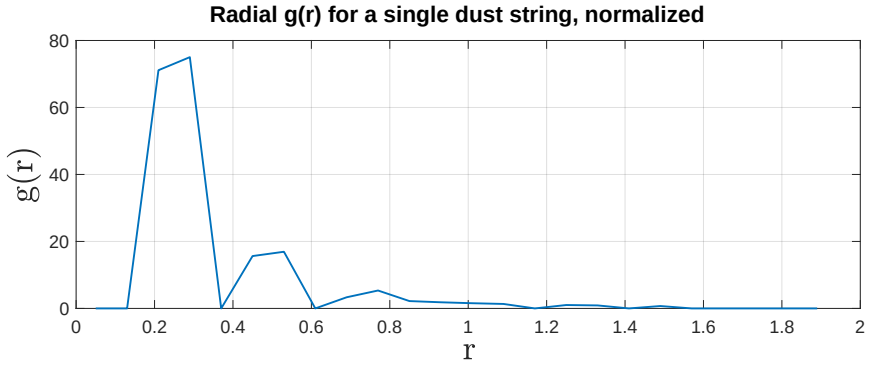
Finding values for `binWidth` and `initialBinWidthIgnore` that give more pronounced peaks therefore helps to reduce the uncertainty of the interparticle distance. Still, even for a single isolated string, the pair correlation functions in figure 5.13 demonstrate that $g(r)$ is unlikely to have perfectly pronounced peaks and valleys even for string-filled regions, because the interparticle distance is not perfectly consistent, combined with the usage of discrete and equally spaced data bins. Only two or in some cases three clearly distinguished peaks should be expected to be seen in $g(r)$ for regions containing strings with equal interparticle distance. Since in practice only the first two peaks will likely be well-defined enough, only the first two peaks will be considered in the estimate of the interparticle distance in particle strings.

Because the distance to the first and second particle in a particle string should be approximately equal, an estimate for the interparticle distance in a string, d, that will be used here is the average of the distance to the first peak and the distance between the first and second peak, which is simply equal to half of the distance to the second peak:

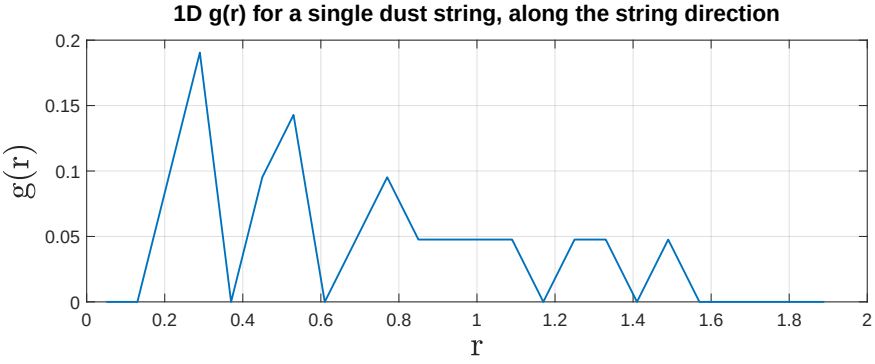
$$d = \frac{r_{\text{peak1}} + (r_{\text{peak2}} - r_{\text{peak1}})}{2} = \frac{r_{\text{peak2}}}{2}.$$



(a)



(b)



(c)

Figure 5.13: The pair correlation function $g(r)$ calculated for a single isolated dust string. Two variants of $g(r)$ were calculated: (a) and (b) are $g(r)$ based on radial, three-dimensional distance, where a) is without normalization and b) is with normalization. (c) is $g(r)$ in one dimension, along the direction of the strings. The bin width of the histograms used to calculate $g(r)$ was here 0.08 mm.

Additionally, an approximate uncertainty of a half of one bin width w_{bin} can be added, so that $d = r_{\text{peak2}}/2 \pm w_{\text{bin}}/2$. Then, in this case $g(r)$ yields

$$d = \frac{0.530\,\text{mm}}{2} \pm \frac{0.08\,\text{mm}}{2} = (0.265 \pm 0.04)\,\text{mm} = (265 \pm 40)\,\mu\text{m}, \tag{5.1}$$

which agrees with the ground truth value of $0.245\,\text{mm} = 245\,\mu\text{m}$ within the specified uncertainty.

Regarding which version of $g(r)$ to use: When studying string structures, which are one-dimensional structures, the rapid decrease of the normalized three-dimensional $g(r)$ as seen in figure 5.13b is not a helpful feature. It makes the peaks in $g(r)$, which are the indicators of interparticle distances in the string, less pronounced. In addition, since dust strings extend along a straight line (approximately), the information about the particle distribution in all directions will mainly add a great deal of 'noise' that is likely to obscure any features in $g(r)$ that would reveal the presence of string structures. For these reasons, calculating $g(r)$ for one dimension, along the direction of the strings, is deemed to be the best option for studying strings. Figure 5.13c serves as an example of $g(r)$ for an ideal realistic case of a string configuration.

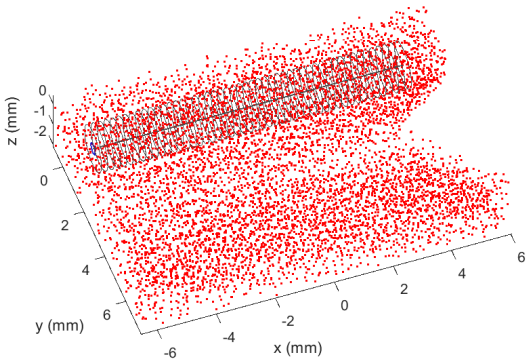
$g(r)$ for larger sets of found particles

With the above considerations in mind, a larger set of the found particles, with apparent string structures present, have been analyzed using the pair correlation function.

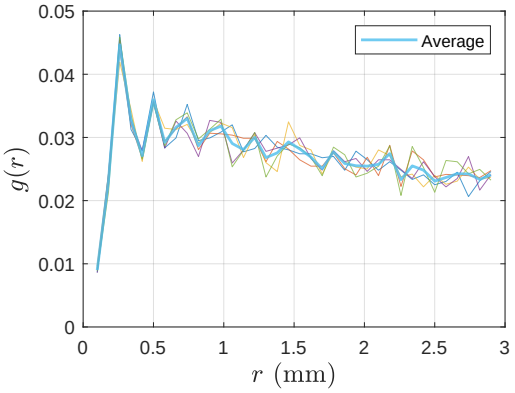
Figure 5.14 shows one-dimensional pair correlation functions along with the different regions they have been calculated for. The cylinders in the plots define the restriction volumes. All particles inside a cylinder is counted toward $g(r)$, and all other particles are not. The line in the center of a cylinder defines the axis or direction along which the one-dimensional $g(r)$ is calculated. The cylinders used in the program are programmed in such a way that they can have any placement and orientation in 3D space, which means that particle strings along an arbitrary direction can be studied by specifying an appropriate direction for the one-dimensional pair correlation function. Each cylinder is defined by a start point, a direction vector, a length and a radius. Whether a particle is inside the cylinder or not is determined by projecting their position vectors relative to the start point of the cylinder onto the center line of the cylinder. This projection vector can be used to determine whether the position is inside the confines of the cylinder both in terms of the length (displacement parallel to the cylinder center line) and the radius (displacement orthogonal to the cylinder center line).

In order to better see indications of strings, a method that has been used here is to specify a maximal distance in the plane orthogonal to the chosen direction along which $g(r)$ is calculated, `maxOrthogonalDistance` $\equiv d_{\text{maxOrth}}$. All registered distances projected onto this plane that are greater than d_{maxOrth} are discarded. This method was also used by Himpel *et al.* (2018) [35] for studying dust particle strings. They studied particle strings in a similar experimental system as in this project, and they used $d_{\text{maxOrth}} = 0.2$ mm, which will also be used here.

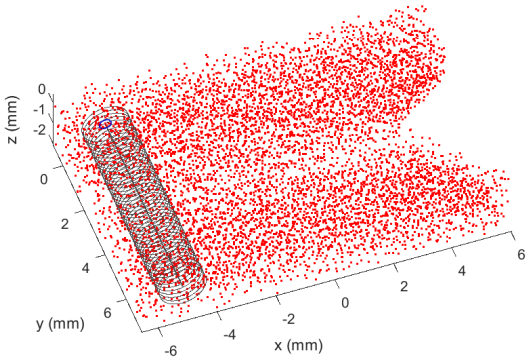
Before continuing with the analysis and interpreting the results, there is one pitfall that should be considered first, regarding the usage of d_{maxOrth} and the interpretation of peaks and valleys in $g(r)$. The main point I will make here is that it is important to not set the value of d_{maxOrth} too small. Figure 5.15 shows the same pair correlation functions as in figure 5.14, but in addition it has also calculated them using $d_{\text{maxOrth}} = 0.8$ mm. In this case this is equal to the radii of the cylinders used as the restriction volumes which define which particles to include in $g(r)$, which essentially means that there are no restrictions on the orthogonal distances other than the inherent restrictions of the cylinders themselves. In this figure it is seen that for $d_{\text{maxOrth}} = 0.2\,\text{mm}$, the pair correlation functions have a higher tendency to have a periodic pattern of peaks and valleys compared to for $d_{\text{maxOrth}} = 0.2\,\text{mm}$ in all three regions, despite



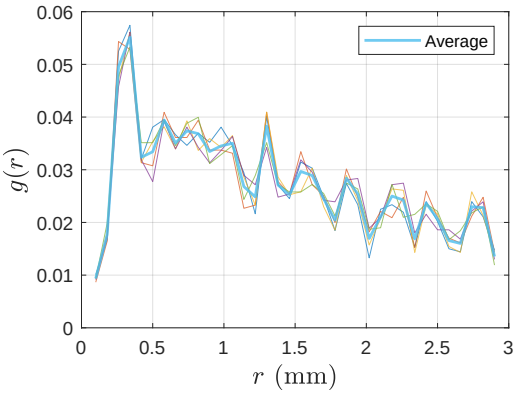
(a) Dense region, parallel with the x-axis



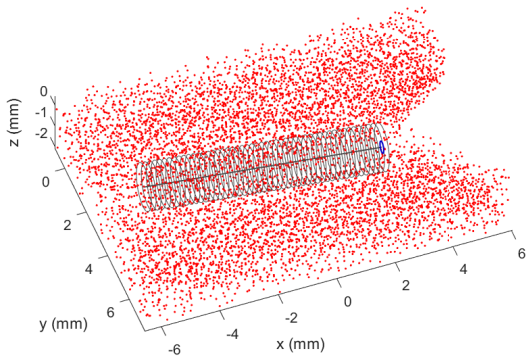
(b)



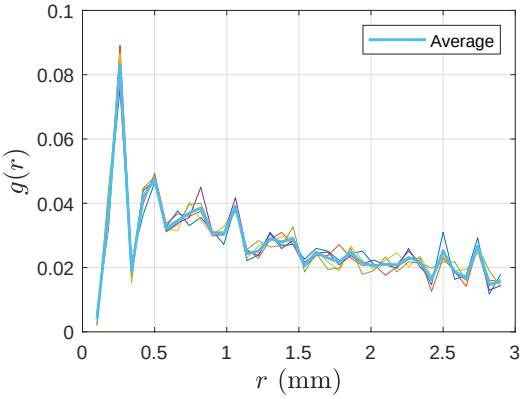
(c) Dense region, parallel with the y-axis



(d)



(e) Dilute region, parallel with the strings



(f)

Figure 5.14: Comparison of $g(r)$ for different regions in the reconstruction volume, highlighted by cylinders. (a), (c) and (e) show PSO position reconstructions of frame 2150 in the image sequence. (b), (d) and (f) show the pair correlation functions of frames 2150-2154, as well as the average of these at each point. The cylinders all have a radius of 0.8 mm. The lines in the middle of the cylinders define the direction $g(r)$ is calculated along. The maximal orthogonal distance to this direction is $d_{\text{maxOrth}} = 0.2$ mm. Note the different maximum values on the y-axes in (b), (d) and (f).

