Mathematisch-Naturwissenschaftliche Fakultät
Ernst-Moritz-Arndt-Universität Greifswald

Master Thesis

Hybrid Plume Modeling

Daniel Lars Kahnfeld
Greifswald, August 20, 2015

Referees:
Prof. Dr. Ralf Schneider
PD Dr. Bernd Bruhn

Contents

1 Motivation

For spacecrafts the concept of ion thrusters presents a very efficient method of propulsion. Ion thrusters generate a low thrust with much higher efficiency than chemical propulsion systems [1] and are commonly used on satellites in earth orbits, such as "Gravity field and steady-state Ocean Circulation Explorer (GOCE)" [2], or on long term deep space missions like "Deep Space 1" [3] or "Dawn" [4].

Thrust is generated by accelerating ions of a plasma discharge and exhausting them into space. The plasma within the thruster channel is dominated by electric and magnetic fields, plasma-wall-interaction and non-linear effects. The shape and size of the plume have to be considered in the design of ion thrusters to account for possible damages caused by ion sputtering, but experimental access is difficult [1].

Plasma simulations offer the means to understand the plasma physics within an ion thruster and can aid the design of new thruster concepts. A widely applied method is the Particle-in-Cell (PIC) scheme, simulating the trajectories of superparticles consisting of many real particles. The simulation of an ion thruster is difficult since the electron Debye length of around $10^{-3}\,\mathrm{m}$ has to be resolved spatially on a domain that covers the dimensions of the thruster and the plume that can extend to a distance from the thruster of some m. The electron plasma frequency has to be resolved, resulting in timesteps of about $10^{-8}\,\mathrm{s}$, for a simulation which should cover a time in the range of up to one second. Even with modern hardware, state-of-the-art features such as similarity scaling [5] and non-uniform grids [6] have to be used to make simulation of an ion thruster conceivable for only short times.

The simulation of the plume is of great difficulty as plume effects become important on the length scale of several meters, which can not be covered by the PIC method. To make plume simulations possible, a hybrid PIC method can be introduced, modeling the electrons as a fluid while retaining the kinetic ions. The electron Debye length in the range of $10^{-3}\,\mathrm{m}$ and electron plasma frequency of $\sim 10^{8}\,\mathrm{Hz}$ no longer have to be resolved, reducing computational costs significantly as the new limiting length and time scales are the ion Debye length of $\sim 1\,\mathrm{m}$ and ion plasma frequency of $\sim 10^{4}\,\mathrm{Hz}$. As ion thruster plasmas are non-thermal and dominated by the magnetic field and sheath effects, the fluid electron model with a Maxwellian velocity distribution cannot be applied within the thruster channel. Although kinetic effects are transferred by the particles into the plume as shown in [7], a hybrid model can still be applied as an approximation, since particle interaction is minimal in that area. Hybrid plume models have been used for Hall thrusters successfully [8].

In this thesis the application of a hybrid plume model to the simulation of the Highly Efficient Multistage Plasma thruster (HEMP-T) will be presented, along with a comparison to the results of the fully kinetic PIC scheme used for the simulation of the thruster, to understand the benefits and limitations of such a fluid model for ion thrusters. The first part of this thesis covers the basic concepts of ion thrusters and a more detailed description of the HEMP-T. The second part contains a detailed description of the electrostatic PIC scheme and a short description of the code used to simulate HEMP-T. Newly implemented numerical concepts for the hybrid plume model and their validation will be presented in the third chapter. The fourth chapter will discuss the physical results of the hybrid plume model along with a quantitative comparison between the fully kinetic PIC scheme and the hybrid model for different applications.

2 Electric propulsion using plasmas

In this chapter, a short overview of different thruster concepts used in space propulsion is given. The basic equations of space propulsion are discussed along with different thruster concepts. The chapter concludes with a more detailed description of the HEMP-T.

2.1 Basics of electric propulsion

To enable movement in space, a propellant of mass m is expelled with exhaust velocity v_{ex} from a thruster mounted on a spacecraft, generating a net thrust T. For a straight line movement and constant v_{ex} the thrust is defined as

$$T = -\frac{\mathrm{d}p}{\mathrm{d}t} = v_{ex}\frac{\mathrm{d}m}{\mathrm{d}t} \,,$$

with the craft's momentum $p(t) = m(t)v(t)$. An important characterization of a spacecraft's operating range is given by the change of velocity Δv that can be generated by the thruster. The mass of the craft $m(t)$ is given as the sum of its fuel mass m_f and its empty mass m_0. If a constant mass flow rate $\dot{m} = \dot{m}_f$ is assumed, Δv is then given by the rocket equation [1]

$$\Delta v = v_{ex} \ln\left(\frac{m_0}{m_0 + m_f}\right) \,. \tag{2.1}$$

It is obvious that an increase in the exhaust velocity of the propellant will increase the spacecraft's range. Vice versa this means for constant rocket mass that the same Δv can be achieved with a lower amount of fuel mass m_f, hence increasing the empty weight m_0 of the spacecraft allowing more freight to be transported in space. This results in an increase in payload ratio, the fraction of the mass of the transported freight compared to the overall weight of the spacecraft.

The exhaust velocity thus is a measure of a thruster's fuel efficiency. This is expressed by the specific impulse [1]

$$I_{sp} = \frac{v_{ex}}{g} \,,$$

with the gravitational acceleration $g \approx 9.81\,\mathrm{m/s^2}$. As real mass exhaustion does not occur in a straight line movement, both the thrust as well as the specific impulse have to be adjusted using the half divergence angle α_{eff} of the mass exhaust

$$\begin{aligned} T &= T_0 \cos\alpha_{eff} \\ I_{sp} &= I_{sp,0} \cos\alpha_{eff} \end{aligned} \tag{2.2}$$

with the subscript 0 denoting the straight line quantities.

