

Some assembly required

Computational simulations of dusty plasma

by

Isabel J. Rodriguez

An undergraduate honors thesis submitted in partial fulfillment of the
requirements for the degree of

Bachelors of Science

in

University Honors

and

Physics

Thesis Adviser

Erik Sánchez

Portland State University

2018

Author's to-do list

- * Add: Tables (experimental parameters and simulation parameters, Appendix)
- * Add: Figures (Appendix)
- * Add: Update references (Mendeley to .bib file)
- * Add: Links to git (code, footnote)
- * Review: equations (ALL, final draft)
- * Consider: would the sections make sense w/o the equations?
- * Formatting: consistency?
- * Update: Experimental results

Abstract

From astronomy to industry, dust's ubiquitous presence in plasma (so called "complex" or "dusty" plasma) makes it an interesting object of study for a number of different fields. In some cases, it plays a critical role in the progression of processes such as the formation of stars, planets, and complex molecules in interstellar clouds. In laboratory environments, it can play a more troublesome role—hindering, for instance, the efficiency of integrated circuits which are the foundation of our modern technological capabilities. Advances in computing power have enabled the utilization of simulations as a tool for exploring the transport of dust particles in these low-pressure radio-frequency discharges. In this work, results of a Particle-in-Cell (PIC) simulation used to investigate plasma dynamics in the presence of individual dust particles are presented.

For those in search of a gentle introduction to computational plasma physics.

To those without whom this thesis would not exist.

Mil gracias.

Acknowledgments

I would first like to thank my thesis advisor Erik Sánchez, who was willing to take me on with an overly ambitious independent project. With his guidance, and with the help of his graduate student Jeff Black, I learned how to adapt to the challenges of research and how to bring a project down to an appropriate level without selling the opportunity for learning and growth short.

I would also like to acknowledge Portland State LSAMP for providing the opportunity to present this research at the 2018 Pacific Northwest LSAMP Conference, and the Oregon NASA Space Grant Consortium for their support of my research endeavors this academic year.

Another individual I would like to acknowledge is Dr. Suzanne Estes, who has served as a long time mentor and second reader of this thesis. I am gratefully indebted to her both for her valuable edits as well as for supporting me during the research process.

Finally, I must express my very profound gratitude to my family, peers, advisers, and mentors for providing me with unfailing support and encouragement. It is because of all of you that this body of work exists in its current form.

Isa

Contents

Abstract	ii
Dedication	iii
Acknowledgments	iv
1 Background	1
2 Characteristics of dusty plasma	4
2.1 Primer: What is a plasma?	4
2.2 Conditions for a dusty (complex) plasma	7
2.3 Debye shielding in a dusty plasma	9
2.4 Forces on dust particles	9
2.5 Charging mechanisms	10
3 Plasma modeling	13
4 Motivation	16
4.1 Dusty plasma in space	16
4.2 Dusty plasma in fusion devices	17
4.3 Plasma sources in semiconductor processing	18
5 Description of the model	20

CONTENTS	vi
5.1 Governing Equations	21
5.2 Particle in Cell Method	22
5.2.1 Weighting (charge deposition)	23
5.2.2 Solving for Φ	24
5.2.3 Field solvers	25
5.2.4 Add/Move Particles	25
6 Simulation results	27
6.1 Simulation parameters	27
6.1.1 Boundary conditions	28
6.2 2D PIC Results	29
6.2.1 Test case: Single dust particle	29
6.2.2 Test case: Dust chain	31
6.3 Why open source?	31
6.4 Future work	34
A	
1D PIC Code in Python	37
A.1 Two-stream instability	37
B	
Sample Python code	39

List of Figures

1.1 Examples of naturally-occurring dusty plasma: a) on Earth: noctilucent clouds are high-altitude clouds consisting of charged ice particles; b) in space: the Cassini mission confirmed the presence of spokes in Saturn's rings caused by interactions of charged particles; c) human-made: rocket exhaust. (Image credit: nasa.gov)	2
2.1 A quasineutral plasma contains roughly equal amounts of ions, or positively charged particles, and electrons, or negatively charged particles. From some distance away, the charges cancel each other out and is considered to be neutral. In this plasma we can pick out a single charge, Q , and use it as a way to understand how this plasma behaves.	4
2.2 The solid line illustrates the drop-off of the electric potential of a single isolated charge over a given distance. The drop-off behavior of our test charge embedded in an ambient plasma with an electron temperature $T_e = 10^3$ eV and electron density $n_e = 10^{19}$ is shown in the dashed line. The Debye length is marked on the x-axis.	6
2.3 Components of a dusty plasma. Left to right: electron (e), ion (i), neutral atom (n), and dust particle (d).	8
2.4 Debye shielding of dust particles. A positively charged sheath with length λ_D results from the attraction of ions to the negatively charged dust particles that are separated by a distance a . This sheath shields the particles from 'seeing' the electric field generated by neighboring particles.	9

2.5 Left: Absorption of electrons incident to the dust particle's surface cause the dust to obtain a net negative charge. Right: An example of secondary emission as a result of highly energetic photons striking the dust particle (photoemission).	11
3.1 Fluid models tend to dominate fields such as astrophysics and astronomy. Pictured: Simulation of black hole accretion flow (REF).	13
4.1 Dusty plasma in (a) Interstellar environments. Pictured: Carina nebula, a dust pillar; (b) Saturn's ring spokes as observed by the Cassini mission. (Image credits: NASA).	16
4.2 Dusty particles in thermonuclear fusion devices (REF).	18
4.3 Dusty plasma in semiconductor processing devices (REF).	19
5.1 A flow chart laying out the Particle-in-Cell algorithm.	22
5.2 Area weighting used to deposit charge.	23
5.3 An illustration of the leap-frog method. We time-center the force while advancing the particle's velocity, and likewise time-center the velocity while advancing a particle's position.	25
6.1 How a dust particle is represented using a rectangular domain. Ions that pass from cells indexed with a 0 into cells indexed with a 1 are removed from the simulation (absorbing boundary conditions). Each cell is $1 \lambda_D$ in length.	29
6.2 Contour plots of the density profile resulting from the flow of heavy ions in a solar wind around an individual dust particle. Simulation parameters can be found in Section 6.1. A time-lapse video of the ion density profile can be found at https://youtu.be/SpYI5T0MIPQ	30
6.3 Contour plots of the density and plasma potential profiles resulting from the flow of argon plasma around an individual dust particle. Simulation parameters can be found in Section 6.1. A time-lapse video of the ion density profile can be found at https://youtu.be/WSDmXrQoUAU	30

6.4	Contour plots of the density (LEFT) and plasma potential (RIGHT) profiles resulting from the flow of argon plasma around a dust chain. Dust particles are spaced $4\lambda_{De}$ apart. Simulation parameters can be found in Section 6.1. A time-lapse video of the ion density profile can be found at https://youtu.be/VqtAZshY5xQ	31
6.5	Line plots of ion density (LEFT), plasma potential (MIDDLE), and electric field in the x-direction (RIGHT) taken at $ts = 50$, $ts = 100$, and $ts = 500$	32
6.6	Contour plots of the density (LEFT) and plasma potential (RIGHT) profiles resulting from the flow of argon plasma around a dust chain. Dust particles are spaced $4\lambda_{De}$ apart. Simulation parameters can be found in Section 6.1. A time-lapse video of the ion density profile can be found at https://youtu.be/O5QVE-SgT10	32
6.7	Line plots of ion density (LEFT), plasma potential (MIDDLE), and electric field in the x-direction (RIGHT) taken at $ts = 50$, $ts = 100$, and $ts = 500$	33
A.1	Top: Two opposing electron streams moving in a background of ions. An instability will develop due to charge bunching as the streams move through each other. Bottom: Evolution of the electron-electron two stream instability in phase space at times a) $t=0$, b) $t = 49$, c) $= 99$, and d) $t = 199$. The initial velocities are ± 0.2 , the grid has 1000 cells, there are 20000 particles per beam, and the beams have an initial sinusoidal perturbation of mode 2.	37

Chapter 1

Background

Everything
Turns,
Rotates,
Spins...
Pulsates,
Resonates,
And
Repeats.

Suzy Kassem

As much as 99% of the matter in the observable universe is in a plasma state—an ionized or partially-ionized gas containing electrons, ions, and neutral atoms—and much of it is laden with dust particulates with sizes ranging from submicrons to millimeters. This additional component in an otherwise typical plasma increases the behavioral complexity of the system, and is thus referred to as a “complex” or a “dusty” plasma.

Naturally-occurring dusty plasma can be found in the interstellar medium, where it plays an important role in the formation of molecular hydrogen (Garscadden et al. 1994); it is embedded in protoplanetary disks, planetary rings, and in the tails of comets



Figure 1.1: Examples of naturally-occurring dusty plasma: a) on Earth: noctilucent clouds are high-altitude clouds consisting of charged ice particles; b) in space: the Cassini mission confirmed the presence of spokes in Saturn's rings caused by interactions of charged particles; c) human-made: rocket exhaust. (Image credit: nasa.gov)

(REF), and can even be observed as noctilucent clouds in the Earth's mesosphere (REF).