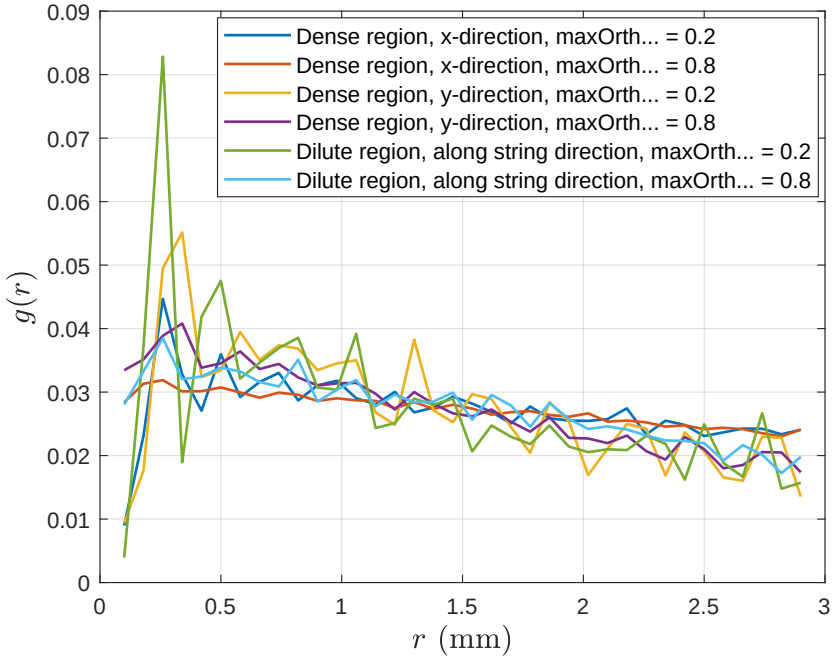


Figure 5.15: Comparision of one-dimensional pair correlation functions for different regions and directions, as in figure 5.14, but here also for two different values of the maximum orthogonal distance d_{maxOrth}. The graphs show the average $g(r)$ of frames 2150-2154.

the fact that only one of the regions has notable string formation. This tendency was also tested and confirmed with even smaller values of d_{maxOrth}. However, the pair correlation function with decidedly the most pronounced peaks and valleys is $g(r)$ for the dilute region with $d_{\text{maxOrth}} = 0.2\,\text{mm}$, which is the expected result. Figure 5.15 also demonstrates the *necessity* of using the maximum orthogonal distance restriction, since $g(r)$ for the dilute region with $d_{\text{maxOrth}} = 0.8\,\text{mm}$ does not show peaks indicating interparticle distances despite the fact that strings are present there.

This demonstrates that periodic peaks and valleys do not *necessarily* represent actual interparticle distances. Holding this assumption can lead to the wrong conclusion that strings are present in a volume even when they are in fact not. However, if relatively *pronounced* peaks and valleys are present then it can be an actual indicator of interparticle distances in the strings in the selected volume. $d_{\text{maxOrth}} = 0.2$ mm seems to be an appropriate value to use for the sets of positions analyzed in this project. In other experimental situations, if the number density is very different from this situation, and therefore the mean length scales between particles are smaller, then other values of d_{maxOrth} could be more suitable.

A time evolution of the pair correlation function between frames 2000 and 2504 is shown in figure 5.16. Figure 5.16a shows the plots of $g(r)$, where each graph is $g(r)$ averaged over the five frames indicated in the upper-right of the plot. Figure 5.16b shows the positions of the first three peaks of $g(r)$ as a function of time (frame number). Only the first three peaks are shown in figure 5.16b because only the first two or in some cases three peaks in the different plots of $g(r)$ are pronounced enough to be considered as indications of string structures. The data bin parameters that were used was `binWidth` $= 0.08\,\text{mm}$ and `initialBinWidthIgnore` $= 0.06\,\text{mm}$. The adjustment of the bin parameters were done manually by trial-and-error, so it is possible

that other parameter values would give the most well-defined peaks, with slightly different peak positions.

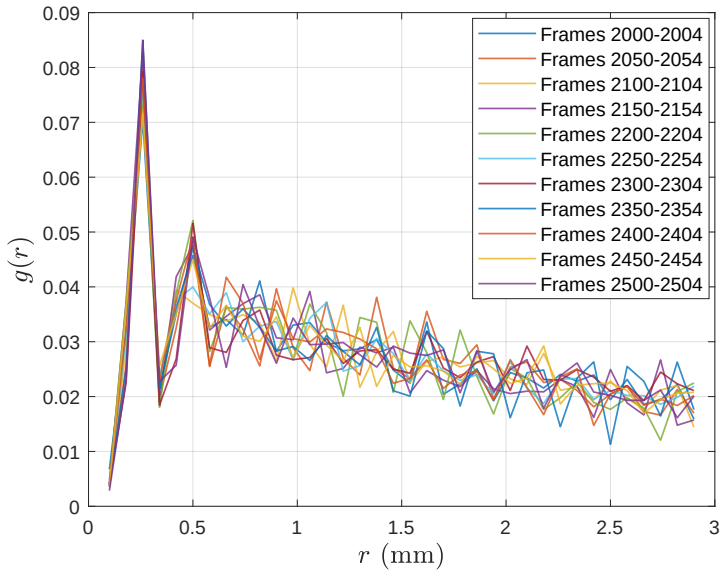
Peak no. 1 in $g(r)$ in the dilute region is pronounced for all points in time, and the second peak also remains quite pronounced. Figure 5.16b shows that the first peak remains at a constant position in time of $r = 0.260\,\text{mm}$, within one bin width. The second peak has a constant position of $r = 0.500\,\text{mm}$ for all points in time except for frame 2100.

Using the estimate for the interparticle distance in the strings from eq. (5.1), along with an uncertainty term of half of one bin width, the interparticle distance is estimated to be

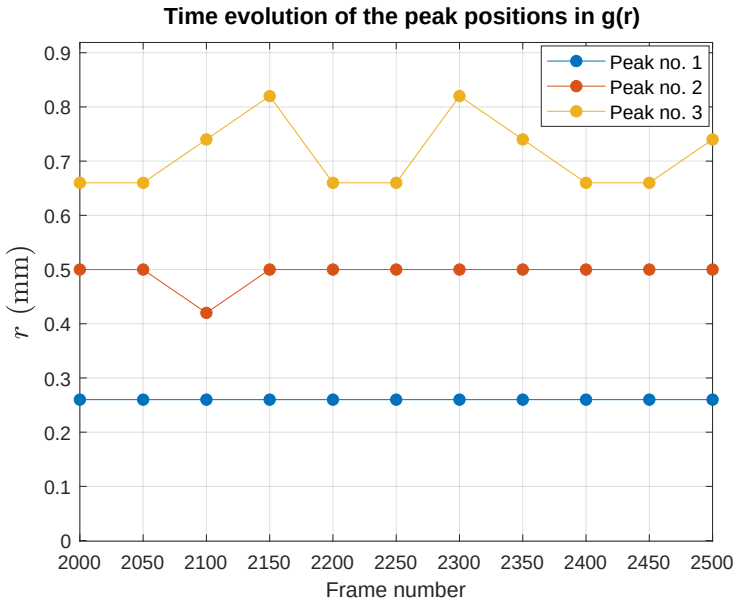
$$d = \frac{0.50\,\text{mm}}{2} \pm \frac{0.08\,\text{mm}}{2} = (0.25 \pm 0.04)\,\text{mm} = (250 \pm 40)\,\mu\text{m}.$$

This is a relatively large uncertainty of 16%. However, it represents the maximum uncertainty of the peak since the uncertainty spans the whole bin width (half of the bin in each direction). Since the bin width and the `initialBinWidthIgnore` parameter were adjusted to find the most pronounced peaks, there is some reason to believe that the result of 250 µm is somewhat closer than 40 µm to the real average interparticle distance in the observed strings.

The distance between $r = 0$ and the first peak is approximately equal to the distance between the first peak and the second peak. For this reason, the peaks can be interpreted as interparticle distances in particle strings. We can then conclude that the interparticle distance in the strings in the dilute region have an approximate interparticle distance of $d = (250 \pm 40)\,\mu\text{m}$, and that the interparticle distance stays approximately constant for the time between frames 2000 and 2500, which is equivalent to 2.5 seconds in real time.



(a) $g(r)$ averages over 5 frames each.



(b)

Figure 5.16: Time evolution of $g(r)$ in the dilute region. $g(r)$ is averaged over five frames every 50 frames between frames 2000 and 2504, giving 11 points in time. For each point in time, the restriction volume/cylinder (see figure 5.14e) is adapted to the current frame in order to only cover the dilute region. Each average $g(r)$ is plotted in (a). The corresponding first three peak locations in each $g(r)$ are plotted in (b). The bin width used when calculating $g(r)$ was 0.08 mm.

5.4 Values of plasma parameters

In order to calculate the forces acting on dust particles in the discharge, the values of various plasma parameters are required. In this subsection, the relevant plasma parameters and their values will be presented. The parameters that required to be calculated were calculated in MATLAB scripts that I developed as part of the work in this thesis.

Experiment parameter values: In the experiment, the pressure p and the RF power P of the plasma chamber are adjustable parameters. The parameters during the experiment analyzed in this thesis were $p = 30\,\text{Pa}$ and $P = 3.5\,\text{W}$.

Distance between dust particles in a string: For the interparticle distance in a dust string, the estimate based on the pair correlation functions in figure 5.16 will be used, which was $d = (250 \pm 40)\,\mu\text{m}$. For simplicity, the uncertainty term will be dropped and a value of $250\,\mu\text{m}$ will be used in the force calculations.

Dust particle velocity: In the dilute midplane region, as highlighted in figure 5.9, the dust particles are drifting in the negative x-direction. By stringing together the frames in the image series from the experiment into a video clip using a video editing software [8], the general movement of the dust particles could be studied in time. By visual inspection it was observed that a significant fraction of the dust particles in the middle of the dilute region moved with approximately constant velocity.

Then, looking at the reconstructed three-dimensional positions from the PSO algorithm for the corresponding frames, a particle in a string with a typical velocity compared to the other particles in the vicinity in the dilute region was singled out, and its position was manually tracked over 5 frames. The velocity of the dust particle was estimated by taking the vector difference of the particle position at the final frame $\mathbf{r}_{\text{final}}$ and the position at the first frame $\mathbf{r}_{\text{first}}$ and then dividing by the time Δt spanned between the two images. With a camera frame rate of 200 FPS, the time between each frame is $(1/200)\,\text{s} = 0.005\,\text{s} = 5\,\text{ms}$. In the case of 5 frames, there are 4 timesteps between them, so $\Delta t = 0.02\,\text{s}$. The velocity was found to be

$$\mathbf{v}_\text{d} = [-2.338, -0.859, 0.635] \cdot 10^{-3}\,\text{m/s} = [-2.338, -0.859, 0.635]\,\text{mm/s},$$

which corresponds to a speed of $v_\text{d} = |\mathbf{v}_\text{d}| = 0.00257\,\text{m/s} = 2.57\,\text{mm/s}$. The direction of this velocity estimate matches the chosen string direction (as seen in figure 5.14e) quite well.

This gives only a rough approximation for the typical dust velocity in a string. However, the dust velocity is required for calculating the neutral drag force, so it will enable us to get an order of magnitude estimate of the neutral drag.