In space travel there are two commonly used types of thrusters relying on mass exhaustion. The first and most well-known ones are called chemical thrusters. The spacecraft either carries solid fuel or liquid fuel and oxidizer, which are injected into the thruster's combustion chamber and burned, resulting in an exhaust of the ensuing reaction products. The maximum exhaust velocity is around 2000 m/s [9], thus limiting the maximum efficiency of chemical thrusters and resulting in a need to use different thruster concepts to gain higher efficiency, such as ion thrusters. Chemical thrusters typically have a I_{sp} of 100 s (solid fuel) to 500 s (liquid fuel and oxidizer) but produce a large amount of thrust in the range of some kN [10].

The other type which was developed only recently is the concept of an ion thruster. The basic idea is that the fuel, usually a noble gas, is ionized within the thruster channel using electrons and is then accelerated toward the thruster exit, generating a thrust by exhaustion of ions. In the thruster channel a plasma is formed. Ion thrusters are very fuel efficient compared to chemical thrusters with an $I_{sp} \approx 5000\,\mathrm{s}$. An ion thruster's efficiency μ_{tot} is given by the ionization efficiency μ_{ion}, the beam power efficiency μ_B and the half divergence angle α_{eff} [1]

$$\mu_{tot} = \mu_B \cdot \mu_{ion} \cdot \cos^2\alpha_{eff}\,.$$

The ionization efficiency is the ratio of ion current to total mass flow and the beam efficiency is the ratio of beam power to total power consumption of the thruster. In order to make an ion thruster more efficient it is necessary to limit the ion beam divergence and to maximize ionization rates within the thruster channel. Ion thrusters usually create a small thrust in the range of several mN [1], thus making them appropriate for long term use on light-weight spacecrafts, such as satellites orbiting the earth or for use on scientific exploration missions.

In space missions the integral thrust is of importance, denoting the overall thrust generated over the lifespan of the propulsion device. As ion thrusters only produce a low net thrust, the runtime of such a thruster increases in order to generate an integral thrust comparable to that of a chemical thruster. In ion propulsion the plasma-wall interactions limit the lifetime of such a device, either by sputtering of the grid on grid thrusters [1] or due to plasma-wall interactions within the thruster channel, as is the case with hall thrusters [11]. Thus, it is of great importance to create thruster concepts with as large of a lifetime as possible by reducing the sputter rate. One such concept is that of the HEMP-T which is presented in the next chapter.

2.2 The HEMP ion thruster

There are different types of ion thrusters, distinguished by their operating principle. The simplest design is to ionize the propulsion gas in an ionization chamber using an electron beam and accelerating the ions in the thruster channel towards the exit using an electrostatic field generated by an anode in the inner thruster and a grid serving as cathode at the end of the channel. This design has only a limited lifetime as the grid is sputtered by the ions passing through.

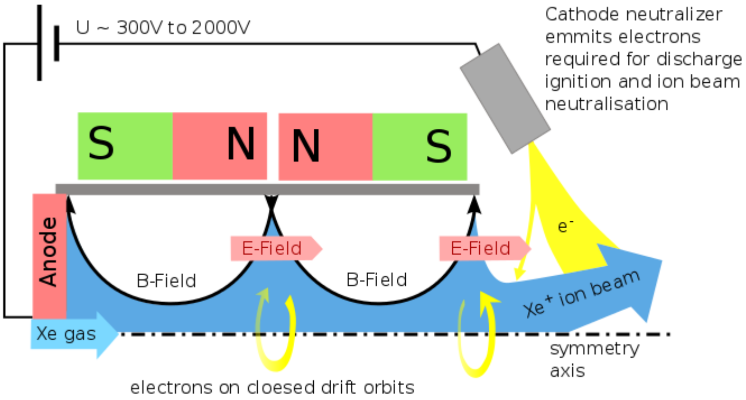


Figure 2.1: Basic structure of the HEMP-DM3a thruster with the cusp structure of the magnetic field.

To avoid such effects, gridless ion thruster concepts have been developed, most notably the Stationary Plasma Thruster (SPT), as described in [12], and the High Efficient Multistage Plasma Thruster (HEMP-T). The HEMP-DM3a design, as shown in fig. 2.1, possesses a rotational symmetry. The bottom of the channel contains the anode with a voltage of 500 V, with the cathode supplied by an electron beam outside the channel which also serves as electron source and neutralizer for the exhausted ions. The thruster channel is surrounded with permanent magnet rings of opposite magnetization. This results in a nearly constant magnetic field at the symmetry axis of the thruster with the exception of cusp regions, where two rings with opposite magnetization are located next to each other. The inner boundary of the thruster channel is made up of a dielectric ceramic consisting of Boron Nitrite which has a high threshold to reduce sputtering [13]. The thruster is concluded with a grounded metal plate attached outside the dielectric at the thruster exit. A more detailed description of the thruster can be found in [14].