In terrestrial environments, dusty plasma can be found everywhere from rocket exhaust to fusion devices, microelectronic fabrication and, of course, in the laboratory . In industry, it can be regarded as technologically valuable for its applications in the medical field, where it can be used to treat wounds (Lacci et al., 2010); in microelectronics, where plasma processing has been responsible for innovation and growth (REF); and in aerospace, where ion propulsion is being pursued as a cheaper, faster, and more efficient alternative to traditional chemical rockets (REF).

The computational exploration of physical systems is a unique niche that helps bridge between the theoretical and experimental worlds. Analytic techniques are powerful, but on their own they can only fully solve a small subset of problems —usually special cases. On the other hand, some experiments, as well as certain types of measurements, are impractical due to lack of resources or (as in the case of fields such as astronomy) are simply impossible to carry out. Numerical techniques help pick up the slack where analytics falls short, and their results inform our knowledge and suggest new experimental directions to take.

In the remainder of this work, we will look at some of the basic theory behind plasma and what makes a dusty plasma, explore current modeling techniques, note some fields of interest, and explore how an electrostatic particle-in-cell algorithm was used to simulate the behavior of a collisionless plasma in the presence of dust.

Chapter 2

Characteristics of dusty plasma

Physics depends on a universe
infinitely centered on an equals sign.

Mark Z. Danielenski

2.1 Primer: What is a plasma?

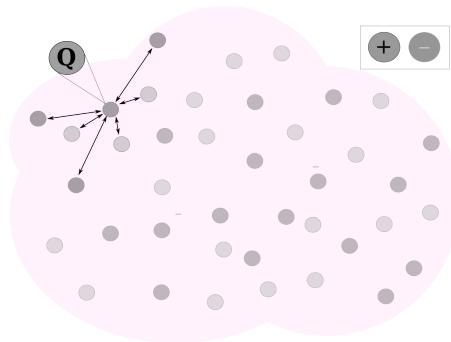


Figure 2.1: A quasineutral plasma contains roughly equal amounts of ions, or positively charged particles, and electrons, or negatively charged particles. From some distance away, the charges cancel each other out and is considered to be neutral. In this plasma we can pick out a single charge, Q , and use it as a way to understand how this plasma behaves.

Here on Earth, we are intimately familiar with three states of matter: solids, liquids,

and gases. Our lived experience with water, for example, has endowed us with the understanding that adding energy (heat) to ice melts it into water, and that with enough heat that water will boil and evaporate into steam. Keep adding energy into that cloud of steam and at some point you will start to strip the atoms in that cloud of their electrons—a process called ionization—and create plasma. The consequences resulting from this process are what really sets the plasma state apart from the three with which we are familiar: its ability to produce electromagnetic (EM) radiation.

In its most basic definition, the plasma state is one in which a gas has become partially or fully ionized. To make this definition a bit more rigorous, there are three conditions that need to be satisfied, but before we can go into what they are, we need to first define a few important time and length scales. To do that, we will consider a very simple plasma containing an equal number of ions and electrons.

If we pick out just one of the charges in this system, let's call Q_T our test particle [FIG], and look at its electric potential ¹, we get the graph shown in [FIG]. This differs from a potential curve that would result if Q had been stranded alone in space, and is shown as the dashed curve in [FIG]. That's the Coulomb potential which drops off as one over the distance. The space between these plots is due to the polarization of the plasma that occurs as a result of the attraction of electrons to that positive test charge.

Recasting this in a mathematical form, we can treat the potential of an isolated charge as a sphere of charge whose potential is given by,

$$\phi = \frac{Q_T}{4\pi\epsilon_0 r}, \quad (2.1)$$

where q is the charge, r is the distance, and ϵ_0 is the vacuum permittivity.

Compare this to the potential of our Q embedded in a plasma,

$$\phi = \frac{Q_T e^{-r/\lambda_D}}{4\pi\epsilon_0 r}, \quad (2.2)$$

¹Recall that electric potential is the work needed to be done in order to move that charge from one point to another.

where λ_{De} is a quantity called the Debye length.

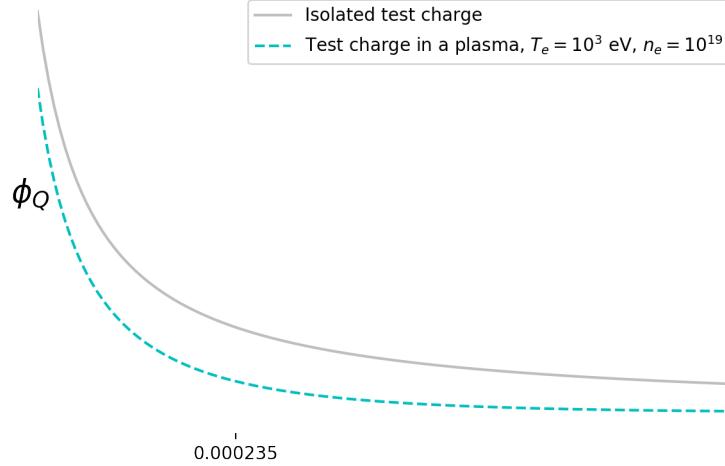


Figure 2.2: The solid line illustrates the drop-off of the electric potential of a single isolated charge over a given distance. The drop-off behavior of our test charge embedded in an ambient plasma with an electron temperature $T_e = 10^3$ eV and electron density $n_e = 10^{19}$ is shown in the dashed line. The Debye length is marked on the x-axis.

As can be observed in both [EQ 2.2] and [FIG], at points larger than λ_D the potential approaches zero. It is at this point that the test charge has been “screened” or “shielded” from other charges in the plasma, and with that we have introduced one of our most important characteristic length scales.

In a quasineutral plasma in which the ions are treated as immobile compared to the electrons, the electron Debye length is defined as,

$$\lambda_{De} = \sqrt{\frac{\epsilon_0 T_e}{n_e q_e}}. \quad (2.3)$$

Plasmas can oscillate with a frequency given by,

$$\omega_p = \sqrt{\frac{4\pi n_s Z^2 e^2}{m_s}}, \quad (2.4)$$

where s is the charge species (electron and ion), and Z is the charge number.

The Debye length is related to this frequency by,

$$\lambda_D = \frac{v_{ts}}{\omega_p}, \quad (2.5)$$

were v_{ts} is the typical velocity of a given species as determined by its Maxwell-Boltzmann distribution.

Now that we have our time and length scales, we can discuss the three criteria for a plasma:

- QUASINEUTRALITY ($\lambda_d \ll L$)
- COLLECTIVE EFFECTS DOMINATE ($N\lambda_D^3 \gg 1$)
- NEUTRAL COLLISIONS ARE NEGIGIBLE ($\omega\tau \gg 1$)

[DISCUSS HERE.]

There are two other characteristic parameters left to mention. One is the skin depth, or the depth at which plasma radiation can penetrate,

$$\lambda_{skin} = \frac{c}{\omega_p}, \quad (2.6)$$

where c is the speed of light.

The final parameter, the Larmour frequency, is necessary to in order to describe magnetic plasma however we will not consider it here. A more comprehensive look at plasma physics can be found in the work of Chen (1964).

2.2 Conditions for a dusty (complex) plasma

The loose definition of dusty plasma is the presence of large ² particles in an otherwise typical plasma (FIG) (REF). However, it is important to distinguish between having

²Remember how we said that in an electron-ion plasma, the ions were massive and immobile compared to the electrons? Well compared to the ions, these dust particles are billions of times more massive and thus can be considered immobile relative to the ions *and* the electrons. This will be important when we talk about charging of a dust particles.

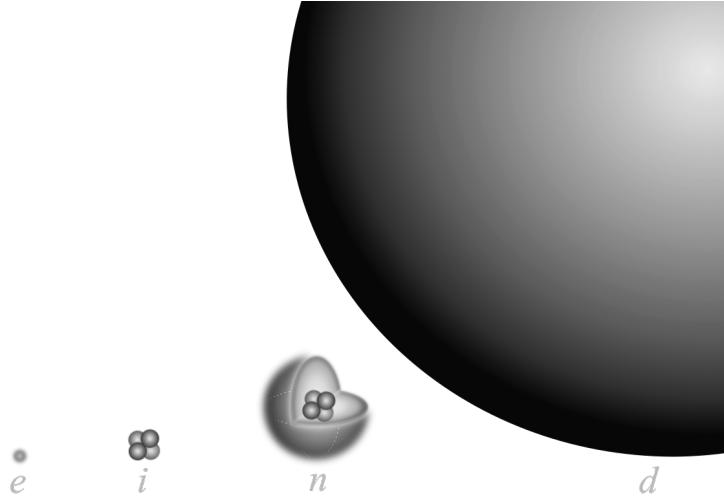


Figure 2.3: Components of a dusty plasma. Left to right: electron (e), ion (i), neutral atom (n), and dust particle (d).

dust *in* plasma and having *dusty plasma*. This distinction depends on the ordering of three characteristic length scales specific to this type of plasma: the dust radius r_d , the distance between dust particles a , and the Debye length λ_D (Shukla et al., 2002),

$$r_d \ll \lambda_D < a, \quad (2.7)$$

$$r_d \ll a < \lambda_D. \quad (2.8)$$

Put another way, if the distance between dust particles is smaller than the Debye length [EQ 2.8], the dust participates in the collective behavior of the system; conversely, if the interparticle distance is larger than the Debye length [EQ 2.7], you can regard it as having a collection of isolated screened grains, as was the case with our Q in the previous section. Furthermore, if the distance between dust particles is much smaller than the Debye length ($a \ll \lambda_D$), then we can treat the dust grains as if they were massive charged particles (Shukla et al., 2002).