Additionally, the dust velocities were estimated for two particles at the beginning of the dense region. These velocities were found to be $\mathbf{v}_1 = [-0.349, -1.005, -0.940]\,\text{mm/s}$, which corresponds to a speed of $v_1 = 1.42\,\text{mm/s}$, and $\mathbf{v}_2 = [-0.3135, -0.4644, -1.0607]\,\text{mm/s}$, which corresponds to a speed of $v_2 = 1.20\,\text{mm/s}$, respectively. Thus, these dust particles are on average slower than the estimated particle speed in the dilute region by roughly a factor of 0.5, and their velocity directions have relatively larger y- and z-components. The change in velocity direction of the particles entering the dense region happens because they are entering the slowly rotating vortex cycle of the dust cloud. This observed change in velocity will be discussed in section 5.5.

Dust particle mass and ion mass: The mass of a spherical MF dust particle of diameter $2a = 6.38\,\mu\text{m}$ is $m_\text{d} = \rho_\text{MF} \cdot \frac{4}{3}\pi r_\text{d}^3 \approx 2.04 \cdot 10^{-13}\,\text{kg}$.

The ion species for these experiments is argon gas atoms. As was mentioned in section 2.1.4, the mass of an argon atom is $m_\text{Ar} = 39.95u \approx 6.63 \cdot 10^{-26}\,\text{kg} = m_\text{i}$. Technically, the argon

mass has one electron mass subtracted from it, but the electron mass is $m_\mathrm{e} = 5.486 \cdot 10^{-4} u = 9.109 \cdot 10^{-31}\,\mathrm{kg}$, which is a negligible loss compared to the mass of the argon atom.

Electron and ion temperatures: The values for the ion and electron temperatures are unknown for the exact physical parameters of this experiment. Himpel *et al.* (2018) used values from a SIGLO2D simulation for the electron and ion temperatures, and these values were $T_\mathrm{e} = 3\,\mathrm{eV}$ and $T_\mathrm{i} = 0.03\,\mathrm{eV}$, giving a temperature ratio of $T_\mathrm{e}/T_\mathrm{i} = 100$ [35]. Since the experimental setup of those experiments used the IMPF-K2 chamber with low pressure and low power, I will use these values for the electron and ion temperatures. These values were also used in simulations by Piel (2011) when testing his particle alignment model due to ion drag forces in RF plasmas with subsonic ion flows [66], so these temperatures are not uncommon for RF dusty plasma experiments.

What is meant by expressing a temperature T in electron volts, which is a unit of energy, is that it is the temperature that is equivalent to an energy of $k_\mathrm{B}T$. In kelvin, the electron and ion temperatures are $T_\mathrm{e} = (3\,\mathrm{eV})/k_\mathrm{B} = 34814\,\mathrm{K}$ and $T_\mathrm{i} = 348.14\,\mathrm{K}$.

Electron and ion number densities: The electrion and ion number densities, n_e and n_i, also have unknown values that cannot be easily calculated. By similar reasoning as for the electron and ion temperatures, I will use the values of n_e and n_i that were used by Himpel *et al.* (2018), which were $n_\mathrm{e} = n_\mathrm{i} = 10^{15}\,\mathrm{m}^{-3}$ [35]. This value for n_i was also used by Piel (2011), while the electron number density was not specified [66].

Floating potential of dust particles: For the floating potential of the dust particles, the 'rule-of-thumb' expression in eq. (2.18) for a temperature ratio of $T_\mathrm{e}/T_\mathrm{i} = 100$ will be used. With $T_\mathrm{e} = 3\,\mathrm{eV} = 34814\,\mathrm{K}$, the resulting floating potential is

$$\phi_\mathrm{fl} = -2.414\, k_\mathrm{B} T_\mathrm{e}/e = -7.242\,\mathrm{V}.$$

Dust charge: The dust particle charges are estimated by the approximate formula for the dust charge given in eq. (2.21). With a particle diameter of $2a = 6.38\,\mu\mathrm{m}$ and electron temperature $T_\mathrm{e} = 3\,\mathrm{eV}$, an estimate of the dust charge is calculated to be

$$Q_\mathrm{d} = -(1400 a_{\mu\mathrm{m}} T_{\mathrm{e,eV}})e = -13\,398\, e = -2.147 \cdot 10^{-15}\,\mathrm{C}.$$

Ion thermal velocity: Based on the ion mass $m_\mathrm{i} = 6.63 \cdot 10^{-26}\,\mathrm{kg}$ and the ion temperature $T_\mathrm{i} = 0.03\,\mathrm{eV} = 348.14\,\mathrm{K}$, the thermal velocity of the ions is calculated using eq. (2.32) to be

$$v_\mathrm{th,i} = \sqrt{\frac{8 k_\mathrm{B} T_i}{\pi m_i}} = 429.5\,\mathrm{m/s}.$$

Ion acoustic speed: The ion acousting speed is calculated using eq. (2.4). With electron temperature $T_\mathrm{e} = 3\,\mathrm{eV}$ and $m_i = 39.95u$, the ion acoustic speed is

$$C_\mathrm{s} = \sqrt{\frac{k_\mathrm{B} T_\mathrm{e}}{m_\mathrm{i}}} = 2692\,\mathrm{m/s}.$$

Effective Debye length of streaming ions, and ion drift velocity: The effective screening length of the plasma has been estimated using eq. (2.5), which takes streaming ions into account. The drift speed u_i of the ions is unknown. For this reason, the Debye length has been calculated for ion stream velocities ranging between $M = 0$ and $M = 2$, where M is the

Mach number of the ion speed and $M = 1$ represents $u_\mathrm{i} = C_\mathrm{s}$, the ion acoustic speed. The resulting screening lengths are shown in figure 5.17.

Some example values from this plot is $\lambda_\mathrm{D} = 100\,\mu\mathrm{m}$ at $M = 0.2$, $\lambda_\mathrm{D} = 200\,\mu\mathrm{m}$ at $M \approx 0.5$ and $\lambda_\mathrm{D} = 300\,\mu\mathrm{m}$ at $M \approx 0.85$. The Debye length converges toward the electron Debye length as $M \to 2$, which, using eq. (2.2), is calculated to be $407\,\mu\mathrm{m}$.

Himpel *et al.* (2018) calculated an estimated screening length in the dilute region of a similar setup using the IMPF-K2 to be $\lambda_\mathrm{D} = 300\,\mu\mathrm{m}$ [35]. They also calculated the interparticle distance in the dilute region of the plasma discharge, which was also observed to contain strings, to be $d \approx 240\,\mu\mathrm{m}$. The ion drift speed was not specified in that case, but is likely to be subsonic, since Arp *et al.* (2012) observed subsonic flows in low-pressure, low-power plasmas using the IMPF-K2, which is the same setup as in the experiments in this thesis and in [35]. Furthermore, the analyzed region of the dust cloud is far from the sheath edge of the walls and electrodes where ions are being accelerated towards the wall to reach Bohm velocity, so the claim of a subsonic ion flow is reasonable [67]. The experimental parameters in this thesis is not identical to the ones in Himpel *et al.*, so the Debye length in our case is not necessarily equal to the Debye length found for that experiment.

The ratio between the interparticle distance and the Debye length in Himpel *et al.* was $d/\lambda_\mathrm{D} = 240\,\mu\mathrm{m}/300\,\mu\mathrm{m} = 0.8$. If the same ratio should be the case for the experiment analyzed in this thesis as well, the Debye length would be $\lambda_\mathrm{D} = d/0.8 = 250\,\mu\mathrm{m}/0.8 = 312.5\,\mu\mathrm{m}$. This Debye length is seen to occur for $M \approx 0.9$ in figure 5.17. Whether there is an exact connection between the effective Debye length and the interparticle distance in a dust string in an ion flow or not is unknown, so determining the ion flow velocity this way is questionable. Though, it is not entirely unreasonable that a downstream dust particle might prefer to align itself at a downstream position where the electric field from the upstream dust particle is sufficiently shielded. This *could* be for 0.8 Debye lengths, or slightly less than one Debye length. Since the experiment analyzed here and the experiment analyzed by Himpel *et al.* are quite similar, this makes it more likely that the ion drift speed is between $M = 0.5$ and $M = 1$, which are equivalent of effective Debye lengths between $\lambda_\mathrm{D} \approx 200\,\mu\mathrm{m}$ and $\lambda_\mathrm{D} \approx 325\,\mu\mathrm{m}$, rather than $M < 0.5$, which corresponds to effective Debye lengths $\lambda_\mathrm{D} \lesssim 200\,\mu\mathrm{m}$.

Neutral temperature: The temperature of the neutrals will be assumed to be $T_\mathrm{n} = 300\,\mathrm{K}$, which is around room temperature. The plasma chamber containing the gas is kept at room temperature during the experiments, and additionally it is not far off from the ion temperature $T_\mathrm{i} \approx 348\,\mathrm{K}$.

Neutral thermal velocity: Using the neutral mass $m_\mathrm{n} = m_\mathrm{Ar} = 39.95u$ and the neutral temperature $T_\mathrm{n} = 300\,\mathrm{K}$, the thermal velocity of the neutrals is calculated using eq. (2.32):

$$v_\mathrm{th,n} = \sqrt{\frac{8k_\mathrm{B}T_i}{\pi m_i}} = 398.7\,\mathrm{m/s},$$

which is comparable to the thermal velocity of the ions.

Neutral number density: The neutral number density is the number density of neutral (non-ionized) argon atoms in the plasma. The number density is used for calculating the neutral drag force. As shown in appendix B, the Epstein formulas for the neutral drag force, which are commonly used formulas for calculating neutral drag forces in dusty plasma contexts [57, 17], have the ideal gas assumption for neutrals, $p = n_\mathrm{n}k_\mathrm{B}T_\mathrm{n}$, built-in. Using this justification, the neutral number density is calculated using the ideal gas law. With $p = 30\,\mathrm{Pa}$ and $T_\mathrm{n} = 300\,\mathrm{K}$, the neutral number density is found to be

$$n_\mathrm{n} = \frac{p}{k_\mathrm{B}T_\mathrm{n}} = 7.243 \cdot 10^{21}\,\mathrm{m}^{-3}.$$

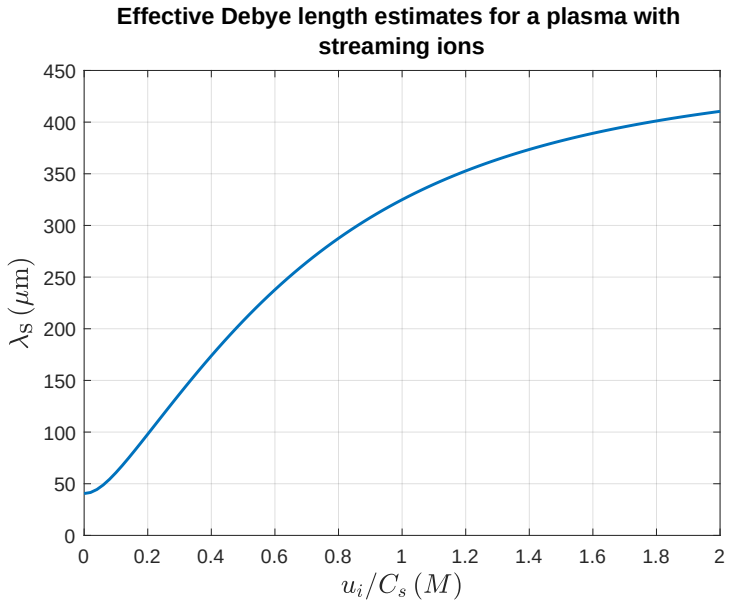


Figure 5.17: The effective screening length of an argon plasma containing streaming ions for different ion drift speeds u_i, based on eq. (2.5). Plasma parameter values: $T_\mathrm{e} = 3\,\mathrm{eV}$, $T_\mathrm{i} = 0.03\,\mathrm{eV}$, $m_\mathrm{i} = m_\mathrm{Ar} = 39.95u$, $n_\mathrm{e} = n_\mathrm{i} = 10^{15}\,\mathrm{m}^{-3}$. Ion acoustic speed: $C_\mathrm{s} = 2.69{\cdot}10^3\,\mathrm{m/s}$.