Thruster channel length	$L = 5.1\,\mathrm{cm}$
Thruster channel radius	$R = 0.9\,\mathrm{cm}$
Discharge voltage	$U_a = 500\,\mathrm{V}$
Approx. magnetic field	$B = 0.1\,\mathrm{T}$
Electron Larmor radius	$r_{L,e} = 4.8 \times 10^{-3}\,\mathrm{cm}$
Electron plasma frequency (channel)	$\omega_{p,e} = 8.98\,\mathrm{GHz}$
Electron plasma frequency (plume)	$\omega_{p,e} = 564\,\mathrm{MHz}$
Electron Debye length (channel)	$\lambda_{D,e} = 1.49 \times 10^{-5}\,\mathrm{m}$
Electron Debye length (plume)	$\lambda_{D,e} = 1.49 \times 10^{-3}\,\mathrm{m}$
Ion Larmor radius	$r_{L,i} = 2.3\,\mathrm{cm}$
Ion plasma frequency (channel)	$\omega_{p,i} = 18.3\,\mathrm{MHz}$
Ion plasma frequency (plume)	$\omega_{p,i} = 14.8\,\mathrm{kHz}$
Ion Debye length (channel)	$\lambda_{D,i} = 1.49 \times 10^{-5}\,\mathrm{m}$
Ion Debye length (plume)	$\lambda_{D,i} = 1.49\,\mathrm{m}$
Neutral density	$n_n = 10^{21} - 10^{19}\,\mathrm{m}^{-3}$
Electron and ion density	$n_e = n_i = 10^{18}\,\mathrm{m}^{-3}$
Electron and ion temperature	$T_e = T_i = 14\,\mathrm{eV}$
Neutral temperature	$T_n = 0.069\,\mathrm{eV}$ (800 K)

Table 2.1: Operational parameters of the HEMP-DM3a thruster model inside the thruster channel.

During operation the thruster channel is filled with Xenon gas. The electrons, generated by the neutralizer in the plume, are accelerated into the thruster channel by the electric field building up from the anode to the exit. Near the thruster axis the electrons are accelerated toward the anode, ionizing the gas and creating mostly $\mathrm{Xe}^{+}-$ ions. Away from the axis the cusp structure acts as a magnetic mirror, trapping and kinetically heating the electrons, thus strongly increasing ionization efficiency and lifetime of the electrons within the thruster channel.

Due to the strong parallel transport of charged particles near the axis, the potential has a flat structure throughout the channel with only little deviations in the cusp regions. The radial acceleration of ions is minimal, with ion energies below the sputtering threshold of the dielectric channel walls, minimizing erosion of the dielectric channel walls. A large potential drop occurs at the thruster exit, accelerating the ions into the plume, which is similar to a grid thruster [10] but without a grid that is subject to sputtering damage.

The plume is of great importance, as a high beam divergence leads to a reduced thruster efficiency, as given in eq. (2.2). Radially accelerated ions may also lead to sputtering damages of spacecraft structures in a large distance from the thruster exit. Hence knowledge of a thruster's plume structure and beam divergence is crucial. To reduce radial acceleration, the ions are partially neutralized by the electron source. The grounded metal further reduces beam divergence. The HEMP-DM3a reaches a specific impulse of up to 2300 s with a thrust $> 44\,\mathrm{mN}$ [15]. The operational parameters are listed in table 2.1. To be able to understand the physics of ion thrusters kinetic plasma models are needed, which will be introduced in the next chapter.

3 Simulation of plasmas

3.1 Basic plasma description

Before introducing the PIC simulation method for plasmas, the basic plasma properties and an introduction to its theoretical description is given. Furthermore, the fluid description of plasmas, plasma sheath effects and collisional effects in plasmas are discussed.

3.1.1 Kinetic description of plasmas

The kinetic description of a plasma species s is given by the distribution function $f_s(\vec{x}, \vec{v}, t)$, which in general is seven dimensional. The evolution of f_s is obtained with the total time derivative of f_s, resulting in the kinetic equation

$$\frac{\partial f_s}{\partial t} + \vec{v} \nabla_{\vec{x}} f_s + \frac{\vec{F}}{m_s} \nabla_{\vec{v}} f_s = \left(\frac{\partial f_s}{\partial t} \right)_{coll} , \tag{3.1}$$

with the force $\vec{F}$ acting on the particles and $\nabla_{\vec{x}}$ and $\nabla_{\vec{v}}$ denoting the gradient in coordinate and velocity space, respectively. The term $\left(\frac{\partial f_s}{\partial t}\right)_{coll}$ denotes the change of f_s in time due to collisional effects [16].

To describe a plasma entirely it would be sufficient to solve this equation for each species s. The difficulties lie in the high dimensionality of f_s since it is seven-dimensional as well as the modeling and calculation of the right hand side, which denotes the change of f_s in time due to collisional effects.

Additional equations have to be solved self-consistently to calculate $\vec{F}$. The calculation of the Coulomb force acting on each particle, which scales with the square of the particle number, can be replaced by solving macroscopic fields on a mesh, which are then obtained by solving Maxwell's equations, scaling linearly with the number of particles. In the description of the HEMP-T, only the electrostatic case is of interest, as ion currents are so small, that the induced magnetic field is negligible compared to the outside field. Thus, Maxwell's equations are reduced to the set

$$\vec{E}(\vec{x},t) = -\nabla\Phi(\vec{x},t) \tag{3.2}$$

$$\Delta\Phi(\vec{x},t) = -\frac{n(\vec{x},t)}{\varepsilon\varepsilon_0} \tag{3.3}$$

for the electric field $\vec{E}\left(\vec{x},t\right)$ and the electrostatic potential $\Phi\left(\vec{x},t\right)$, along with the charge density $n\left(\vec{x},t\right)$. The force then acting on each particle with velocity $\vec{v}$ is the Lorentz force

$$\vec{F}_L\left(\vec{x},\vec{v},t\right) = q\left(\vec{E}\left(\vec{x},t\right) + \vec{v}\times\vec{B}\left(\vec{x},t\right)\right) \tag{3.4}$$

with an external magnetic field $\vec{B}\left(\vec{x},t\right)$.