2.3 Debye shielding in a dusty plasma

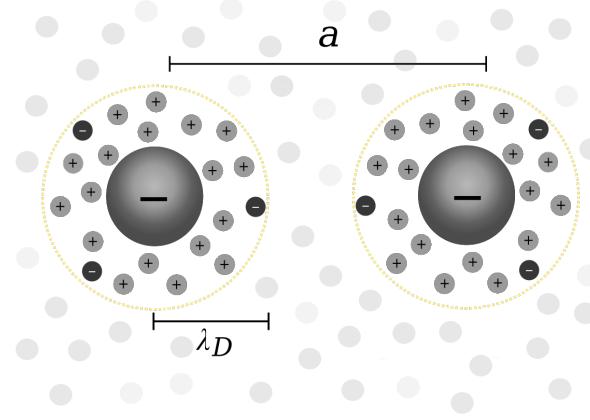


Figure 2.4: Debye shielding of dust particles. A positively charged sheath with length λ_D results from the attraction of ions to the negatively charged dust particles that are separated by a distance a . This sheath shields the particles from ‘seeing’ the electric field generated by neighboring particles.

Figure 2 illustrates the concept of Debye shielding in a dusty plasma. If we regard a dust particle as a sphere of charge, we expect that it would attract particles of opposite charge (i.e., negative electrons if the dust has a net positive charge, or ions if the dust has net negative charge). This attraction creates a cloud surrounding the dust particle, called a sheath, which shields the electric field of our dust particle from the rest of the plasma. In a dusty plasma, however, dust particles are not perfectly shielded due to the velocity distribution of ions and electrons in the sheath.

Another important consequence is the interplay of charges that allows particles to cluster and nucleate, creating larger grains.

2.4 Forces on dust particles

The basic governing equation describing the dynamics of a charged grain of dust with a mass m_d and velocity v_d is given to us by Newton’s second law,

$$M_d \frac{dv_d}{dt} = \sum F = \vec{F}_E + \vec{F}_G + \vec{F}_T + \vec{F}_D + \vec{F}_P, \quad (2.9)$$

where \vec{F}_E is the electromagnetic force, F_G is the gravitational force, \vec{F}_T is the thermophoretic force (associated with a temperature gradient in the plasma), \vec{F}_D is the ion drag force, and \vec{F}_P is the radiation pressure force. The electromagnetic force \vec{F}_E is a combination of the Coulomb force and the Lorentz force, where \vec{E} is the associated electric field, and \vec{B} is the associated magnetic field.

$$\vec{F}_E = \vec{F}_C + \vec{F}_L = q_d (\vec{E} + \vec{v} \times \vec{B}). \quad (2.10)$$

For an electrostatic plasma, we would leave out the Lorentz force term.

2.5 Charging mechanisms

There are three basic processes by which dust particles immersed in an ambient plasma become charged:

- Interactions between dust and neutral particles
- Interactions between dust and energetic charged particles
- Interactions between dust and energetic light particles

Don't be fooled, however! These elementary processes are actually quite complex and are difficult to understand, especially when trying to consider the different processes at one time as well as when looking at collections of dust particles. We will instead focus on the case of an isolated dust particle of finite size.

A dust particle placed in a plasma acts as a probe that will collect a primary species in that plasma (REF). Absorption of the ions present will cause a dust particle to become positively charged. However, in laboratory plasmas it is often the case for dust particles to acquire a negative surface charge. This is because the thermal velocity of electrons is much greater than that of ions, thus the electrons will tend to reach the dust first.

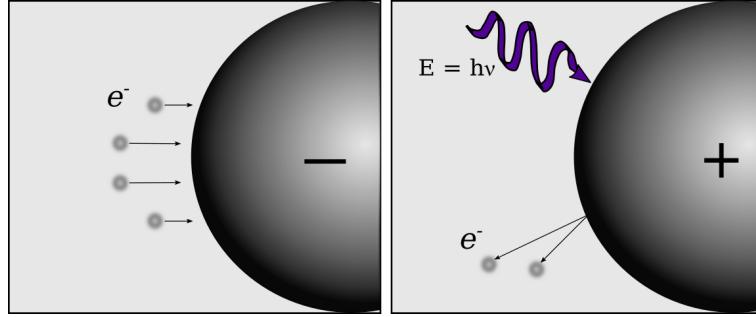


Figure 2.5: Left: Absorption of electrons incident to the dust particle's surface cause the dust to obtain a net negative charge. Right: An example of secondary emission as a result of highly energetic photons striking the dust particle (photoemission).

The charge of a dust grain is described by the rate at which it absorbs or collects charged particles,

$$\frac{dq_d}{dt} = \sum_s I_s(q), \quad (2.11)$$

where I is the current and s is the particle species. For an electron-ion plasma, [EQ 2.11] becomes,

$$\frac{dq_d}{dt} = I_e + I_i. \quad (2.12)$$

At equilibrium, no additional charge is being collected and thus

$$\frac{dq_d}{dt} = I_e + I_i = 0. \quad (2.13)$$

With no net current flow, the dust particle is left with a surface potential (which again, tends to be negative for laboratory dusty plasmas). A [TABLE] can be found listing the typical surface potentials for the most commonly used plasma sources.

Lets take another look at the current.

$$I_s = \sum_s \int q_s f_s \sigma_s(v, q) v d\vec{v}, \quad (2.14)$$

Here $v \equiv |\vec{v}|$ is the absolute value of the speed of the particles, and σ_s is the charge-collection cross-section ³ (Vladimirov, 1997) given as

$$\sigma_s = \pi a^2 \left(1 - \frac{2q_s q_d}{am_s v^2}\right) \quad \text{if} \quad \frac{2q_s q}{am_s v^2} < 1, \quad (2.15)$$

and

$$\sigma_s = 0 \quad \text{if} \quad \frac{2q_s q_d}{am_s v^2} \geq 1. \quad (2.16)$$

A negatively charged dust particle can become positively charged through secondary electron emissions resulting from surface impacts with energetic electrons and ions, or through the process of photoemission, in which an energetic ultraviolet photon (found both in space and industry environments) incident on the dust's surface results in the emission of electrons [FIG, RIGHT].

³A cross-section is an area over which there is a probability for a process to occur (Here, it's the probability of a dust particle absorbing a charge on its surface.)

Chapter 3

Plasma modeling

There is no need to ask the question,
Is the model true?...The only question
of interest is *Is the model illuminating
and useful?*

George Box

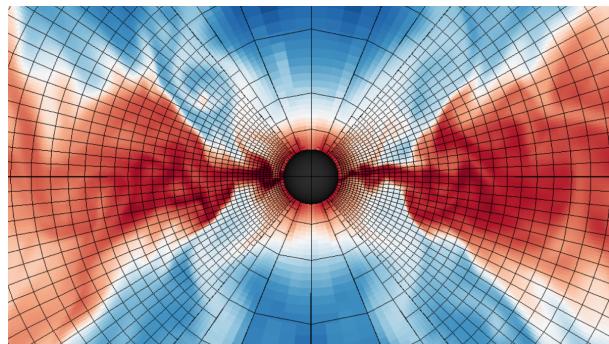


Figure 3.1: Fluid models tend to dominate fields such as astrophysics and astronomy. Pictured: Simulation of black hole accretion flow (REF).

Plasmas are dynamic, nonlinear, and can often be unstable, which is another way of saying they are too complicated to understand using analytic techniques alone. Numerical modeling serves as an extension of theory and is critical for informing experimental

diagnosis and experimental design (Bell, 1998).

To be able to *fully* describe a problem in three dimensions, you actually need six: three in position space, and three in velocity space. This, as you might imagine, can be problematic in spite of our advances in computing power, especially for small research groups without access to resources such as supercomputing clusters. Therefore, the goal is to find ways to get at the *essence* of plasma. The challenge is in the details —or rather, in determining which details to leave out so in order to optimize computational cost and the integrity of the information you're after. Surprisingly, you can leave out a lot! Four dimensions, for instance, as in the case of 1D1V plasma simulations (1 dimension and 1 velocity component). We can do this because plasma is a collective phenomena. Going back to the condition we laid out in Section 2.1, we are interested in plasma systems that are much longer than the Debye length ($L \ll \lambda_D$).