Dust particle number density: The number density of the dust particles in a specific region of the plasma chamber is estimated by dividing the number of particles in the selection, N_sel, by the volume V_sel that the particles in the selection spans. The volume V_sel of a three-dimensional rectangular selection of particles is determined in a script by inputting a selection of particles and setting the corners of the selection volume to be the maximum and minimum values of the x-, y- and z-coordinates among the selected particles, and then multiplying the side lengths together.

Due to the values of the fraction of found particles (`foundFraction`) and the fraction of ghost particles (`ghostFraction`) from the testing of the PSO algorithm on realistic particle sets, the number density obtained from the calculation based on the returned positions from the PSO algorithm will naturally also be slightly inaccurate compared to the real value of the number density. A correction to the calculated number density based on the positions returned by the PSO algorithm can be performed by using estimates for the accuracy parameters `foundFraction` and `ghostFraction`. Rename these parameters as `foundFraction` $\equiv f$ and `ghostFraction` $\equiv g$. If we first assume that the values of f and g are known exactly, the correction comes from the following relation:

$$n_\mathrm{PSO} \cdot \frac{(1-g)}{f} = n_\mathrm{GT}, \tag{5.2}$$

where $n_\mathrm{PSO} \equiv N_\mathrm{PSO}/V$ is the number density of the returned PSO positions (total number of returned PSO positions, N_PSO, divided by the reconstruction volume V) and $n_\mathrm{GT} \equiv N_\mathrm{GT}/V$ is the number density of the ground truth positions (total number of ground truth positions, N_GT, divided by the reconstruction volume V). A proof of eq. (5.2) can be found in appendix A. For the real data set, f and g are not known, and therefore the results from the test performed in section 5.1.2 on the realistic artificial particle set will be used as estimates for the accuracy parameter values for the real images.

The correction in eq. (5.2) was tested on the PSO results for an artificial particle set (similar to figure 4.1c). The ground truth particle density in the entire reconstruction volume was 4480, and the volume was $V = 224\,\mathrm{mm}^3$, giving a ground truth particle number density of $n_{\mathrm{GT}} = 20\,\mathrm{mm}^{-3}$. 3620 particles were returned by PSO, leading to a particle density of $n_{\mathrm{PSO}} = 15.0\,\mathrm{mm}^{-3}$. Using the correction, with $g = 0.19$ and $f = 0.62$, the estimate for the ground truth number density was $n_{\mathrm{GT,est}} = n_{\mathrm{PSO}} \cdot (1 - g)/f = 19.6\,\mathrm{mm}^{-3}$, which is significantly closer to the ground truth number density of $n_{\mathrm{GT}} = 20\,\mathrm{mm}^{-3}$.

This correction is based on the assumption that the PSO found fraction and ghost fraction for the real images are the same as for the artificial particle sets. When running PSO on artificial particle sets and images that are designed to look as similar as possible to the real images (with respect to particle width, particle intensity, random intensity fluctuations, image noise and particle number density), the ghost fraction and found fraction in the results from the artificial particle set will likely be quite similar to the results when analyzing real images, where ground truth positions are not available. Still, there is most likely at least some discrepancy between the PSO accuracy for the real images compared to the artificial images, which should be taken into consideration.

An additional correction that was made was to multiply the selected volume V_{sel} by a factor C which is slightly larger than 1 if the selected volume is smaller than the entire volume V (which is true in all cases except for when all particles are selected). This correction was done with the purpose of increasing the accuracy of the calculated number density for smaller selections of particles, since any empty space between the selected particles and the nearest neighbours to the selected volume is not taken into account. This makes the selected volume somewhat too small, resulting in the calculated number density being slightly too high, especially for smaller selection volumes. The correction factor C is not large; its maximum value is $C = C_{\max} = 1.05$ for $V_{\mathrm{selected}} \ll V_{\mathrm{total}}$, and it decreases linearly until $V_{\mathrm{selected}} = V_{\mathrm{total}}$, where $C = C_{\min} = 1$. This value of $C_{\max}$ was chosen by calculating the number density using this correction for multiple different selection volumes for different values of $C_{\max}$, and $C_{\max} = 1.05$ gave good corrections for a wide range of selection volumes.

Since the correction factor C is calculated based on a very simple linear model, it might not give the best corrections to the number density for all volume selections. However, it was shown through testing to be an improvement over the number density calculated without this correction.

Using these corrections, the number density of dust particles has been calculated for the returned PSO positions from analyzing the real images. The dust number density in frame 2150 of the dataset was calculated to be $n_d = 2.7 \cdot 10^{10}\,\mathrm{m}^{-3}$. This is within an order of magnitude of the approximate number density of the estimated value of $n_d \approx 10^{11}\,\mathrm{m}^{-3}$ in [35], where the experimental setup was similar to the experiments in this thesis.

5.5 Estimates for forces on dust particles in a string

Using the estimates for the physical parameters calculated in the preceding subsection, the various forces on dust particles in a string, as presented in section 2.3.2, will now be estimated.

Longitudinal ion drag force: The ion drag force on a dust particle in a subsonic ion flow is calculated using the Barnes model, eqs. (2.34) and (2.37). In the context of a dust string this can be called a *longitudinal* ion drag force in the sense that it acts along the direction of the dust string, which here will be assumed to be aligned with the ion drift velocity.

Since the exact ion drift speed is unknown, the ion drag is calculated for ion drift speeds

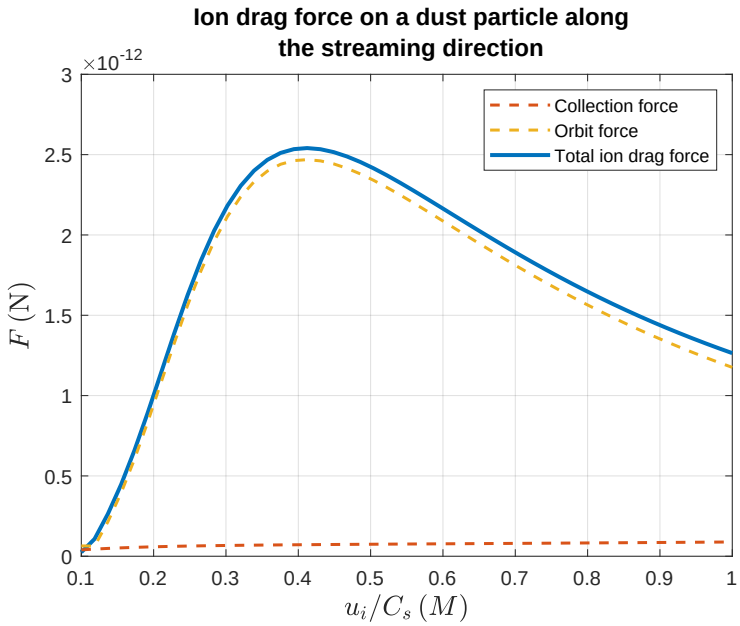


Figure 5.18: Longitudinal ion drag force for subsonic ion drift velocities, calculated using the Barnes model. This force acts in the longitudinal direction with respect to the axis of the string direction. The force in the graph represents the magnitude of the ion drag force vector, where the force vector has the same direction as the ion drift velocity.

between $M = 0.1$ and 1. The results are shown in figure 5.18. The ion drag force is seen to have a magnitude in the order of $10^{-12}\,\mathrm{N} = 1\,\mathrm{pN}$. It is seen that the orbit force is the dominant contributor to the ion drag force compared to collection force, which is relatively weak for subsonic ion drift velocities. This is the expected result for the Barnes model of the ion drag for a subsonic ion stream [57, p. 37]. At large ion drift speeds $u_i \gg C_s$, the collection force becomes dominant and the orbit force becomes negligible.

Neutral drag force: The neutral drag force on a dust particle was calculated using both eq. (2.29) and eq. (2.30). As previously mentioned, the neutral number density n_n, which is required in eq. (2.29), was estimated with the ideal gas law for the neutrals.

Using eq. (2.29) with $\delta = 1.44$, $2a = 6.38\,\mu\mathrm{m}$, $m_\mathrm{n} = 39.95u$, $v_{\mathrm{th,n}} = 398.7\,\mathrm{m/s}$, $n_\mathrm{n} = 7.243 \cdot 10^{21}\,\mathrm{m}^{-3}$ and $\mathbf{v}_\mathrm{d} = [-2.338, -0.859, 0.635] \cdot 10^{-3}\,\mathrm{m/s}$, the neutral drag force was calculated to be

$$|\mathbf{F}_\mathrm{n}| = \left| -\delta \frac{4}{3} \pi a^2 m_\mathrm{n} v_{\mathrm{th,n}} n_\mathrm{n} \mathbf{v}_\mathrm{d} \right| = 3.023 \cdot 10^{-14}\,\mathrm{N}.$$

Using eq. (2.30) with $\rho_\mathrm{d} = 1500\,\mathrm{kg/m}^{-3}$ and $p = 30\,\mathrm{Pa}$, the exact same result was obtained. This force is directed in the direction opposite of the dust velocity, so the force along the string axis is $F_\mathrm{n} = -3.023 \cdot 10^{-14}\,\mathrm{N}$, where the positive direction is defined to be equal to the dust velocity direction. For dust particles in a string, since the dust velocity most often coincides with the string direction in the dilute region (and therefore also with the direction of the ion stream), the neutral drag force acts as a resistance force to the the movement of the individual particles, and to the movement of the string as a unit, along the string axis.

As discussed in section 2.3.2 and appendix B, the ideal gas law for the neutral gas,

$p = n_n k_B T_n$, is inherent in eq. (2.30). This is not an immediately obvious assumption to make since a plasma is not an ideal gas, because a plasma is very much interacting with itself. It *might* be a reasonable assumption since the neutrals aren't affected by electromagnetic forces and because they are by a large margin the dominant species in the gas.

If it turns out, for other experimental situations for dusty plasmas, that the neutral number density is different than the ideal gas case, then I would argue that eq. (2.29) is likely to give a more accurate neutral drag force, since it does not contain inherent assumptions about the neutral number density. Regardless, eq. (2.30) is a commonly used version of the neutral drag force in dusty plasmas [57], and has been used for the force analysis of similar experimental setups, also with the IMPF-K2 plasma chamber [36], so I consider the ideal gas assumption for the neutral number density to be an acceptable approximation in this case.

Electrostatic repulsion between particles in a string: Based on the estimate of the dust charge Q_d, the estimate of the effective Debye length of the plasma, and the interparticle distance from the pair correlation functions, an estimate for the electrostatic repulsive force between dust particles can be calculated by using eq. (2.25).

Figure 5.19 shows the electrostatic force on a dust particle that is located in a dust string between two other equally charged dust particles with $Q_d = -13\,398\,e$. The three particles are all situated along one axis. The axis direction is arbitrary, and has here been set to be the x-axis. The two fixed particles are located at $\pm d$, where $d = 250\,\mu m$ is the estimated interparticle distance in the string. An ion stream is imagined to flow in the positive x-direction, so that the dust grain at $x = -d$ can be referred to as the upstream particle and the dust grain at $x = +d$ can be referred to as the downstream particle. Since all three particles are assumed to be located on a straight line, the force F_x is effectively a one-dimensional force. $F_x > 0$ corresponds to a force in the positive x-direction, and vice versa.