Depending on the type of plasma considered, the kinetic equation eq. (3.1) takes different forms. If the plasma particles move with high energy, the collision term nearly vanishes, resulting in Vlasov's equation for collisionless plasma. If the molecular chaos assumption is used, the Boltzmann equation is obtained. A binary collision approximation leads to the Fokker-Planck equation. A more detailed description can be found in [17].

If the distribution function is known, the macroscopic plasma properties are obtained by calculation of the moments of the distribution function. The most important moments are the 0th moment, giving the species' charge density

$$n_s\left(\vec{x},t\right) = q_s\int f_s\left(\vec{x},\vec{v},t\right)\,\mathrm{d}\vec{v}$$

and the first moment in velocity space giving the mean velocity

$$\vec{u}_s\left(\vec{x},t\right) = \frac{1}{n\left(\vec{x},t\right)}\int \vec{v}f_s\left(\vec{x},\vec{v},t\right)\,\mathrm{d}\vec{v}$$

of s. From the above equations one can see that the distribution function may be written as

$$f_s(\vec{x},\vec{v},t) = \frac{1}{q_s}n_s\left(\vec{x},t\right)\hat{f}_s\left(\vec{x},\vec{v},t\right) \tag{3.5}$$

with a normalized velocity distribution function

$$\int \hat{f}_s\left(\vec{x},\vec{v},t\right)\,\mathrm{d}\vec{v} = 1\,. \tag{3.6}$$

A thermodynamic equilibrium may be assumed by each species separately, generating a partial thermodynamic equilibrium. In this case, the distribution function takes the form of a Maxwell-Boltzmann-distribution [16]

$$f_s^m((\vec{x}, \vec{v}, t) = \frac{1}{q_s} n_s\left(\vec{x}, t\right) \left(\frac{m_s}{2\pi k_B T_s}\right)^{3/2} \exp\left[-\frac{|\vec{v}|^2}{v_{th,s}^2}\right] \tag{3.7}$$

with species temperature T_s and thermal velocity

$$v_{th,s} = \sqrt{\frac{2k_B T_s}{m_s}} \tag{3.8}$$

for each degree of freedom. This distribution gives an average velocity $\langle|\vec{v}|\rangle$ and a root-mean-square velocity $v_{rms} = \sqrt{\langle\vec{v}^2\rangle}$ of

$$\langle|\vec{v}|\rangle = \sqrt{\frac{8k_B T_s}{m_s}} \tag{3.9}$$

$$v_{rm,s} = \sqrt{\frac{3k_B T_s}{m_s}}\,. \tag{3.10}$$

In a fully equilibrated plasma all species have the same temperature. A thermodynamic equilibrium is assumed in a collision dominated plasma, with no sheath effects present. Particle motion then resembles a random movement with a mean velocity as in eq. (3.9), similar to Brownian motion [16].

3.1.2 Fluid description of plasmas

Instead of using the kinetic equation or following each particle's trajectory, a plasma may be described as a fluid carrying electric charge. For species s the fluid equation is obtained by multiplying the equation of single particle motion with the species' charge density n_s

$$m_s n_s \frac{\mathrm{d}\vec{u}_s}{\mathrm{d}t} = q_s n_s \left(\vec{E} + \vec{u}_s \times \vec{B}\right)$$

and applying the convective derivative on the species' average velocity $\vec{u}_s$

$$\frac{\mathrm{d}\vec{u}_s}{\mathrm{d}t} = \frac{\partial \vec{u}_s}{\partial t} + \left(\vec{u}_s \cdot \nabla_{\vec{x}}\right) \vec{u}_s$$

to observe the fluid elements in a fixed spatial frame. The thermal motion of particles has to be taken into account by the equation as well, generating a pressure force proportional to $-\nabla_{\vec{x}} \boldsymbol{P}$, with the stress tensor $\boldsymbol{P}$. The collisionless fluid equation of motion then becomes

$$m_s n_s \left[\frac{\partial \vec{u}_s}{\partial t} + \left(\vec{u}_s \cdot \nabla_{\vec{x}} \right) \vec{u}_s \right] = q_s n_s \left(\vec{E} + \vec{u}_s \times \vec{B} \right) - \nabla_{\vec{x}} \boldsymbol{P} \, . \tag{3.11}$$

To obtain a self-consistent solution, the equations for the electric field, eqs. (3.2) and (3.3), as well as the continuity equation

$$\frac{\partial n_s}{\partial t} + \nabla_{\vec{x}} \left(n_s \vec{u}_s \right) = 0 \tag{3.12}$$

have to be solved.

The stress tensor $\boldsymbol{P}$ is often obtained with the assumption of a Maxwellian velocity distribution for s. If the distribution is isotropic in all dimensions and heat flow is neglected, the stress tensor reduces to the scalar pressure $p_s = n_s k_B T_s$ per unit volume. Most times such a Maxwellian distribution for ions is assumed as a more complicated model for the stress tensor would increase the comlplexity of the model tremendously.

In the fluid description of a plasma it is often beneficial to neglect the mass of Maxwellian electrons, meaning they follow changes of the electric field instantaneously. For this case the Boltzmann relation for electron density

$$n_e \left(\Phi \right) = n_{e0} \exp \left[\frac{e \left(\Phi - \Phi_0 \right)}{k_b T} \right] \tag{3.13}$$

can be derived from eq. (3.11) [16]. This leads to a reduced calculational expense in solving the set of fluid equations. This relation holds if only forces on timescales much larger than that of the electron plasma frequency are considered, such that electron inertia can be neglected.