For a plasma that includes magnetic fields, and for whose particles you can ignore quantum effects, we can fully describe it using Maxwell's equations ¹:

$$\nabla \cdot \vec{E} = -\frac{\rho}{\epsilon_0}, \quad (3.1)$$

$$\nabla \times \vec{E} = -\frac{\partial B}{\partial t}, \quad (3.2)$$

$$\nabla \cdot \vec{B} = 0, \quad (3.3)$$

$$\nabla \times \vec{B} = \mu_0 \left(J + \epsilon_0 \frac{\partial \vec{E}}{\partial t} \right). \quad (3.4)$$

Plasma simulations can be split into two main categories: fluid and kinetic [FIG 3.1]. A kinetic description is the more physically realistic of the two, as it focuses on the effects of motion of charged particles within a plasma (Callen, 2003). It is this

¹For more on electrodynamics, I refer you to Griffiths (YEAR).

type of model that dominates the literature [REFS]. Particle-in-cell (PIC) and cloud-in-cell (CIC) simulation methods are some of the more common examples of kinetic simulations. At the most basic level, PIC and CIC rely on a field solver and particle mover to add up behaviors and effects on an individual-particle level. While the physics here is more accurate, there are computational challenges to be considered: to simulate the trajectories billions of particles ² in a fully three dimensional problem, or to take into account interparticle interactions.

In contrast, fluid simulations reduce the computational complexity by focusing on macroscopic behaviors (e.g. density) of charged particles. This dispenses with phase space information, collapsing a 6d problem into a 3d velocity space. The payoff is the to simulate large systems for long periods of time (Bell, 1998).

In both the kinetic and fluid cases, the trade offs can be mitigated through the use of a hybrid approach. In a hybrid model, certain species (hot, thermal electrons for instance) would be modeled as a fluid, and others (the cooler, slower, and more massive ion, neutral species, and dust species) are given the kinetic treatment.

²That's 2^N bits of computer memory, where N can be upwards of 10^{12} !

Chapter 4

Motivation

Fascinating.

Mr. Spock

4.1 Dusty plasma in space

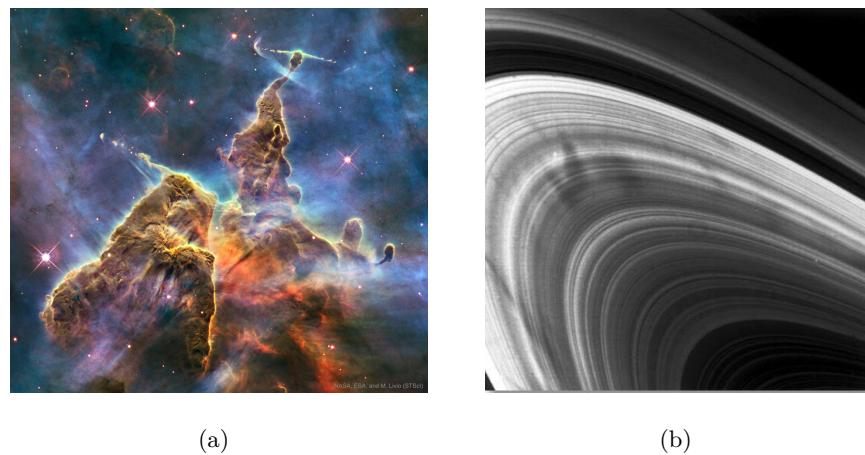


Figure 4.1: Dusty plasma in (a) Interstellar environments. Pictured: Carina nebula, a dust pillar; (b) Saturn's ring spokes as observed by the Cassini mission. (Image credits: NASA).

Interest in plasma as an astrophysical phenomena dates back to the 1980's, at a around the time most of the electromagnetic (EM) spectrum was being discovered (Per-

att, 1998). ¹ Examples in our own backyard include dusty cometary coma and Saturn's rings, both of which consist of tiny grains of ice and rock particles charged by the solar wind. The 1980 flyby of Saturn by Voyager 2 was the first time dynamic behavior was observed in Saturn's rings. Termed 'spokes', it's believed that these seasonal features are the result not of gravitational influences, but of the interplay between the charged particles and their electric fields (REF). The Voyager observations were followed up by the Cassini mission in 2005, and more than 10 years on are still considered an intriguing puzzle.

Venturing to the space between stars, the interstellar medium, we find that nebula (clouds of dust and ionized gases) are also full of complex organic molecules which may hold clues to the origin of life (REF). Collisional or collisionless shock waves ² in these dust-molecular clouds are thought to be responsible for creating the density condensation necessary to accelerate gravitational collapse and form stars (Popel et al., 2001). Dust charging may also be important factors for such things as shocks in supernovae explosions and particle acceleration (Popel and Tsytovich, 1999).

4.2 Dusty plasma in fusion devices

Fusion devices, whose success as a viable alternative source of renewable energy seems perpetually a decade or two on the horizon, harness the power of highly energetic plasma to overcome the Coulomb barrier. In the case of controlled thermonuclear fusion, tokamak reactors generate plasmas with energies upwards of 1000 eV (REF) – equivalent to 10 million degrees Celsius. These hot dense plasmas are confined using magnetic fields

¹Early models of our universe supposed that all mass in the universe was like the type of mass found on our planet (Peratt, 1998). With the help of technology, we've since been able to extend our observational senses into other parts of the EM spectrum (x-rays, gamma rays, etc.) and have since come to the understanding that most of the observable matter in our universe is in the plasma state.

²Resulting from interparticle friction and wave-particle interactions, respectively.

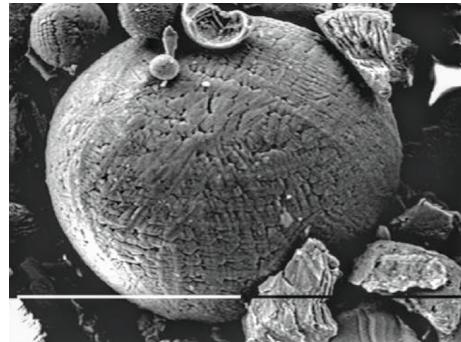


Figure 4.2: Dusty particles in thermonuclear fusion devices (REF).

³ (so called ‘magnetic confinement’), but confinement systems are not perfect and this can lead to damage to the damage of chamber walls and in-vessel ancillaries (REF). This not only has the potential to compromise structural integrity but the debris enter the plasma at a rate of several kilograms per day in a device such as the International Thermonuclear Experimental Reactor (ITER) (Thomas Jr., 2015), and is evidenced by the particulates observed at the bottom of fusion devices post-operation (Winter, 1998) [FIG]. These particles can not only change the transport properties of a plasma, they also migrate and can wind up compromising components that require active cooling, or block necessary gaps within the chamber (Winter, 1998).

4.3 Plasma sources in semiconductor processing

Research of dusty plasmas in industrial applications took off during the 1990’s and was driven by the formation of particles in reactors (Thomas Jr., 2015) which used radio frequency (RF) plasmas for semiconductor and thin-film processes. Capacitively Coupled Plasmas are the most frequently used in the development of the semiconductor wafers that are found in all of our electronic devices. Plasma etching is of particular interest for very-large-scale integration (VLSI), especially as we attempt to scale down electronics from the micro to the nano scale (Seo and Kim 2002). As we continue to downsize,

³Magnetic fields are only one of a number of confinement strategies.

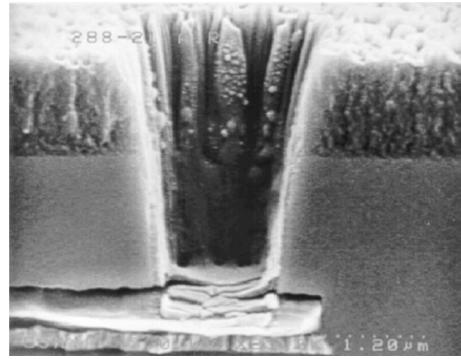


Figure 4.3: Dusty plasma in semiconductor processing devices (REF).

challenges resulting from increased resistance and capacitance in integrated circuits (IC) have been met with the introduction of low-dielectric (low-k) materials (REFS). The fluorocarbon plasmas used to etch these materials leave behind a fluorocarbon polymer (CF_x) residue which is subsequently cleaned using O₂ and H₂ plasmas. Plasma-induced damage during this cleaning process changes the dielectric properties of the material (REFS), posing further challenges to IC manufacturing. Plasma modeling can play a role in the innovation of efficient and effective solutions by informing the most optimal combination of plasma characteristics such that interactions with the CF_x residue minimizes damage to the low-k material (REFS).

Chapter 5

Description of the model

There is no problem more difficult to solve than that created by ourselves.