It is seen that the repulsive forces act as a restoring force on the middle dust particle toward an equilibrium point situated at the midpoint between the two fixed dust particles at $x = 0$, where the sum of the repulsive forces is $F_x = 0$. The force is seen to range from $F_x = 0$ at the equilibrium point, and it increases to $F_x \sim 10^{-13}\,N$ at $|x| \approx 10\,\mu m$, $F_x \sim 10^{-12}\,N$ at $|x| \approx 100\,\mu m$, $F_x \sim 10^{-11}\,N$ at $|x| \approx 200\,\mu m$, and keeps increasing as the middle particle gets closer to either of the other two particles. It is seen that this force exceeds that of the neutral drag force, $|F_n| \sim 10^{-14}\,N$, even for small displacements, and exceeds the ion drag in magnitude for $|x| \gtrsim 100\,\mu m$. The magnitude of the electrostatic force from *one* of the other two dust particles at the equilibrium point is $|F_x| \approx 5 \cdot 10^{-13}$. This means that the electrostatic repulsive force exerts a relatively strong force on a dust particle in a string from each particle when it comes to magnitude, though the net force is zero for equal dust charges. The fact that the force is relatively strong and that there is an equilibrium in the midpoint between the particles suggests that this force could be a contributing force that creates consistent interparticle distances in a string, which is observed for many of the strings in the dilute region. Though, the reason for why there is a consistent interparticle distance must include other forces in addition to the electrostatic repulsive force, since there must be attractive forces involved that keeps the particles somewhat fixed in the longitudinal direction relative to each other. This is what the ion drag forces and the wakefield force are theorized to do.

As mentioned in section 2.4, simulations performed by Miloch *et al.* (2012) [60] led to the conclusion that the charge distribution in a string is likely to be unequal for the different dust particles, and that the change is non-monotonous (the charge does not increase/decrease monotonously for each subsequent dust particle in the string). This means that assuming an equal dust charge on all particles in a string is not entirely realistic, and this would in turn affect the electrostatic forces between particles in a string and also shift the longitudinal equilibrium points. Dust strings consisting of 4 or more particles were not studied in their

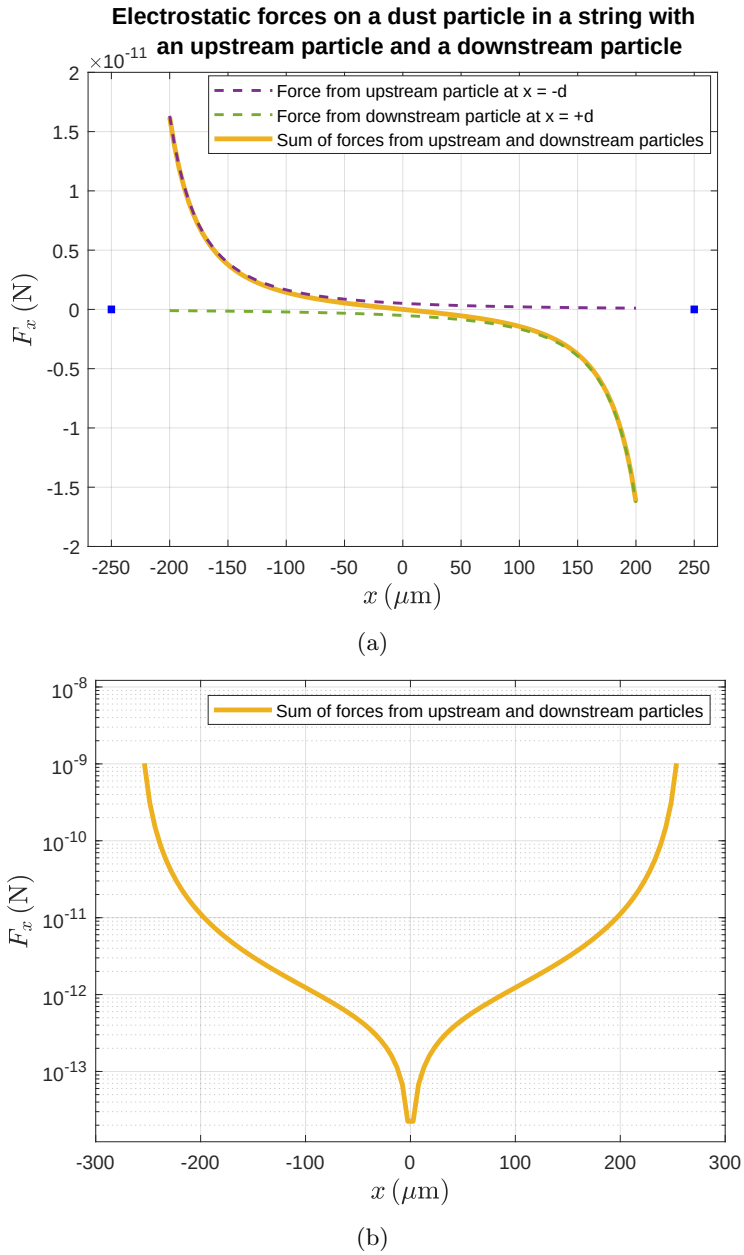


Figure 5.19: The electrostatic forces on a dust particle with an upstream particle and a downstream particle. The ions are assumed to flow in the positive x-direction, so that the fixed particle at $x = -d$ is the upstream particle and the particle at $x = +d$ is the downstream particle, where $d = 250\,\mu\text{m}$ is the interparticle distance of the string. The charge on each dust particle is $Q_\text{d} = -13\,398\,e$. (a) The electrostatic forces of the upstream and downstream particle between $x = -0.8d = -200\,\mu\text{m}$ and $x = +0.8d = 200\,\mu\text{m}$. The force direction is defined so that $F_x > 0$ acts in the positive x-direction. The dark blue squares at $x = \pm 250\,\mu\text{m}$ represent the upstream and downstream dust particles. (b) Same forces as calculated in (a), but with the the absolute value of the net force plotted on a log scale and the x-axis extending out to $x = \pm(d - 2a)$, where $a = 6.38\,\mu\text{m}$ is the radii of the dust particles. These are the points where the middle dust particle touches the the upstream or downstream particle.

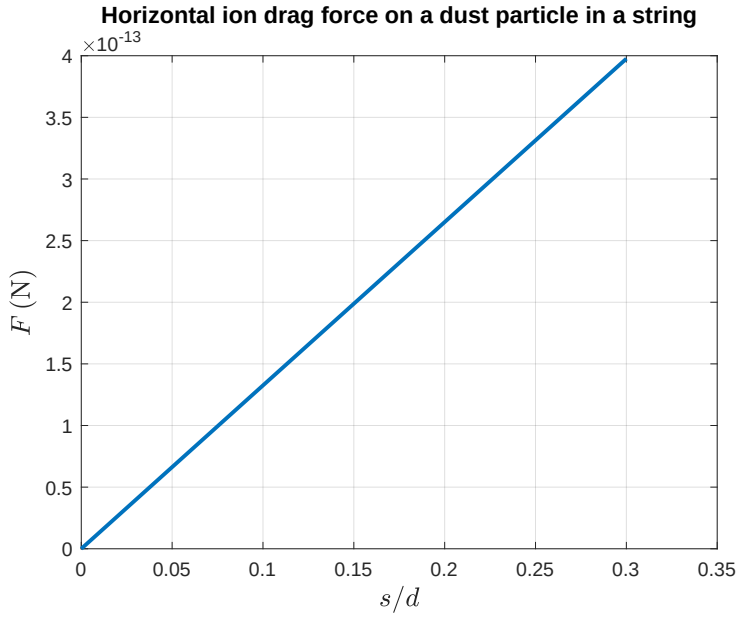


Figure 5.20: Estimates for the magnitude of the transverse or horizontal restoring ion drag force in subsonic ion flows at different displacements s from the equilibrium point $s = 0$ situated on the dust string axis. The force acts toward the string axis. The interparticle distance used in these calculations is $d = 250\,\mu\text{m}$. Calculated from the ion drag model by Piel (2011) [66].

simulations, and has to my knowledge not been studied by other researchers, so detailed knowledge about the charge distribution in longer strings is unknown.

Thermophoretic force: For the low RF power and low pressure conditions in the IMPF-K2 chamber in this experiment, the temperature gradient in the plasma chamber is negligible [36, p. 70] [30]. So, in this analysis, the thermophoretic force will be neglected.

$$\mathbf{F}_{\text{th}} \approx \mathbf{0}.$$

Transverse restoring ion drag force: Estimates for the transverse restoring ion drag force toward the axis of the dust string were calculated using eq. (2.39) from Piel (2011) [66]. The values used were $Q_{\text{d}} = -13\,398\,e$ and $d = 250\,\mu\text{m}$. The results are shown in figure 5.20. The distance s is the displacement from the string axis, and d is the interparticle distance of the string. The equation is only valid for displacements $s \ll d$, so only values up to $s = 0.3d$ have been included in the plot.

The equation for the transverse restoring force in eq. (2.39) assumes equal dust charge on the upstream particle and the downstream particle. As discussed in section 2.4, the dust charge on the different particles in the string are not equal for the particles in the string due to decharging in the ion focus regions, so the transverse restoring force estimate based on eq. (2.39) will most likely be slightly inaccurate compared to the real values.

Residual acceleration force: Based on the maximum magnitude of the residual acceleration of the analyzed data, which was $a_{\text{max}} = 0.018g = 0.177\,\text{m/s}^2$, along with the mass of a dust particle, the magnitude of the equivalent force of the residual acceleration is calculated to be

$F_{\mathrm{acc,max}} = m_{\mathrm{d}} a_{\mathrm{max}} = 3.60 \cdot 10^{-14}\,\mathrm{N}$.

5.5.1 Sum of forces and comparison between forces

Using the estimates of the forces acting on a dust particle in a string, I will now consider the sum of the forces on such a dust particle. I will study the case of a dust particle with constant velocity based on the previously mentioned observation that a considerable amount of the dust particles in the middle part of the dilute region, including entire strings in some cases, have approximately constant velocity. There are some remaining forces that are unknown, namely the wakefield force, the electric field force and the electric polarization force. The electric field force is unknown because the electric field in the dilute region is unknown, and the polarization force is unknown because the gradient of the effective shielding length is unknown. Investigating the case of a stable (non-accelerating) string can give insight into the magnitude of the remaining unknown forces.

Using the assumption of zero acceleration, Newton's second law gives the following:

$$\Sigma \mathbf{F} = \mathbf{F}_{\mathrm{acc}} + \mathbf{F}_E + \mathbf{F}_{\mathrm{n}} + \mathbf{F}_{\mathrm{drag,i}} + \mathbf{F}_{\mathrm{w}} + \mathbf{F}_{\mathrm{th}} = \mathbf{0}.$$

$\mathbf{F}_{\mathrm{acc}}$ is equivalent force due to the residual acceleration during the parabolic flight experiment. $\mathbf{F}_E = \mathbf{F}_{\mathrm{es,part}} + \mathbf{F}_{\mathrm{es,E}} + \mathbf{F}_{\mathrm{pol}}$ is the sum of the electric field forces, where $\mathbf{F}_{\mathrm{es,part}}$ is the electrostatic forces from the electric fields of other dust particles, $\mathbf{F}_{\mathrm{es,E}}$ is the electric force due to other electric fields in the chamber and $\mathbf{F}_{\mathrm{pol}}$ is the electric polarization force. $\mathbf{F}_{\mathrm{n}}$ is the neutral drag force, $\mathbf{F}_{\mathrm{drag,i}} = \mathbf{F}_{\mathrm{i,long}} + \mathbf{F}_{\mathrm{i,tr}}$ is the total ion drag force, which is the sum of the longitudinal and the transverse ion drag components, $\mathbf{F}_{\mathrm{w}}$ is the wakefield force and $\mathbf{F}_{\mathrm{th}}$ is the thermophoretic force.

As previously mentioned, the thermophoretic force is negligible in this experiment, $\mathbf{F}_{\mathrm{th}} \approx \mathbf{0}$. We will consider the case of a stable dust string, so the assumption will be made that the dust particle is positioned with no displacement from the string axis, which gives zero transverse ion drag force, $\mathbf{F}_{\mathrm{i,tr}} \approx \mathbf{0}$. If we make the assumption that the dust charges in the string are equal, the interparticle electrostatic forces from the upstream and downstream particle will cancel each other out, so that $\mathbf{F}_{\mathrm{es,part}} = 0$.