To further reduce model complexity, approximations to the fluid model may be used. One such approach links the species' temperature to its charge density

$$T_s = T_{ref,s} \left(\frac{n_s}{n_{ref}} \right)^{\alpha - 1} \, . \tag{3.14}$$

The parameter α is chosen to be in the range of $\in [1, 5/3]$, the lower limit resulting in constant temperature and the upper limit chosen to be the adiabatic constant for a monoatomic gas. This model as been previously applied to plume modeling of ion thrusters with good success [8].

The fluid description is best suited for species' in a Maxwellian regime, especially for electrons if the Boltzmann relation eq. (3.13) can be used. This simplifies the plasma description significantly when compared to the kinetic description eq. (3.1). A Maxwellian distribution is assumed if a plasma is dominated by collisions, or vice versa, if particles do not move in a straight line for long times and distances. That means the mean free paths for particle collisions need to be small. But straight line motion is also limited by external fields, via $\vec{E} \times \vec{B}-$drift and gyration around magnetic field lines in non-magnetized plasmas. Such systems may be described suitably by the fluid approach. Respectively, magnetized plasmas with effective transport along the magnetic field lines with long mean free paths are not suitable for a fluid description. Also plasma sheaths and regions with strong kinetic effects, such as particle heating are not described well by this approximation, as these present a deviation from the equilibrium assumption.

As the HEMP-T electrons are dominated by magnetic transport, electron heating in the magnetic cusp regions and plasma wall contact, it is not to be expected that a fluid description of the electrons will work well within the thruster channel, reducing the applicability of an electron fluid model to regions outside the thruster channel.

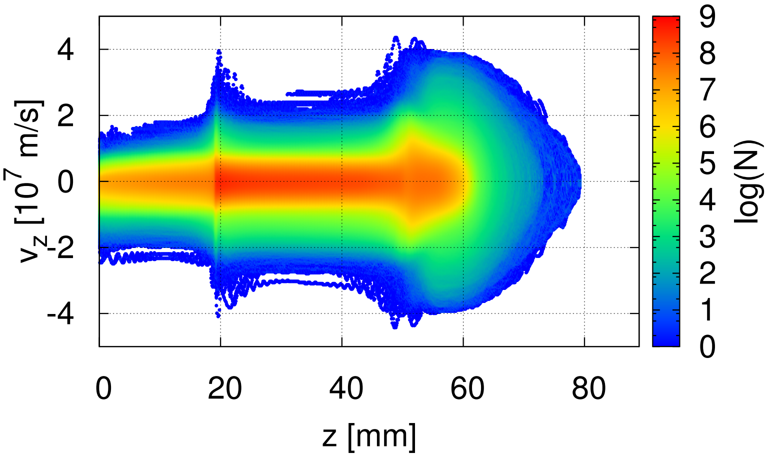


Figure 3.1: Simulation results for the axial distribution of the z-component of electron velocity [7]

The simulation results for the electron energy distribution of [7], as shown in fig. 3.1, further support that fact. On the left the axial distribution of the $z-$component of electron velocity shows that electron heating takes place mostly in the magnetic cusp region of the thruster at $r = 20\,\text{mm}$, where kinetic effects are dominant. Thus, a fluid description of electrons in the thruster channel is not expected to be suitable.

3.1.3 Plasma sheath

The plasmas in ion thrusters are strongly influenced by plasma wall interactions. The plasma sheath is dominated by a negative potential drop $\Delta\Phi$ near the wall, due to plasma electrons with higher mobility being absorbed by the wall, resulting in a negatively charged boundary. Ions are thus accelerated towards and neutralized close to the wall, while electrons with energies lower than $e\Delta\Phi$ are reflected. Hence particle densities decrease near the wall, with a stronger reduction in electron density. This situation in a plasma sheath is represented in fig. 3.2.