Felix Alba-Juez

A well-designed computational model can provide a wealth of information that is otherwise difficult to determine from experiments or observation alone. The Particle-In-Cell (PIC) technique, a powerful and widely used tool, simulates the motion of the charged particles in a plasma.

PIC simulations range in sophistication depending on the initial assumptions made when appraising a problem. On the simplest end of the spectrum, there are 1D electrostatic plasma models that are comprised of electrons and ions that are either stationary or moving in such a way that we can ignore collisions. We can continue to add dimensions of motion, introduce magnetic fields, allow charges to move at appreciable rates, allow charges to interact with one another, or add massive objects such as dust particles into the mix. Each of these features adds to the complexity of the algorithm, and increases computational strain.

In this work, we are working with looking at two cases of electrostatic collisionless plasma: low-temperature argon plasma, and heavy-ion solar wind. The dust particles used will come with a preset potential, implying that the particle has already undergone

charging.

Relative to the massive ions, dust and neutral atoms, we can, as is often done in PIC code, consider the electrons to be a fluid ¹ (Maiorov et al., 2000) (Other REFS). Furthermore, as we are working with a low density plasma, we can treat massive particles discrete particles (REFS). For simplicity, the dust grain will remain stationary and we will ignore particle interactions with neutral atoms.

With these validating assumptions in hand, we can make the choice to use a hybrid kinetic-fluid electrostatic model in which the Boltzmann relation can be applied to electron species. The latter choice will help to reduce computational time, as we will not have to follow the trajectories of every individual ion *and* electron— just the ions.

In the sections that follow, we will outline the equations that need to be solved in order to simulate our equation, and discuss how the PIC algorithm does this.

5.1 Governing Equations

For an electrostatic plasma we need only concern ourselves with the electric potential and electric fields. These are given to us through [EQ 2.1], where the electric field can be expressed as,

$$\vec{E} = -\nabla\phi. \quad (5.1)$$

This turns [EQ 3.1] into Poisson's equation,

$$\nabla^2\phi = -\frac{\rho}{\epsilon_0} \quad (5.2)$$

We can then pair these equations with a recasting of Newton's second law of motion to move the particles.

¹This is due to the fact that, given their low mass, electrons have a much higher velocity than the other species.

$$\frac{d\vec{v}}{dt} = \frac{q}{m} \vec{E}, \quad (5.3)$$

$$\frac{d\vec{x}}{dt} = \vec{v}. \quad (5.4)$$

5.2 Particle in Cell Method

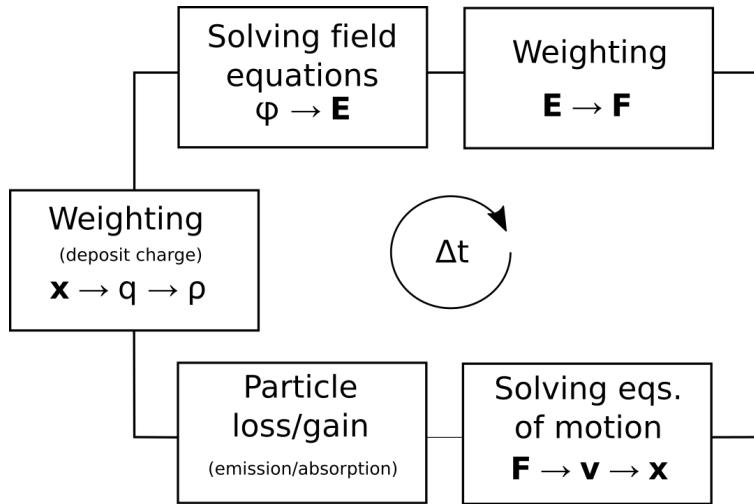


Figure 5.1: A flow chart laying out the Particle-in-Cell algorithm.

The standard outline of a PIC code [FIG] proceeds as follows: deposit the particles across the grid, determine the charge density, solve for the potential and electric fields, and use the electric fields to determine the force that will then cause the particles to accelerate, update particle position and velocities:

$$(1) \vec{x}_{particle} \rightarrow (2) \rho_{grid} \rightarrow (3) \phi_{grid} \rightarrow (4) \vec{E}_{grid} \rightarrow (5) \vec{F}_{particle}.$$

This is looped until a steady state is reached.

Steps (1) to (2) and (4) to (5) are achieved using a weighting scheme, while the others require a recasting of the equations to a finite-difference form. The use of a spatial grid in 2-dimensions leads to finite-sized square particles, and while they are

fairly symmetric, this can lead to some unwanted effects due to the fact that the forces on the particles will depend on their position within the cell in addition to the distance to other particles (REF).

A more comprehensive look at plasma simulations, including how to scale up to 3D, can be found in the seminal work Birdsall and Langdon (2005). Let's take a closer look at the individual steps.

5.2.1 Weighting (charge deposition)

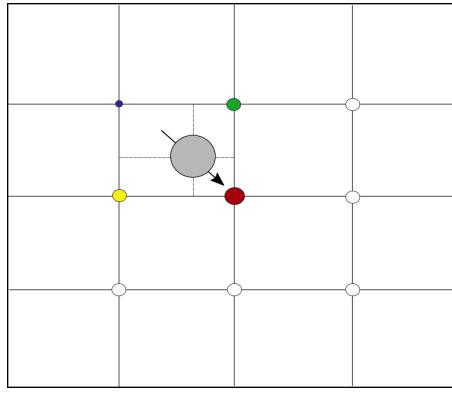


Figure 5.2: Area weighting used to deposit charge.

Once we've established our spatial domain, the first thing we need to do is distribute our charged particles, a process called weighting. There are several ways to go about this, but one of the common schemes is to use a first order weighting scheme [FIG], also called area weighting or linear interpolation. This is given by the following,

$$w_1 = \frac{(\Delta x - x)(\Delta y - y)}{\Delta x \Delta y}, \quad (5.5)$$

$$w_2 = \frac{x(\Delta y - y)}{\Delta x \Delta y}, \quad (5.6)$$

$$w_3 = \frac{(\Delta x - x)y}{\Delta x \Delta y}, \quad (5.7)$$

$$w_4 = \frac{xy}{\Delta x \Delta y}. \quad (5.8)$$

Here, Δx and Δy are fractional directions in the x and y directions respectively and $\Delta x \Delta y$ is the cell volume. At the boundaries of our domain, we have to account for the fact that only half of the grid cells are contributing (REF).

Once the charge has been distributed, the charge density, ρ , is then computed by dividing the total charge by the volume of each grid cell.

5.2.2 Solving for Φ

Once we have the charge density in hand, we can plug that into the right hand of [EQ 3.2] and put it into a finite-difference form using central differencing (also called a five-point difference),

$$\frac{\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j}}{\Delta x^2} + \frac{\phi_{i,j-1} - 2\phi_{i,j} + \phi_{i,j+1}}{\Delta y^2} = -\frac{\rho_{i,j}}{\epsilon_0}. \quad (5.9)$$

The right hand side of [EQ 5.9] is recast in a way that allows us to use an iterative technique to solve for the potential which is based on the Gauss-Siedel method ² (REF),

$$\phi = \frac{1}{M_{ii}} \left[b_i - \sum_{j=1}^{i-1} M_{ij} \phi_j - \sum_{j=i+1}^n M_{ij} \phi_j^k \right], \quad (5.10)$$

where M is a stencil matrix, and b is the density term which incorporates the Boltzmann treatment of the electrons,

$$b = \left(\frac{e}{\epsilon_0} \right) n_i - n_0 e^{\phi/T_e}, \quad (5.11)$$

where e is the elementary charge of the electron, ϵ_0 is the vacuum permittivity, n_i is the ion density, n_0 is the background density, and T_e is the electron temperature in eV.

²Up on your linear algebra? The Gauss-Siedel method is one in which you are solving the matrix equation $Ax = b$ for x.

5.2.3 Field solvers

With the potential calculated, we can put that into [EQ 5.1] and use a two-point difference to calculate the electric field at those points,

$$\vec{E}_x = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x}, \quad (5.12)$$

and

$$\vec{E}_y = \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2\Delta y}. \quad (5.13)$$

5.2.4 Add/Move Particles

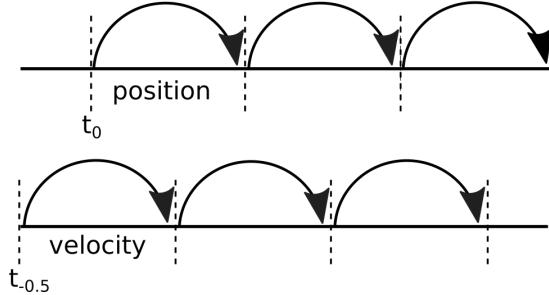


Figure 5.3: An illustration of the leap-frog method. We time-center the force while advancing the particle's velocity, and likewise time-center the velocity while advancing a particle's position.