While the residual acceleration force $\mathbf{F}_{\mathrm{acc}}$ *can* have some non-negligible influence on the dust dynamics with a maximal magnitude of $3.610^{-14}\,\mathrm{N}$ in the analyzed image sequence, the main usage of the reconstructed positions has been to calculate the pair correlation function and to estimate the dust velocity. The dust velocity is only used in the calculation of the neutral drag force, whose magnitude depends linearly on the dust velocity. If the dust velocity was estimated wrong by a factor of 2, this would be equivalent to the neutral drag force being estimated wrong with a factor of 2 as well. This would still lead to a neutral drag force that is between one and two orders of magnitude weaker than the ion drag force, meaning that it would still be a relatively weak force. The pair correlation function is not affected to a large degree by small movements of the frame as a whole, since it only takes the relative positions between particles into account. The other forces are calculated based on estimations of the various physical parameters of the plasma, and are not based directly on the analysis of positions or trajectories. For these reasons, $\mathbf{F}_{\mathrm{acc}}$ will be neglected in the following force analysis.

The remaining terms in Newton's second law are only in the longitudinal direction, along the strings, so the force equation becomes one-dimensional. The remaining terms are:

$$F_{\mathrm{es,E}} + F_{\mathrm{pol}} + F_{\mathrm{n}} + F_{\mathrm{i,long}} + F_{\mathrm{w}} = 0$$
$$\Updownarrow$$
$$F_{\mathrm{es,E}} + F_{\mathrm{pol}} + F_{\mathrm{w}} = -(F_{\mathrm{i,long}} + F_{\mathrm{n}}).$$

The magnitude of the ion drag force only becomes relatively small at $M \lesssim 0.15$. If $M < 0.15$ then the effective Debye length is $\lambda_{\mathrm{D}} \lesssim 75\,\mu\mathrm{m}$, which is much smaller than the effective Debye length of the similar, though not equal, experimental conditions of Himpel *et al.* (2018), who found an effective Debye length of $\lambda_{\mathrm{D}} \approx 300\,\mu\mathrm{m}$. As previously discussed, it is more likely that the ion drift speed is between $M = 0.5$ and $M = 1$. For these ranges of the ion drift speed, the ion drag force will have a magnitude in the order of $10^{-12}\,\mathrm{N}$. As a concrete example, for $M = 0.75$ the ion drag force is $F_{\mathrm{i,long}} = 1.765 \cdot 10^{-12}\,\mathrm{N}$. The neutral drag force was estimated to be $F_{\mathrm{n}} = -3.023 \cdot 10^{-14}\,\mathrm{N}$. Then, $F_{\mathrm{i,long}} + F_{\mathrm{n}} = 1.735 \cdot 10^{-12}\,\mathrm{N}$. This gives

$$F_{\mathrm{es,E}} + F_{\mathrm{pol}} + F_{\mathrm{w}} = -1.735 \cdot 10^{-12}\,\mathrm{N}.$$

The sum of the electric field force, the polarization force and the wakefield force must then provide the resistive force of magnitude $1.735 \cdot 10^{-12}\,\mathrm{N}$ in the direction opposite of the string movement and ion flow, otherwise, the particles would be accelerated toward the wall.

Because the total electric field is unknown in the dilute region, it is possible that an electric field force accelerates the ion stream in such a way that the ion drift speed gets a spatial dependence, meaning that $\nabla u_{\mathrm{i}} \neq \mathbf{0}$. In that case, the effective Debye length for streaming ions in eq. (2.5) will also have a non-zero gradient since it is dependent on u_{i}. This would give rise to the electric polarization force, as in eq. (2.26). An acceleration of the ion stream would also change the longitudinal ion drag force since it depends on the ion drift velocity. More detailed knowledge about the electric fields and the ion drift velocity would be required in order to gain more detailed knowledge about which forces contribute the most to the resistive force acting on the dust particles in the dilute region.

As mentioned in section 2.3.5, Ludwig *et al.* (2012) [54] showed in simulations that for a dust floating potential of $\phi_{\mathrm{fl}} = -2.0\,\mathrm{V}$ and an ion drift speed of $M = 0.75$, the peak of the ion focus was located at $1.1\lambda_{\mathrm{D,e}}$. In our case, the floating potential was estimated to be $-7.242\,\mathrm{V}$, which is quite a bit higher. The electron Debye length is 407 µm for our plasma, so using this assumption, the first wakefield peak (with a positive potential) should be located at approximately 448 µm downstream from the first dust particle. If the first peak is found at 448 µm, then it is actually closer to the second downstream particle in the string than the first downstream particle, for the found typical interparticle distance of 250 µm. This does not support wakefield attraction as the main force responsible for the transverse alignment of dust particles, because then we would expect the first wakefield peak to coincide more closely with the position of the first downstream dust particle in the string. To what extent the floating potential affected the position of the peak was not studied by Ludwig *et al.*, so this wakefield position might not be accurate for our case. It is possible that the actual location of the first wakefield peak is closer to the first downstream particle in the string. The interparticle distance does, however, match closely with the effective Debye length for streaming ions at certain ion velocities. In our case, at $M = 0.75$, the effective Debye length is 276 µm.

An alternative explanation for the opposing force on the dust particles in the dilute region is that the electric field force in actuality comes from the combined effect of the cloud of dust particles just outside of the dilute region. In a simplified picture, if we follow a particle P_1 in the dilute region that approaches the dense region, it will experience an electrostatic interaction with the first particle in the dense region it encounters, P_2. P_1 will then experience

an acceleration opposite to its velocity and be slowed down, and P_2 will be accelerated further into the dense region. P_2 will then experience resistive forces by the other particles in the dense region that impede further movement into the dense region. P_1 will essentially be met with the inertia of more particles the closer to the dense region it gets. This way, the dust cloud could act like a large electrostatic 'cushion' for the incoming dust particles or strings from the dilute region.

These dynamics are easier to see when viewing the video of the image series where one can visually track single dust particles. It is apparent in the experiment videos that there are relatively slowly moving vortices in the dense regions of the dust cloud. Particles that start out in the dilute region, sometimes in strings, are accelerated by (most likely) the ion drag force, and eventually they enter and merge into the denser regions of the dust cloud. Meanwhile, some of the particles from the dense regions enter the dilute region due to the vortex rotation of the dust cloud. Such a cycle makes it so that there will always be particles present in the dense region at the end of the dilute region, and the particles in the dense dust cloud could potentially be among the main contributors in the electric field force term $F_{\mathrm{es,E}}$ to provide a force balance in the dilute region. The velocities of two dust particles in the beginning of the dense region were calculated in section 5.4 to have approximately half the speed of the dust particles in the dilute region. The velocity of the majority of the particles in the dense cloud do not decrease to zero due to the vortex movement of the dust cloud.

In addition, the de-acceleration of the dust particles and strings in the middle of the dilute region could also be in part because of a reduction in the longitudinal ion drag force due to deflection of ions in the ion stream, which is one of the main features in the ion drag model by Piel (2011) [66], in the beginning of the dilute region. If a significant fraction of the ions in the ion stream is deflected in various directions, it would decrease the incoming flux of ions on subsequent particles in the string and therefore reduce the longitudinal ion drag force on the downstream particles. Such a diffusion of the ion stream could explain why strings often appear less pronounced the further into the dust cloud they are.

6 Conclusions

In this thesis, the PSO algorithm has been applied as a new method for reconstructing the dust particle positions from the data images using stereoscopy. It has been optimized through a systematic benchmarking process using artificial particle sets and images, and then it has been compared to one of the current state-of-the-art algorithms for particle position reconstruction, the STB algorithm. Through testing on an artificial particle set with realistic-looking images, it was found that the PSO algorithm was able to accurately find a slightly higher number of the ground truth positions, but it also returned a somewhat higher fraction of ghost particles. With regards to runtime, STB is significantly faster than the PSO algorithm. If initial trajectories are obtained, the STB algorithm can process a set of 10 frames in less than 30 minutes. For the PSO algorithm, one frame can take 2-4 hours on a laptop computer. However, due to the necessity of having initial trajectories to run the STB algorithm, which needs to be obtained from a separate algorithm, when analyzing real images the STB algorithm will likely see a reduction in accuracy. The PSO algorithm does not have this problem since it does not require initial trajectories. Overall, it is demonstrated that PSO is a viable method for detection of dust particles with a stereoscopic setup when compared to other algorithms currently used for this purpose.

The PSO algorithm was then used to analyze real data from microgravity dusty plasma experiments performed on parabolic flights. Three-dimensional string formation of dust particles was observed in the dilute midplane region of the plasma chamber. The directions of the strings were mainly in the x-direction in the coordinate system of the plasma chamber. The y-components of the string directions had some variation between the different strings, but was on average zero. The strings were observed to have a slight slope of approximately 7.9° in the z-direction, which was contrary to expectations. The string directions were expected to be symmetric in the z-component as well. Suggested causes for these slopes include that the laser sheet was not illuminating the exact center of the void, or due to residual acceleration. The dust particles in the dilute region, including the dust strings, drifted outward along the x-axis before at some point reaching the denser regions of the dust cloud, where string formation was observed to be less prominent.

By calculating the pair correlation function $g(r)$ in the dilute region, where strings were observed, an interparticle distance of $d = (250 \pm 40)\,\mu\text{m}$ was found. The interparticle distance was found to be constant in time in the analyzed frame range, which was equivalent to a time of 2.5 seconds. The effective Debye length was calculated for the plasma when taking the ion drift speed into account for ion flow Mach numbers M between 0 and 2, and a Debye length of 276 µm was found for the case of $M = 0.75$. This coincides quite closely with the estimated interparticle distance in the strings. The ion drift speed is not known, though it is argued that it is likely to be subsonic for this experimental setup [66], and more specifically between $M = 0.5$ and $M = 1$.

Lastly, various forces acting on a dust particle in a string were calculated. The longitudinal forces that were considered were the longitudinal ion drag force, the neutral drag force, the ion wakefield force, the electric polarization force, the electrostatic force between dust particles and the electrostatic force due to other remaining electric fields (the 'electric field force'). The transverse force that was considered was the transverse ion drag force.

For a dust particle located in between two other dust particles in a string, the electrostatic force between particles naturally cancelled each other out at the midpoint between the dust particles when assuming that the dust charges were equal. At a displacement of $x = 0.1d = 25\,\mu\text{m}$ in the longitudinal direction, the net electrostatic force was in the order of 10^{-13} N toward

the equilibrium point $x = 0$. The transverse restoring ion drag force was also calculated to be in the order of 10^{-13} N for a displacement of $s = 0.1d$ in the transverse direction. The electrostatic repulsive force between particles in a string could be an important contributing force for establishing consistent interparticle spacing within particle strings, in addition to the transverse alignment caused by the transverse ion drag force. However, previous simulations have shown that dust particles in a string could have different charges due to decharging in the downstream ion focus regions [15], which would affect the electrostatic force between dust particles. For this reason, we cannot say with certainty to what extent the electrostatic repulsion brings stability to the interparticle distances in a string.

The neutral drag force is most likely not one of the main forces that contribute to the dynamics of the dust particles. It does provide a small force of $|F_{\mathrm{n}}| \sim 10^{-14}$ N that opposes the movement of dust particles, but it is not significant compared to the longitudinal ion drag force at ion flow speeds $M > 0.2$. The longitudinal ion drag force was calculated to have a magnitude of $|F_{\mathrm{i,long}}| \sim 10^{-12}$ N for ion flow Mach number $M = 0.75$. It is concluded that the ion drag force is the main driver of the longitudinal drift of the dust particles and strings in the dilute region.