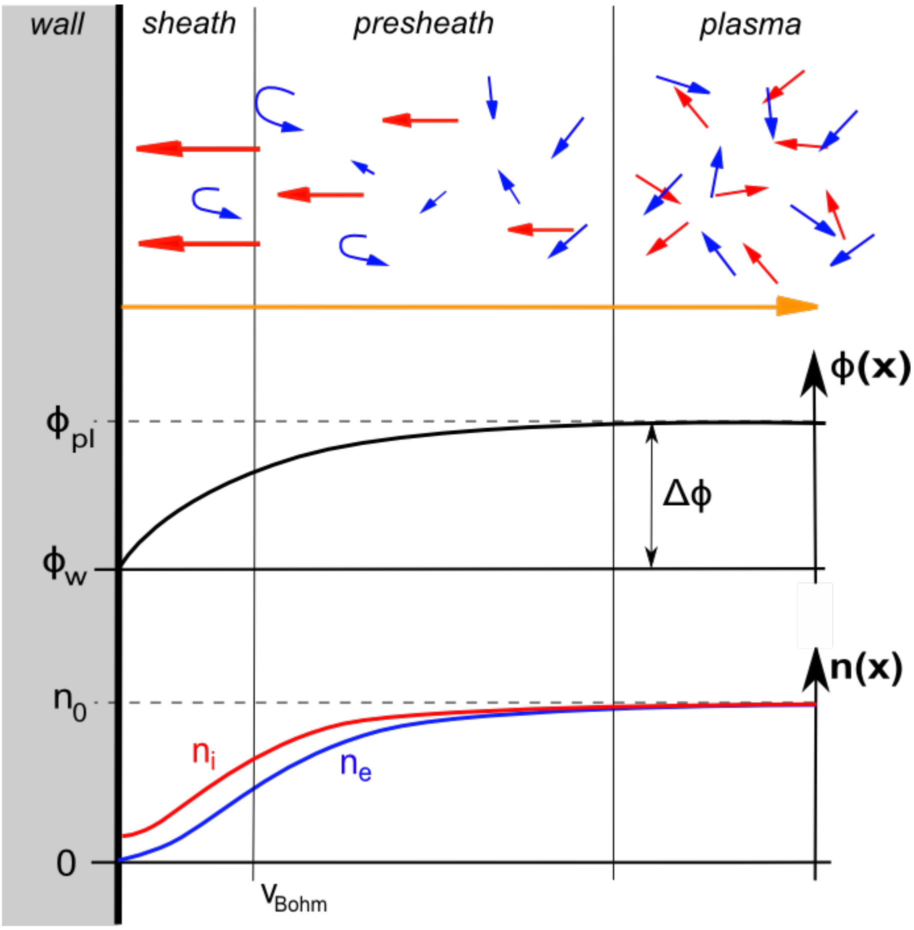


Figure 3.2: Situation in a plasma sheath: red arrows indicate ion motion, blue arrows indicate electron motion and orange arrows indicate neutral motion. [18]

In steady state the particle fluxes of electrons and ions are assumed to be equal. If one also assumes zero net current $j_i = j_e$ and quasi-neutrality the potential drop can be calculated, according to [19], to be

$$e\Delta\Phi \approx -k_B T_e \ln\left(\sqrt{\frac{m_i}{2\pi m_e}}\right)$$

with the electron current density

$$j_e = -\frac{n_e}{4}\langle v_e \rangle \exp\left[-\frac{e\Delta\Phi}{k_B T_e}\right] .$$

This model neglects effects such as sputtering of the plasma wall and secondary electron emission (SEE) with coefficient γ. If the latter is considered the effective potential drop, according to [20], is modified to be

$$e\Delta\Phi^{eff} \approx -k_B T_e \ln\left((1-\gamma)\sqrt{\frac{m_i}{2\pi m_e}}\right) .$$

The potential drop is thus reduced, leading to a larger number of electrons in the sheath, resulting in lower electron energies within the plasma. For high SEE coefficients $\gamma \approx 1$, which can be found in ion thrusters, the classical Debye sheath transforms into a double layer structure with a potential well near the wall that can trap a fraction of the electrons. Spatial sheath oscillations with frequencies in the GHz range can occur, leading to sheath instabilities [21].

3.1.4 Collisional effects in the HEMP-T

In order to fully understand the physics of the HEMP thruster, the influence of collisional effects in the plasma has to be considered. In order to quantify the influence, the mean free path of an electron within the thruster is considered. If the mean free path is of the order of the thruster dimensions, the electrons will not have a Maxwellian velocity distribution, as the plasma is not dominated by particle collisions. The shortest mean free path is that of charge exchange collisions between ions and neutrals in the channel. An estimate of the mean free path is given by

$$\lambda_{mfp,CX} = \frac{v_n}{\langle \sigma_{CX} v_{ion} \rangle n_n} ,$$

which will be considered to quantify the collisional effects. Due to very strong coupling of ions and neutrals by charge exchange collisions both have similar temperatures in the channel, hence the velocities v_{ion} and v_n cancel each other. The neutral density in the channel is estimated to be about $5 \cdot 10^{19}\,\mathrm{m}^{-3}$ and the cross section for charge-exchange collisions can be calculated according to [22]. Therefore, the estimated mean free path is

$$\lambda_{mfp,CX} = \frac{1}{1.57 \cdot 10^{-14}\,\mathrm{cm}^2 \cdot 5 \cdot 10^{13}\,\mathrm{cm}^{-3}} \approx 1.2\,\mathrm{cm} . \tag{3.15}$$

As this is the smallest collisional length scale in the thruster channel, one can see that the channel plasma cannot be Maxwellian, as this length scale is still larger than the channel radius of 0.9 cm and the channel plasma is not collision dominated. The mean free paths for the electrons in particular are larger, resulting in kinetic effects dominating throughout the thruster channel and even reaching into the plume. This non-Maxwellian behavior is further supported by fig. 3.3, showing the electron energy distribution at different axial positions. In this logarithmic plot, a Maxwellian energy distribution function would correspond to a straight line, but for all positions on the thruster axis the electron distribution differs from such. It can be better described by a disturbed Maxwellian distribution [23].

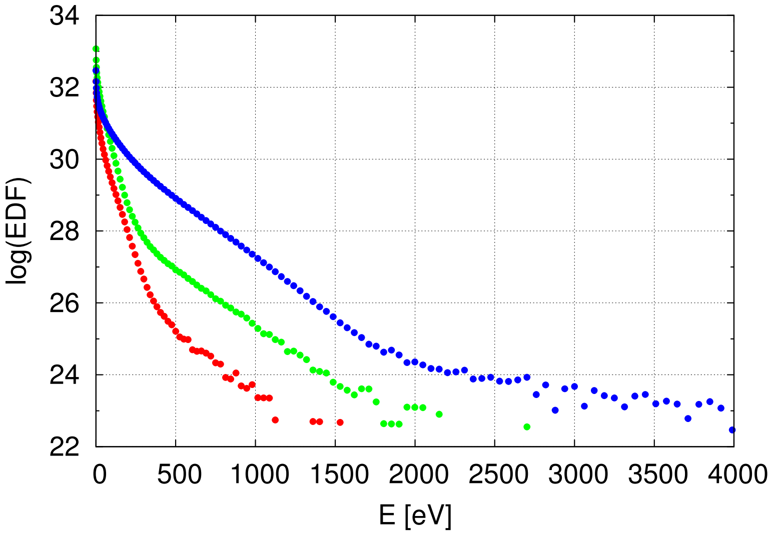


Figure 3.3: Simulation results for the electron energy distribution at 10 mm (red), 35 mm (green) and $r = 51$ mm (blue) [7]

As shown above, these non-Maxwellian effects extend into the plume due to the long mean free paths in the system. Hence, a fluid model for the electrons in the plume would only be an approximation. But as the focus in plume simulations is on the ions, as these generate the thrust and are the cause of sputtering damages. With only small particle densities and collision frequencies in the plume, the fluid electron model can still be suitable to describe the plume physics.