Using the force (EQ 5.3), we then advance our particles one time step Δt . To do this, the leapfrog method is commonly employed [FIG 5.4] by taking EQ. 5.3 and 5.4 and replacing them with a finite difference,

$$\frac{v_{new} - v_{old}}{\Delta t} = \frac{q}{m} \vec{E} = \vec{F}_{old}, \quad (5.14)$$

and

$$\frac{x_{new} - x_{old}}{\Delta t} = v_{new}. \quad (5.15)$$

This requires the velocity to be pushed back to a negative half time step using the force calculated at $t = 0$, and for resulting calculations (i.e. the electric field) to be adjusted in such a way that they appear at the same time.

Chapter 6

Simulation results

Physicists like to think that all you have to do is say, these are the conditions, now what happens next?

Richard Feynman

6.1 Simulation parameters

Table 6.1: Domain parameters

nx	ny	dt (s)	dh	T_e (eV)	T_i (eV)
25	20	7.01e-7	λ_{De}	2	0.1

The presence of a charged dust particle influences the local electric potential and electric field in the background plasma which, on a larger scale, affects the properties of that plasma (REF).

In the cases that follow, a dust grain with a diameter of 1 mm and a negative potential was simulated on a grid. Electrons were treated as a background fluid, which reduced computational expense and allowed the focus to remain on the ion flow. Simulated ion macroparticles were introduced along the y-axis, throughout the run, and their

paths were calculated through the integration of the equations of motion (EQ) and (EQ) with the equations of motion given by (EQ) and (EQ). Five hundred timesteps, representing approximately 3.5 ms of dynamics, were sufficient for the plasma to reach a steady state. (TABLE) summarizes the dust parameters while (TABLE) and (TABLE) summarizes the plasma parameters for the two test cases: a heavy-ion solar wind, and an argon plasma.

Table 6.2: Solar wind parameters

Ion species	Mass (AMU)	v_{th} (km/s)	V_{dust} (V)	d_{dust} (mm)
O	16	10	-15	1

Table 6.3: Argon plasma parameters

Ion species	Mass (AMU)	v_{th} (km/s)	V_{dust} (V)	d_{dust} (mm)
Ar	40	1.5	-5	1

6.1.1 Boundary conditions

Dust structure

This model uses a rectangular domain consisting of a number of cells, and [FIG] shows how Dirichlet boundary conditions on the dust particle were applied to the stencil matrix used to calculate the potential. Ions that passed into the region occupied by the dust (as well as any ions that left the confines of the domain) were absorbed and removed from the simulation.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 6.1: How a dust particle is represented using a rectangular domain. Ions that pass from cells indexed with a 0 into cells indexed with a 1 are removed from the simulation (absorbing boundary conditions). Each cell is $1 \lambda_D$ in length.

6.2 2D PIC Results

6.2.1 Test case: Single dust particle

Solar wind flow around dust particle

Oxygen ions tend to make up the majority of heavy ions observed in the solar wind (REF). In this test case, an O+ plasma ($m = 16$ AMU) with a $v_{th} = 10$ km/s and $T_i = 0.1$ eV. [FIG] shows a color map of the ion density after solar wind flowing around a dust particle reached steady state. The number of timesteps in this simulation was 500, corresponding to a runtime of approximately 3.5 ms. Features of note include a region of high density in front of the dust particle and ion focusing behind the dust particle, which is a feature observed in other computational research (REF).

[Consider: Shock wave formation + wake?]

Argon plasma flow around dust particle

Low temperature, low pressure Ar plasma is typical for laboratory study (REF). In this test case, an Ar+ plasma ($m = 40$ AMU) with a $v_{th} = 1.5$ km/s and $T_i = 0.1$ eV. [FIG] shows a color map of the ion density after the Ar plasma flow around the dust

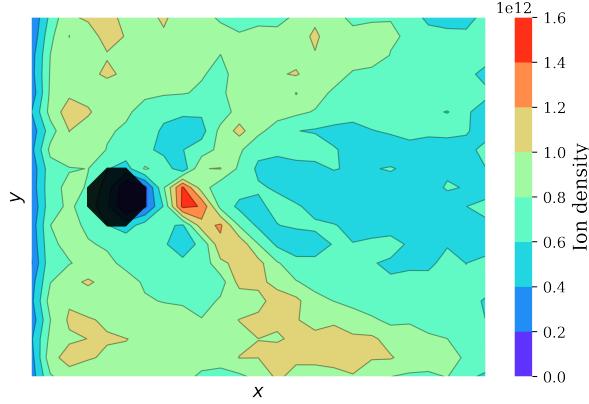


Figure 6.2: Contour plots of the density profile resulting from the flow of heavy ions in a solar wind around an individual dust particle. Simulation parameters can be found in Section 6.1. A time-lapse video of the ion density profile can be found at <https://youtu.be/SpYI5T0MIPQ>.

particle reached steady state. The number of timesteps in this simulation was 500, corresponding to a runtime of approximately 3.5 ms. Ion focusing can be found behind the dust particle in this environment as well, however the effect is not as strong as the ions are heavier and traveling at slower speeds.

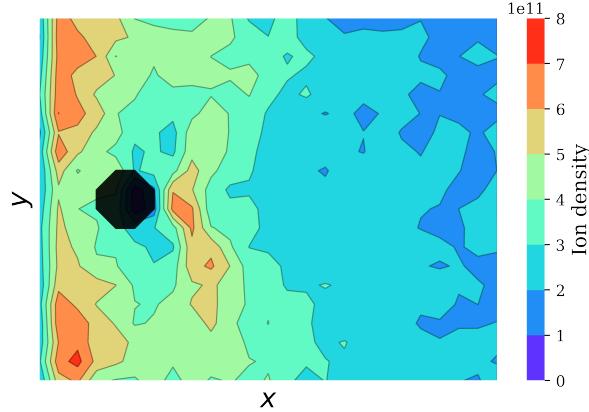


Figure 6.3: Contour plots of the density and plasma potential profiles resulting from the flow of argon plasma around an individual dust particle. Simulation parameters can be found in Section 6.1. A time-lapse video of the ion density profile can be found at <https://youtu.be/WSDmXrQoUAU>.

6.2.2 Test case: Dust chain

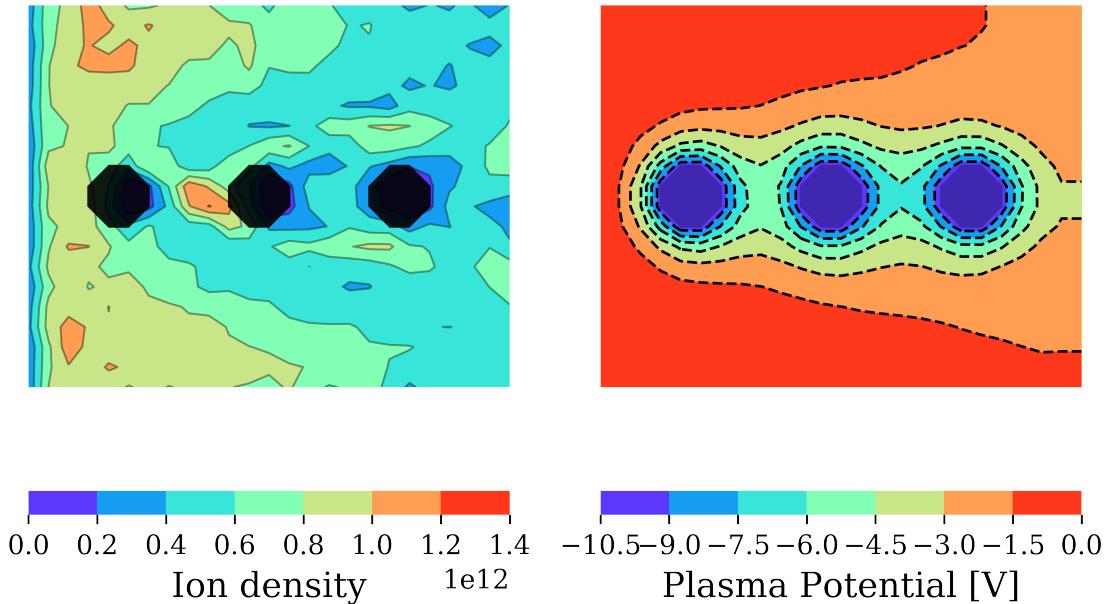


Figure 6.4: Contour plots of the density (LEFT) and plasma potential (RIGHT) profiles resulting from the flow of argon plasma around a dust chain. Dust particles are spaced $4\lambda_{De}$ apart. Simulation parameters can be found in Section 6.1. A time-lapse video of the ion density profile can be found at <https://youtu.be/VqtAZshY5xQ>.

Holding the parameters of the two environments (solar wind and argon plasma) constant, two additional dust particles were included in the simulation to create a dust chain. The interparticle distance was four grid cells, or about $4 \lambda_{De}$.