The case of a dust particle in a string with constant velocity, or zero acceleration, was considered. Due to Newton's second law, the remaining forces, namely the ion wakefield force, the electric polarization force the electric field force, must together provide the resistive forces. Though the ion wakefield force could not directly be calculated, the calculated interparticle distance in a dust string was compared to an estimated location of the first peak of the ion wakefield, based on information found in the literature. The interparticle distance and the position of the first wakefield peak did not coincide in this case. It would be expected for the first downstream particle to coincide quite closely with the first positive peak of the wakefield if the wakefield force was responsible for the alignment of dust particles. However, the floating potential of the dust particles used in the relevant study was $\phi_{\mathrm{fl}} = -2.0$ V, and the estimated floating potential of the dust particles in this thesis was $\phi_{\mathrm{fl}} \approx -7.2$ V. For this reason, the estimated position of the first peak could be erroneous, and therefore the importance of the ion wakefield force for string formation remains inconclusive. It is suggested that the remaining forces that cause the de-acceleration of the dust particles in the dilute region could be a combination of the reduction of the longitudinal ion drag force due to upstream particles in the dilute region deflecting the ions in the ion stream, and the accumulated electrostatic resistive forces from the dust particles in the dense region of the dust cloud.

7 Outlook

In this section some future work is suggested, based on the work that was done in this thesis.

The Hough transform: A possibility for automatic detection of strings in three dimensions is to use the three-dimensional *Hough transform*. One of the simplest usages of the Hough transform is to detect lines in two-dimensional images, and a generalization for three dimensions exists as well [23].

Trajectory filter for PSO: After the found positions are retrieved from the PSO algorithm, using a particle trajectory filter on the set of found positions can help to eliminate ghost particles. This can be done for example using similar methods as used in the STB algorithm. The `residualLimit` parameter can then be set to a value slightly closer to 0, which will result in more candidate particles being accepted by PSO, including both more accurate particles and more ghost particles. By applying a trajectory filter, many of the extra ghost particles can be filtered away, thus keeping the high fraction of discovered particle positions but reducing the fraction of ghost particles. Note that the trajectory filtering algorithm should be able to handle 7000 or more found particles per frame, since it seems that at least this many particles are present in the plasma chamber, at least for experiments similar to the one analyzed in this thesis.

Full benchmarking of STB: When benchmarking the accuracy of the STB algorithm, I used the ground truth trajectories of the first 10 frames as initial trajectories for STB. A more realistic approach would be to obtain the initial tracks from running a separate algorithm, such as IPR, on the 10 first frames, as one would do for real images. This would give more realistic accuracy results for STB compared to when using perfect initial trajectories.

References

[1] AirZeroG - How parabolic flights work.

[2] Melamine-formaldehyde resins. https://polymerdatabase.com/polymer%20classes/MelamineFormaldehyde%20type.html. Accessed: 17-07-2021.

[3] NASA - *Our Sun*. https://solarsystem.nasa.gov/solar-system/sun/in-depth/. Accessed: 20-07-2021.

[4] Particle swarm optimization - MATLAB particleswarm. mathworks.com/help/gads/particleswarm.html. Accessed: 07-07-2021.

[5] Particle Swarm Optimization Algorithm. mathworks.com/help/gads/particle-swarm-optimization-algorithm.html. Accessed: 07-07-2021.

[6] Plasma Universe - *Plasma*. https://www.plasma-universe.com/plasma/. Accessed: 21-07-2021.

[7] Plasma Universe - *Quasi-neutrality*. https://www.plasma-universe.com/quasi-neutrality/. Accessed: 21-07-2021.

[8] Shotcut - Open source video editor.

[9] Techopedia - *Radio frequency*. https://www.techopedia.com/definition/5083/radio-frequency-rf. Accessed: 25-07-2021.

[10] Chemistry Libre Texts - *Photoelectron Spectroscopy: Theory*. https://chem.libretexts.org/@go/page/1837, 8 2020. Accessed: 21-07-2021.

[11] O. Arp, J. Goree, and A. Piel. Particle chains in a dilute dusty plasma with subsonic ion flow. *Phys. Rev. E*, 85:046409, Apr 2012.

[12] Oliver Arp, David Caliebe, Kristoffer Menzel, Alexander Piel, and John Goree. Experimental investigation of dust density waves and plasma glow. *Plasma Science, IEEE Transactions on*, 38:842 – 846, 05 2010.

[13] Michael S. Barnes, John H. Keller, John C. Forster, James A. O'Neill, and D. Keith Coultas. Transport of dust particles in glow-discharge plasmas. *Phys. Rev. Lett.*, 68:313–316, Jan 1992.

[14] P. M Bellan. *Fundamentals of Plasma Physics*. Wiley, 1 edition, 2006.

[15] D. Block and W. Miloch. Charging of multiple grains in subsonic and supersonic plasma flows. *Plasma Physics and Controlled Fusion*, 57:014019, 2015.

[16] Dietmar Block and Andre Melzer. Dusty (complex) plasmas, routes towards magnetized and polydisperse systems. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 52(6):063001, feb 2019.

[17] A. Bouchoule, J.-P. Bœuf, L. Boufendi, C. Hollenstein, J. Perrin, C. Punset, E. Stoffels, and W. Stoffels. *Dusty Plasmas: Physics, Chemistry and Technological Impacts in plasma Processing*. Wiley, 1 edition, 1999.

[18] Michael J. Braus. The theory of electrolytes. I. Freezing point depression and related phenomena (translation). 2019.

[19] L. Brieda. *Plasma Simulations by Example*. CRC Press, 1 edition, 2019.

[20] L. E. Brus. A simple model for the ionization potential, electron affinity, and aqueous redox potentials of small semiconductor crystallites. *The Journal of Chemical Physics*, 79(11):5566–5571, 1983.

[21] P. Chabert and N. Braithwaite. *Physics of Radio-Frequency Plasmas*. Cambridge University Press, 3 edition, 2011.

[22] F. F. Chen. *Introduction to Plasma Physics and Controlled Fusion*. Springer, 3 edition, 2015.

[23] Christoph Dalitz, Tilman Schramke, and Manuel Jeltsch. Iterative Hough Transform for Line Detection in 3D Point Clouds. *Image Processing On Line*, 7:184–196, 2017. `https://doi.org/10.5201/ipol.2017.208`.

[24] J. E. Daugherty, R. K. Porteous, and D. B. Graves. Electrostatic forces on small particles in low-pressure discharges. *Journal of Applied Physics*, 73(4):1617–1620, 1993.

[25] A. David and B. Miller. Optical physics of quantum wells. 2020.

[26] T. Deka, A. Boruah, S. K. Sharma, and H. Bailung. Observation of self-excited dust acoustic wave in dusty plasma with nanometer size dust grains. *Physics of Plasmas*, 24(9):093706, 2017.

[27] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 39–43, 1995.

[28] Paul S. Epstein. On the Resistance Experienced by Spheres in their Motion through Gases. *Phys. Rev.*, 23:710–733, Jun 1924.

[29] Sergey V. Gaponenko and Hilmi Volkan Demir. *Electrons in potential wells and in solids*, page 9–51. Cambridge University Press, 2018.

[30] W. Goedheer and V. Land. Simulation of dust voids in complex plasmas. *Plasma Physics and Controlled Fusion*, 50:124022, 12 2008.

[31] J. Goree, G. E. Morfill, V. N. Tsytovich, and S. V. Vladimirov. Theory of dust voids in plasmas. *Phys. Rev. E*, 59:7055–7067, Jun 1999.

[32] D. J. Griffiths. *Introduction to Electromagnetism*. Prentice Hall, 3 edition, 1999.

[33] S. Hamaguchi and R. T. Farouki. Polarization force on a charged particulate in a nonuniform plasma. *Phys. Rev. E*, 49:4430–4441, May 1994.

[34] W. M. Haynes. *Handbook of Chemistry and Physics*. CRC Press, 92 edition, 2011.

[35] M. Himpel, S. Schütt, W. J. Miloch, and A. Melzer. Layered structures in extended dust clouds under microgravity. In *Phys. Plasmas 25, 083707 (2018); https://doi.org/10.1063/1.5046049*, 2018.

[36] Michael Himpel. Single Particle Dynamics in Dusty Plasmas and Dust-Density Waves. 2014.

[37] Michael Himpel, Birger Buttenschön, and André Melzer. Three-view stereoscopy in dusty plasmas under microgravity: A calibration and reconstruction approach. *Review of Scientific Instruments*, 82(5):053706, 2011.

[38] Michael Himpel, Carsten Killer, Birger Buttenschön, and André Melzer. Three-dimensional single particle tracking in dense dust clouds by stereoscopy of fluorescent particles. *Physics of Plasmas*, 19, 12 2012.

[39] Michael Himpel and André Melzer. Three-dimensional reconstruction of individual particles in dense dust clouds: Benchmarking camera orientations and reconstruction algorithms. *Journal of Imaging*, 5, 2019.

[40] R. Hood, B. Scheiner, S. D. Baalrud, M. M. Hopkins, E. V. Barnat, B. T. Yee, R. L. Merlino, and F. Skiff. Ion flow and sheath structure near positively biased electrodes. *Physics of Plasmas*, 23(11):113503, 2016.

[41] Zuquan Hu, Yinhua Chen, Xiang Zheng, Feng Huang, Gei-fen Shi, and M. Y. Yu. Theory of void formation in dusty plasmas. *Physics of Plasmas*, 16(6):063707, 2009.

[42] I H Hutchinson. Collisionless ion drag force on a spherical grain. *Plasma Physics and Controlled Fusion*, 48(2):185–202, jan 2006.

[43] Yuriy Ivanov and André Melzer. Modes of three-dimensional dust crystals in dusty plasmas. *Phys. Rev. E*, 79:036402, Mar 2009.

[44] K. Jiang, Chengran Du, K. Sütterlin, A. Ivlev, and G. Morfill. Lane formation in binary complex plasmas: Role of non-additive interactions and initial configurations. *EPL (Europhysics Letters)*, 92:65002, 01 2011.

[45] Hendrik Jung, Franko Greiner, Oguz Han Asnaz, Jan Carstensen, and Alexander Piel. Exploring the wake of a dust particle by a continuously approaching test grain. *Physics of Plasmas*, 22(5):053702, 2015.

[46] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995.

[47] A.G. Khrapak, V. Molotkov, A. Lipaev, D. Zhukhovitskii, Vadim Naumkin, Valdimir Fortov, Oleg Petrov, Hubertus Thomas, Sergey Khrapak, Peter Huber, A. Ivlev, and Gregor Morfill. Complex Plasma Research under Microgravity Conditions: PK-3 Plus Laboratory on the International Space Station. *Contributions to Plasma Physics*, 56:253–262, 04 2016.

[48] M. Kroll, J. Schablinski, D. Block, and A. Piel. On the influence of wakefields on three-dimensional particle arrangements. *Physics of Plasmas*, 17(1):013702, 2010.

[49] Sebastian Käding and André Melzer. Three-dimensional stereoscopy of yukawa (coulomb) balls in dusty plasmas. *Physics of Plasmas*, 13(9):090701, 2006.

[50] Irving Langmuir. The interaction of electron and positive ion space charges in cathode sheaths. *Phys. Rev.*, 33:954–989, Jun 1929.

[51] G. Lapenta. Ion flow induced attractive force in complex plasma crystals. *Physica Scripta*, 64:599–604, 2001.

[52] Giovanni Lapenta. Nature of the force field in plasma wakes. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 66:026409, 09 2002.

[53] Victoria Emilie Lofstad. Stereoscopic imaging of dust density waves in microgravity. Master's thesis, University of Oslo, 2018.

[54] Patrick Ludwig, Wojciech J Miloch, Hanno Kählert, and Michael Bonitz. On the wake structure in streaming complex plasmas. *New Journal of Physics*, 14(5):053016, may 2012.