3.2 The Particle-in-Cell scheme

One method to simulate a plasma is the Particle-in-Cell scheme. It was first introduced in the late 1950s by Bunemann and Dawson [24] and has developed into an often applied method not only in plasma physics but in other fields as well. This approach has proven to be better applicable than direct solution of the kinetic equation eq. (3.1), where difficulties arise due to the high dimensionality and coupling of the equations.

In the PIC cycle, a grid is introduced, dividing the simulation region into cells. This enables treatment of large systems by calculating the elctrostatic potential only on the grid instead of N^2 direct particle interactions. Collisional effects are only taken into account within each cell seperately.

The PIC cycle starts at a given time t_0 by initializing the system and calculating the macroscopic quantities, such as the particle density, on the grid points using the particle positions and velocities. The forces acting on the particles are calculated on the grid and then reassigned to each particle, resulting in a change of the particle's position and velocity. After calculating further particle interactions, i.e. particle collisions and surface interactions, the system is advanced by a discrete timestep Δt and returns to the start of the cycle. To assure stability, the timestep has to be chosen small enough to resolve the fastest particle movement, so it has to be of the order of the electron plasma frequency

$$\omega_{P,e} = \sqrt{\frac{n_e q_e}{\varepsilon \varepsilon_0 m_e}} \,. \tag{3.16}$$

The grid spacing has to be of the order of the electron Debye length

$$\lambda_{D,e} = \sqrt{\frac{\varepsilon_0 k_B T_e}{n_e q_e}} \tag{3.17}$$

to resolve these scales.

In a plasma the force acting on a charged particle is the Lorentz force eq. (3.4). Along with Newton's third law the equations of motion for the charged particles

$$\begin{aligned} \frac{\mathrm{d}\vec{x}_i}{\mathrm{d}t} &= \vec{v}_i \\ \frac{\mathrm{d}\vec{v}_i}{\mathrm{d}t} &= \frac{1}{m_i}\vec{F}(\vec{x}_i, \vec{v}_i) = \frac{q_i}{m_i}\left(\vec{E}(\vec{x}_i) + \vec{v}_i \times \vec{B}(\vec{x}_i)\right) \,, \end{aligned} \tag{3.18}$$

with $i = 1...N$ as particle index, are obtained. In plasma physics the number of particles in a system is very large ($\gtrsim 10^{12}$) so that superparticles consisting of many physical particles are introduced. Due to the fact that the mass-to-charge ratio is constant for superparticles and real particles it is not necessary to rescale the laws describing the plasma [25]. Up to 10^8 computational particles may be simulated on modern computers within reasonable computation time [26].

The electric field $\vec{E}(\vec{x})$ and and the elctrostatic potential $\Phi\left(\vec{x}\right)$ are calculated by first solving Poisson's eq. (3.3) on the grid and then solving eq. (3.2), using the particle density $n(\vec{x})$ on the grid. The electrostatic PIC cycle consists of the following steps:

i) Calculate $n(\vec{x})$ on the grid by weighting the particle positions onto the grid points using a given shape function.

ii) Integrate the field equation eq. (3.3) and calculate $\vec{E}(\vec{x})$ on the grid via eq. (3.2).

iii) Calculate the force on each particle with the same weighting function as before.

iv) Integrate the equations of motion eq. (3.18) and move each particle.

v) Check boundaries for boundary effects, particle loss or others.

vi) Calculate particle collisions.

This cycle is visualized in fig. 3.4. The collisions may be neglected in some approximations resulting in the collisionless PIC scheme.

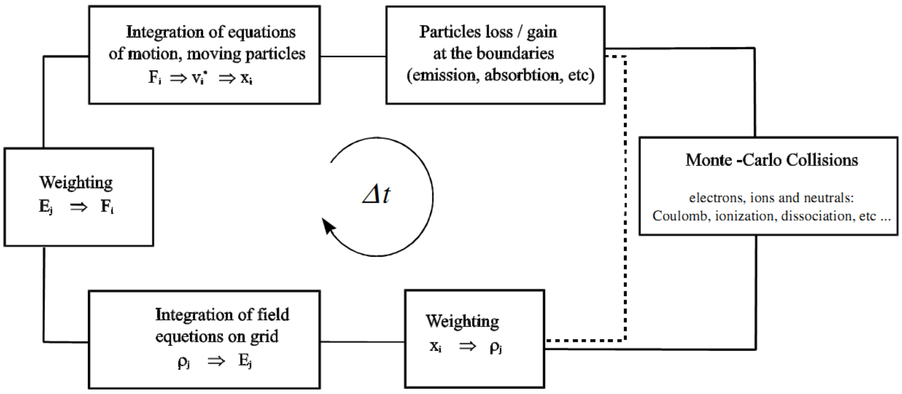


Figure 3.4: Basic steps of the PIC-cycle

3.2.1 Particle weighting and shape function

In PIC the macroscopic quantities calculated on the grid depend on the particle charge densities that have to be known on the grid as well. A shape function $S(\vec{x} - \vec{x}_i)$ is applied to each particle giving it a spatial distribution. The same function has to be chosen for each particle, containing at least one grid point.