6.3 Why open source?

For those without the resources to access commercial products, (i.e. students and small research groups), free and open source tools offer a means by which to pursue research interests. From a philosophical standpoint, free and open source platforms offers a level of transparency that encourages community contribution. Software can be developed and shared with others either for critique, as an invitation for collaboration, or as a tool

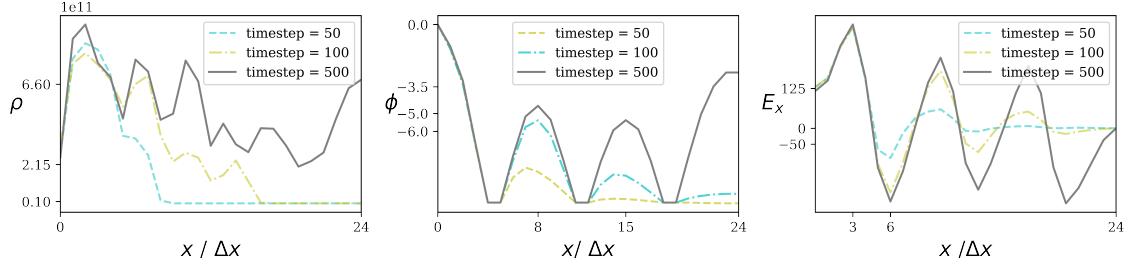


Figure 6.5: Line plots of ion density (LEFT), plasma potential (MIDDLE), and electric field in the x -direction (RIGHT) taken at $ts = 50$, $ts = 100$, and $ts = 500$.

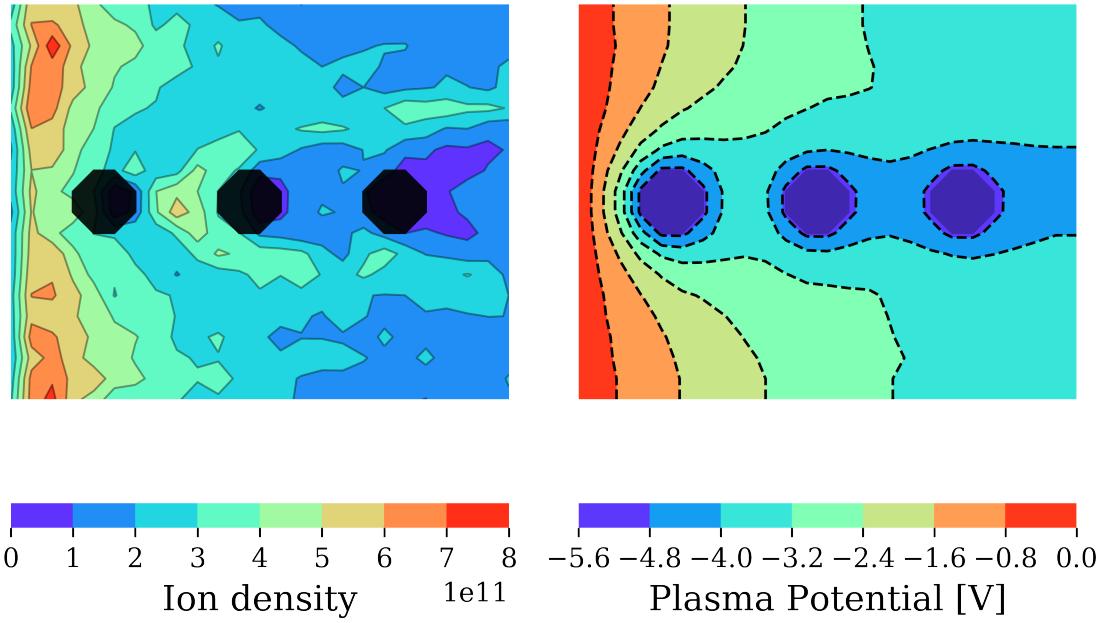


Figure 6.6: Contour plots of the density (LEFT) and plasma potential (RIGHT) profiles resulting from the flow of argon plasma around a dust chain. Dust particles are spaced $4\lambda_{De}$ apart. Simulation parameters can be found in Section 6.1. A time-lapse video of the ion density profile can be found at <https://youtu.be/O5QVE-SgT10>.

for others to take and modify for their individual needs.

Python, a popular open source programming language, is widely used primarily to manage large volumes of data in fields such as astronomy (REF), in part because of its ease of use, versatility, and readability compared to other lower level programming

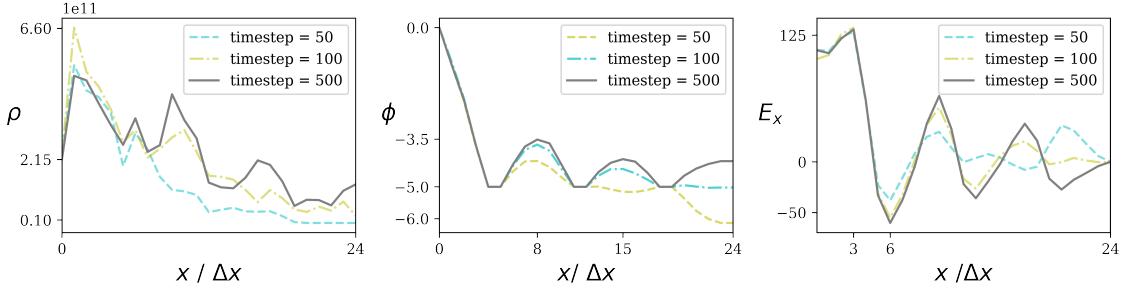


Figure 6.7: Line plots of ion density (LEFT), plasma potential (MIDDLE), and electric field in the x -direction (RIGHT) taken at $ts = 50$, $ts = 100$, and $ts = 500$.

languages. The community aspect of code developing and code sharing is reinforced through the use of web-based hosting services such as GitHub¹, and interactive projects such as Jupyter Notebook².

For all it's advantages, there are some disadvantages to consider; this is especially true when wading into under developed (or undeveloped!) programming territory. One of the main cost-benefit analyses to consider is time: commercially available software (e.g. MATLAB) have built-in features and functions that help you get to the science faster than trying to build them on your own (even with help from external sources).

When considering how to carry out simulations effectively, it is here where Python and MATLAB fall short. Even simple simulations like the ones discussed above took on the order of minutes to carry out, meaning that even marginal improvements to code (a finer mesh size, more particles simulated) would further bog down computational speed. This makes these platforms ideal for teaching environments, but less practical for research purposes—more robust languages such as C, while they have a higher learning curve, would be necessary for the sake of efficiency.

The above work was coded in MATLAB, however the Appendix contains 1D code that was developed in Python, which was the starting point for this project, as well as a link to the GitHub repository where all of the source code (1D and 2D) can be found.

¹<https://github.com/>

²<http://jupyter.org/>

6.4 Future work

A fully 3D self-consistent simulation that includes molecular dynamics and that account dust charging is the type of model that would most closely capture the dynamics of a dusty plasma; however, a 2D model that takes interparticle interactions into account would be a natural place to start. Interparticle interactions (collisions) would be addressed through the addition of a Monte Carlo Collisions handle which is an algorithm that checks each particle for a collision, determines the type of collision (inelastic, elastic) and assigns the appropriate reaction (ionization, excitation, etc.).

Using the model as it stands, the effects of ion flow around dust particles of different shapes ("rough" vs "symmetric") could be investigated, as well as the dynamics resulting from larger collections of smaller dust particles. Another thing that could be studied is the effect of ion collisions with the dust particle. Using SRIM (The Stopping and Range of Ions in Matter) to gather data on the stopping range of Ar and O in a dust particle of a given composition, a dust particle could be simulated with a number of layers consisting of different boundary conditions and the PIC-simulated stopping range can be compared with expected results.

References

- de Angelis, U., Tolias, P., Ratynskaia, S. (2012). Effects of dust particles in plasma kinetics: Ion dynamics time scales. *Physics of Plasmas*, 19(7), 073701.
- Baklanov, M. R., de Marneffe, J. F., Shamiryan, D., Urbanowicz, A. M., Shi, H., Rakimova, T. V., ... Ho, P. S. (2013). Plasma processing of low-k dielectrics. *Journal of Applied Physics*, 113(4), 4.
- Blandón, J. S., Grisales, J. P., Riascos, H. (2017, June). Electrostatic plasma simulation by Particle-In-Cell method using ANACONDA package. In *Journal of Physics: Conference Series*(Vol. 850, No. 1, p. 012007). IOP Publishing.
- Donnelly, V. M., Kornblit, A. (2013). Plasma etching: Yesterday, today, and tomorrow. *Journal of Vacuum Science Technology A: Vacuum, Surfaces, and Films*, 31(5), 050825.
- Garscadden, A. (1994). Nucleation, growth, and morphology of dust in plasmas. *Pure and applied chemistry*, 66(6), 1319-1326.
- Kanarik, K. J., Lill, T., Hudson, E. A., Sriraman, S., Tan, S., Marks, J., ... Gottscho, R. A. (2015). Overview of atomic layer etching in the semiconductor industry. *Journal of Vacuum Science Technology A: Vacuum, Surfaces, and Films*, 33(2), 020802.
- Lacci, K. M., Dardik, A. (2010). Platelet-rich plasma: support for its use in wound healing. *The Yale journal of biology and medicine*, 83(1), 1.
- Markidis, S., Henri, P., Lapenta, G., Ronnmark, K., Hamrin, M., Meliani, Z., Laure,

E. (2013). The fluid-kinetic particle-in-cell solver for plasma simulations. arXiv preprint arXiv:1306.1089.

Pinhão, N. (2007). pypk-A Python extension module to handle chemical kinetics in plasma physics modeling.