[55] R. G. Martin and M. Livio. The solar system as an exoplanetary system. *The Astrophysical Journal*, 09 2015.

[56] Manuel Martinez-Corral and Bahram Javidi. Fundamentals of 3d imaging and displays: A tutorial on integral imaging, light-field, and plenoptic systems. *Advances in Optics and Photonics*, 10:512, 09 2018.

[57] A. Melzer. *Physics of Dusty Plasmas, An Introduction*. Springer, 2019.

[58] K Menzel, O Arp, and A Piel. Spatial frequency clustering in nonlinear dust-density waves. *Physical review letters*, 104:235002, 06 2010.

[59] E. Million. The Hadamard Product. 2007.

[60] W. J. Miloch and D. Block. Dust grain charging in a wake of other grains. *Physics of Plasmas*, 19(12):123703, 2012.

[61] W. J. Miloch, M. Kroll, and D. Block. Charging and dynamics of a dust grain in the wake of another grain in flowing plasmas. *Physics of Plasmas*, 17(10):103703, 2010.

[62] Nobuki Mutsukura, Kenji Kobayashi, and Yoshio Machi. Plasma sheath thickness in radio-frequency discharges. *Journal of Applied Physics*, 68(6):2657–2660, 1990.

[63] Gonçalo Pereira. Particle Swarm Optimization. 05 2011.

[64] A. Piel. *Plasma Physics: An Introduction to Laboratory, Space and Fusion Plasmas*. Springer, 1 edition, 2010.

[65] A. Piel, M. Klindworth, O. Arp, A. Melzer, and M. Wolter. Obliquely Propagating Dust-Density Plasma Waves in the Presence of an Ion Beam. *Phys. Rev. Lett.*, 97:205009, Nov 2006.

[66] Alexander Piel. Alignment of dust particles by ion drag forces in subsonic flows. *Physics of Plasmas*, 18(7):073704, 2011.

[67] H. L. Pécseli. *Waves and Oscillations in Plasmas*. CRC Press, 1 edition, 2013.

[68] RA Quinn, C Cui, John Goree, J. Pieper, Hubertus Thomas, and G. Morfill. Structural analysis of a coulomb lattice in a dusty plasma. *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics*, 53:R2049–R2052, 04 1996.

[69] D Samsonov and John Goree. Instabilities in a dusty plasma with ion drag and ionization. *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics*, 59, 02 1999.

[70] Daniel Schanz, Andreas Schröder, Sebastian Gesemann, Dirk Michaelis, and Bernhard Wieneke. 'shake the box': A highly efficient and accurate tomographic particle tracking velocimetry (tomo-ptv) method using prediction of particle positions. pages 1–13, 01 2013.

[71] Daniel Schanz, Andreas Schröder, Sebastian Gesemann, and Bernhard Wieneke. 'shake the box': Lagrangian particle tracking in densely seeded flows at high spatial resolution. 01 2015.

[72] Alexander A. Schekochihin. Lectures on Kinetic Theory of Gases and Statistical Physics, November 2020.

[73] D. V. Schroeder. *An Introduction to Thermal Physics*. Pearson, 7 edition, 1999.

[74] Shaukat Ali Shan and Nadia Imtiaz. Shocks in an electronegative plasma with boltzmann negative ions and k-distributed trapped electrons. *Physics Letters A*, 383(18):2176–2184, 2019.

[75] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360)*, pages 69–73, 1998.

[76] P. K. Shukla and A. A. Mamun. *Introduction to Dusty Plasma Physics*. CRC Press, 1 edition, 2001.

[77] B. Smith, J. Vasut, T. Hyde, L. Matthews, J. Reay, M. Cook, and J. Schmoke. Dusty plasma correlation function experiment. *Advances in Space Research*, 34(11):2379–2383, 2004. Scientific Exploration, Planetary Protection, Active Experiments and Dusty Plasmas.

[78] R. J. M. M. Snijkers. *The sheath of an RF plasma : measurements and simulations of the ion energy distribution*. PhD thesis, Eindhoven University of Technology, 1993.

[79] N. Takata and T. Matsumoto. The effect of the applied voltage on ionization in a parallel plate ionization chamber. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 302(2):327–330, 1991.

[80] Hubertus Thomas, Mierk Schwabe, and M. Pustylnik. PK-4 - Complex plasma research on the International Space Station. 01 2020.

[81] Durga Tripathi and Umaranjan Jena. Cognitive and social information based pso. *International Journal of Engineering, Science and Technology*, 8:64, 11 2016.

[82] F. Veerhest. *Waves In Dusty Space Plasmas*. Kluwer Academic Publishers, 2000.

[83] H. Vestner and L. Waldmann. Generalized hydrodynamics of thermal transpiration, thermal force and friction force. *Physica A: Statistical Mechanics and its Applications*, 86(2):303–336, 1977.

[84] Sergey V. Vladimirov and Mitsuhiro Nambu. Attraction of charged particulates in plasmas with finite flows. *Phys. Rev. E*, 52:R2172–R2174, Sep 1995.

[85] Dongshu Wang, Dapei Tan, and L. Liu. Particle swarm optimization algorithm: an overview. *Soft Computing*, 22:387–408, 2018.

[86] F. White. *Fluid Mechanics*. McGraw-Hill, 7 edition, 2011.

[87] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22:1330 – 1334, 12 1999.

[88] X. H. Zheng and J. C. Earnshaw. Plasma-Dust Crystals and Brownian Motion. *Phys. Rev. Lett.*, 75:4214–4217, Dec 1995.

A Proof of number density correction factor

A proof for eq. (5.2) is presented here. Let V be the total volume of the space containing the particles. Let N_{GT} be the real (ground truth) total number of particles. Let N_{PSO} be the total number of positions returned by the PSO algorithm. Let N_{acc} be the number of accurate positions returned by the PSO algorithm, and let N_{ghost} be the number of ghost particles returned by the PSO algorithm. We have

$$N_{\mathrm{PSO}} = N_{\mathrm{acc}} + N_{\mathrm{ghost}}. \tag{A.1}$$

The number densities for the ground truth positions and the PSO-returned positions are defined as

$$n_{\mathrm{GT}} \equiv N_{\mathrm{GT}}/V, \quad \text{and} \tag{A.2}$$
$$n_{\mathrm{PSO}} \equiv N_{\mathrm{PSO}}/V, \tag{A.3}$$

respectively.

The ghost fraction is defined as $\texttt{ghostFraction} \equiv N_{\mathrm{ghost}}/N_{\mathrm{PSO}}$, and the found fraction is defined as $\texttt{foundFraction} \equiv N_{\mathrm{acc}}/N_{\mathrm{GT}}$. In addition, we can define the *accurate fraction* as $\texttt{accurateFraction} \equiv 1 - \texttt{ghostFraction}$. The accurate fraction is the fraction of accurate positions in the positions returned by PSO. We then have the following definitions:

$$\texttt{foundFraction} \equiv N_{\mathrm{acc}}/N_{\mathrm{GT}}, \tag{A.4}$$
$$\texttt{ghostFraction} \equiv N_{\mathrm{ghost}}/N_{\mathrm{PSO}}, \tag{A.5}$$
$$\texttt{accurateFraction} \equiv N_{\mathrm{acc}}/N_{\mathrm{PSO}}. \tag{A.6}$$

Rewrite the accuracy parameters as $\texttt{foundFraction} \equiv f$, $\texttt{ghostFraction} \equiv g$ and $\texttt{accurateFraction} \equiv a$.

Statement A.1.

$$n_{PSO} \cdot \frac{(1-g)}{f} = n_{GT}$$

Proof.

$$\begin{aligned} n_{\mathrm{PSO}} \cdot \frac{(1-g)}{f} &= \frac{N_{\mathrm{PSO}}}{V} \cdot \frac{a}{f} \\ &= \frac{N_{\mathrm{PSO}}}{V} \frac{N_{\mathrm{acc}}/N_{\mathrm{PSO}}}{N_{\mathrm{acc}}/N_{\mathrm{GT}}} \\ &= \frac{\cancel{N_{\mathrm{PSO}}}}{V} \frac{\cancel{N_{\mathrm{acc}}} N_{\mathrm{GT}}}{\cancel{N_{\mathrm{acc}}} \cancel{N_{\mathrm{PSO}}}} \\ &= \frac{N_{\mathrm{GT}}}{V} \\ &= n_{\mathrm{GT}} \end{aligned}$$

□

B Proof of ideal gas relationship for neutrals in the Epstein neutral drag formula

We have two expressions for the neutral drag force, from eqs. (2.29) and (2.30):

$$\mathbf{F}_{\mathrm{n1}} = -\delta\frac{4}{3}\pi a^2 m_{\mathrm{n}} v_{\mathrm{th,n}} n_{\mathrm{n}} \mathbf{v}_{\mathrm{d}}$$

and

$$\mathbf{F}_{\mathrm{n2}} = -m_{\mathrm{d}}\beta\mathbf{v}_{\mathrm{d}} = -m_{\mathrm{d}}\delta\frac{8}{\pi}\frac{p}{a\rho_{\mathrm{d}}v_{\mathrm{th,n}}}\mathbf{v}_{\mathrm{d}}.$$

We will assume that these two formulas give the same force, that is, that $\mathbf{F}_{\mathrm{n1}} = \mathbf{F}_{\mathrm{n2}}$.

The following expressions will be useful: The thermal velocity is $v_{\mathrm{tn,n}} = \sqrt{8k_{\mathrm{B}}T_s/(\pi m_s)}$ (from eq. (2.32)). The volume of a dust particle is $V_{\mathrm{d}} = (4/3)\pi a^3$. The mass of a dust particle is $m_{\mathrm{d}} = \rho_{\mathrm{d}}V_{\mathrm{d}}$.

Statement B.1. *If $\mathbf{F}_{n1} = \mathbf{F}_{n2}$, then the ideal gas law holds for the neutrals, $p = n_n k_B T_n$.*

Proof.

$$\begin{aligned}
\mathbf{F}_{\mathrm{n1}} &= \mathbf{F}_{\mathrm{n2}} \\
-\delta\frac{4}{3}\pi a^2 m_{\mathrm{n}} v_{\mathrm{th,n}} n_{\mathrm{n}} \mathbf{v}_{\mathrm{d}} &= -m_{\mathrm{d}}\delta\frac{8}{\pi}\frac{p}{a\rho_{\mathrm{d}}v_{\mathrm{th,n}}}\mathbf{v}_{\mathrm{d}} \\
&\Updownarrow \\
-\cancel{\delta}\frac{4}{3}\pi a^2 m_{\mathrm{n}} v_{\mathrm{th,n}} n_{\mathrm{n}} &= -m_{\mathrm{d}}\cancel{\delta}\frac{8}{\pi}\frac{p}{a\rho_{\mathrm{d}}v_{\mathrm{th,n}}} \\
\rho_{\mathrm{d}}\frac{4}{3}\pi a^3 m_{\mathrm{n}} n_{\mathrm{n}} &= m_{\mathrm{d}}\frac{8}{\pi}\frac{p}{v_{\mathrm{th,n}}^2} \\
\rho_{\mathrm{d}}V_{\mathrm{d}}\cancel{m_{\mathrm{n}}}n_{\mathrm{n}} &= m_{\mathrm{d}}\frac{\cancel{8}}{\cancel{\pi}}\frac{p\cancel{\pi}\cancel{m_{\mathrm{n}}}}{\cancel{8}k_{\mathrm{B}}T_{\mathrm{n}}} \\
\cancel{m_{\mathrm{d}}}n_{\mathrm{n}} &= \cancel{m_{\mathrm{d}}}\frac{p}{k_{\mathrm{B}}T_{\mathrm{n}}} \\
p &= n_{\mathrm{n}}k_{\mathrm{B}}T_{\mathrm{n}}
\end{aligned}$$

□