Ricci, P., Lapenta, G., de Angelis, U., Tsytovich, V. N. (2001). Plasma kinetics in dusty plasmas. *Physics of Plasmas*, 8(3), 769-776.

Seo, H., Kim, S. B., Song, J., Kim, Y., Soh, H., Kim, Y. C., Jeon, H. (2002). Low temperature remote plasma cleaning of the fluorocarbon and polymerized residues formed during contact hole dry etching. *Journal of Vacuum Science Technology B: Microelectronics and Nanometer Structures Processing, Measurement, and Phenomena*, 20(4), 1548-1555.

Shoeb, J., Kushner, M. J. Comparison Of Cleaning And Damage Of Porous Low-k SiCOH In Ar/O₂ And He/H₂ Plasmas With UV/VUV Fluxes.

Shoeb, J., Wang, M. M., Kushner, M. J. (2012). Damage by radicals and photons during plasma cleaning of porous low-k SiOCH. I. Ar/O₂ and He/H₂ plasmas. *Journal of Vacuum Science Technology A: Vacuum, Surfaces, and Films*, 30(4), 041303.

Shukla, P. K., Mamun, A. A. (2002). Introduction to dusty plasma physics. IoP Publishing.

Somashekhar, A., Ying, H., Smith, P. B., Aldrich, D. B., Nemanich, R. J. (1999). Hydrogen Plasma Removal of Post-RIE Residue for Backend Processing. *Journal of The Electrochemical Society*, 146(6), 2318-2321.

Appendix A

1D PIC Code in Python

A.1 Two-stream instability

[TABLE: SIMULATION PARAMETERS (NX , NY , DT , $N\dots$)]

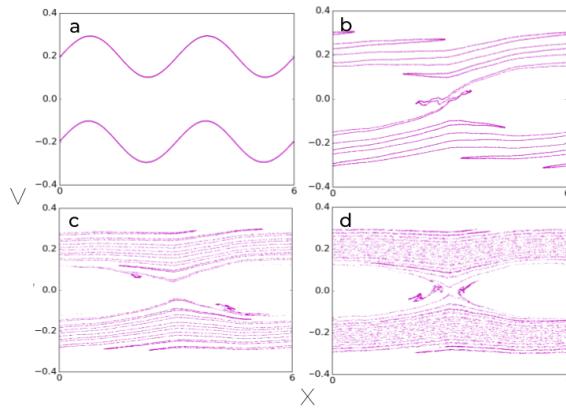


Figure A.1: Top: Two opposing electron streams moving in a background of ions. An instability will develop due to charge bunching as the streams move through each other. Bottom: Evolution of the electron-electron two stream instability in phase space at times a) $t=0$, b) $t = 49$, c) $= 99$, and d) $t = 199$. The initial velocities are ± 0.2 , the grid has 1000 cells, there are 20000 particles per beam, and the beams have an initial sinusoidal perturbation of mode 2.

Building computational environments is a modular process. As such, it's critical to test as you go. We'll start with the 1D case which consists simply of electrons moving

in a background of ions. To this program, we'll apply a benchmark test: the two-stream instability. This problem, illustrated in [FIG] models two opposing streams of electrons. [FIG] shows the time evolution of this stream, given an initial perturbation, in phase space (position vs velocity instead of the familiar position/velocity vs time).

What do we mean by perturbation? Well consider that a perfectly stationary beam of like charges will be equally spaced from each other in some equilibrium configuration. Disturbing a single particle from its equilibrium position is going to oscillate around that point at some frequency. To disturb *all* of the charged particles in this beam, as is the case in these simulations, we get a sinusoidal response where a particle's position is described as follows,

$$x = x_0 + x_1 \cos\left(x_0 \frac{2\pi n}{L}\right), \quad (\text{A.1})$$

Here L is the length of the domain and n is the mode of excitation of the wave.

Allow that initial perturbation to run over time, and adding the second opposing beam in the mix, you get an instability that grows exponentially over time (Landon and Birdsall, 2005). We see this play out in [FIG].

Appendix B

Sample Python code

For the most current version of source code used in this project (in both 1D and 2D) visit my [GitHub](#) or copy and paste the url: <https://github.com/space-isa/PIC>. Included in the repository is an annotated notebook created using the Jupyter platform, as well as the .py and .m files that can be downloaded and run. Take it, reproduce it, break it, change it, share it! Below is a sample of code showing the main cycle used to run the simulations presented in Appendix A.

```
start = time.clock()

NP = 0
print("Calculating...")

for count in range(0, iterations):

    q = np.zeros((nx, ny))
    rho = np.zeros((ny, nx))

    for p in range (1, NP):
        fi = (1 + p_pos[p,0]) / (dh)
```

```

i = np.floor(fi)
hx = fi - i

fj = (1 + p_pos[p,1]) / (dh)
j = np.floor(fj)
hy = fj - j

q[i,j] = q[i,j] + (1-hx) * (1-hy)
q[i+1, j] = q[i+1, j] + hx * (1-hy)
q[i, j+1] = q[i, j+1] + (1-hx) * hy
q[i+1, j+1] = q[i+1, j+1] + hx * hy

rho = (sw + q_mp * q) / (dh * dh)

rho[0,:] = 2 * rho[0,:]
rho[-1, :] = 2 * rho[-1, :]
rho[:, 0] = 2 * rho[:, 0]
rho[:, -1] = 2 * rho[:, -1]

rho = rho + (1 * 10 ** 4)
#print(rho)

#potential solver
V = solver_2d(rho, tol, Ti, n0, V_ref, QE)

#E field solver
Ex = np.zeros([nx, ny])

```

```

Ey = np.zeros([nx, ny])
E = np.zeros([nx, ny])

#internal nodes
Ex[1:nx-1, :] = V[0:nx-2, :] - V[nx-(nx-2):, :]
Ey[0: , 1:nx-1] = V[:, 0:ny-2] - V[:, 2:ny]

#boundaries
#multiplied by 2 to keep values equivalent to internal nodes
Ex[0, :] = 2* (V[0, :] - V[1, :])
Ex[nx-1, :] = 2 * (V[nx-2, :] - V[nx-1, :])

Ey[:, 0] = 2 * (V[:, 0] - V[:, 1])
Ey[:, ny-1] = 2 * (V[:, ny-2] - V[:, ny-1])

Ex = np.floor (Ex / (2 * dx) )
Ey = Ey / (2 * dy)

#generate particles

if NP + np_in >= N:
    np_in = N - NP

#insert particles
 #(NOTE: save this for after 2d environment works)

#x position

```

```

p_pos[NP:NP+np_in, 1:] = np.random.rand(np_in,1) * dh

#y position
p_pos[NP:NP+np_in, 1:] = np.random.rand(np_in,1) * Ly

#sample Maxwellian in x,y
#add drift velocity in x

p_velo[NP:NP+np_in, 1:] = v_drift + (-1.5 + np.random.rand(np_in,1)
    + np.random.rand(np_in,1) + np.random.rand(np_in, 1)) * vth
p_velo[NP:NP+np_in, 1:] = 0.5 * (-1.5 + np.random.rand(np_in,1)
    + np.random.rand(np_in,1) + np.random.rand(np_in, 1)) * vth

#move particles
p = 1

while p <= NP:

    fi = 1 + p_pos[p,0]/dx
    i = np.floor(fi)
    hx = fi - i

    fj = 1+ p_pos[p,1]/dy
    j = np.floor(fj)
    hy = fj - j

    E = ([Ex[i, j], Ey[i,j]]) * (1-hx) * (1-hy)
    E = E + ([Ex[i + 1, j], Ey[i + 1, j]]) * hx * (1-hy)

```

```
E = E + ([Ex[i, j + 1], Ey[i + 1, j]]) * (1-hx) * hy
E = E + ([Ex[i + 1, j + 1], Ey[i + 1, j + 1]]) * hx * hy

F = QE * E
a = F/MI

p_pos[p, :] = p_velo[p, :] + a * dt
p_velo[p, :] = p_pos[p, :] + p_velo[p, :] * dt

if p_pos[p,1] < 0:
    p_pos[p,1] = -p_pos[p,1]
    p_velo[p,1] = -p_velo[p,1]

p = p + 1
print(p_pos, p_velo)

print("Clocking in at %s seconds" % (time.clock() - start))
```