Furthermore, it must be greater or equal to zero, has to satisfy charge conservation and has to be isotropic in space or numerical artifacts can occur [5]. The density is then calculated on a grid point $\vec{x}_j$ as

$$n(\vec{x}_j) = \sum_{i=1}^{N} q_i S(\vec{x}_j - \vec{x}_i)\,. \tag{3.19}$$

For a linear weighting function this is called a Cloud-in-Cell (CIC) approach and the weighting process is known as mapping. After the forces $\vec{F}(\vec{x}_j) = \vec{F}_j$ are calculated on the grid points $\vec{x}_j$ with $j = 1, ..., N_g$, they are mapped back onto the particles using the same shape function to avoid numerical artifacts. Thus, the force acting on the ith particle is

$$\vec{F}_i = \sum_{j=1}^{N_g} \vec{F}_j S(\vec{x}_j - \vec{x}_i)\,. \tag{3.20}$$

The use of the same shape function for both mapping and backmapping ensures momentum conservation and avoidance of self forces on equidistant grids.

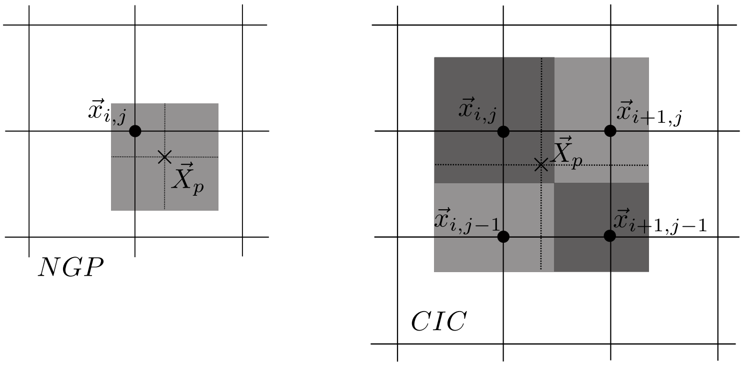


Figure 3.5: Visualization of the NGP and CIC shape functions

Within the requirements stated above the shape function can be chosen arbitrarily but two factors should be considered. The choice of shape function can be the cause of numerical noise. The nearest grid point (NGP) approximation, mapping the entire particle onto the closest grid point, is proven to cause substantially more noise than other shape functions [27]. A better choice is the CIC approach, where the particle shape covers at least two grid points. Secondly, the shape function should be chosen so that the mapping eq. (3.19) and backmapping eq. (3.20) will be computed efficiently. Some choices of shape functions are given in fig. 3.5.

The clouds around each particle do not interfere with each other, thus modifying the short range coulomb force neighbored particles experience.

3.2.2 Field solve phase

Finite difference scheme

To calculate the electric field on the grid, Poisson's eq. (3.3), an elliptical partial differential equation has to be solved. The solutions will be acquired by introducing a finite difference scheme for the spatial second order derivatives. For a two dimensional $M \times N$ cartesian grid (x_i, y_j) with constant permittivity ε and the charge density $n = n_i - n_e$ eq. (3.3) takes the form

$$\begin{aligned}\Delta\Phi_{i,j} &= \left(\frac{\partial^2\Phi_{i,j}}{\partial x_i^2} + \frac{\partial^2\Phi_{i,j}}{\partial y_j^2}\right)\\ &= \left(\frac{\Phi_{i-1,j} - 2\Phi_{i,j} + \Phi_{i+1,j}}{\Delta x^2} + \frac{\Phi_{i,j-1} - 2\Phi_{i,j} + \Phi_{i,j+1}}{\Delta y^2}\right)\\ &= -\frac{n}{\varepsilon\varepsilon_0}\,, \qquad (3.21)\end{aligned}$$

creating a system of linear equations to be solved instead. If the grid spacing is equal in both dimensions, i.e. $\Delta x = \Delta y = \Delta$, and the array $\Phi_{i,j}$ is aligned as a $(M \cdot N)-$dimensional vector, the above scheme can simply be written as

$$\frac{1}{\Delta^2}\mathrm{A}\Phi = -\frac{n}{\varepsilon\varepsilon_0}\,. \qquad (3.22)$$

The used five-point finite difference scheme is accurate of the order of Δ^2, which may be increased by incorporating more points into the difference scheme, i.e. using a nine-point scheme, which increases computation time by additional matrix entries [28].

The $(M \cdot N) \times (M \cdot N)-$dimensional matrix $\mathrm{A} = (a_{ij})$ has a characteristic block structure

$$\mathrm{A} = \begin{pmatrix} \mathrm{D} & \mathrm{I} & 0 & \cdots & 0 \\ \mathrm{I} & \mathrm{D} & \mathrm{I} & \ddots & \vdots \\ 0 & \mathrm{I} & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \mathrm{I} \\ 0 & \cdots & 0 & \mathrm{I} & \mathrm{D} \end{pmatrix}$$

$$\text{with} \qquad \mathrm{D} = \begin{pmatrix} -4 & 1 & 0 & \cdots & 0 \\ 1 & -4 & 1 & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & 1 & -4 \end{pmatrix} \tag{3.23}$$

The tridiagonal matrix D then has a dimension of $M \times M$ and I is the $M \times M$ unity matrix. Each line and column contains N such block matrices. This basic structure neglects the inclusion of boundary conditions or boundary points but is suitable to discuss the basic properties of A.

A is symmetric and regular, with all eigenvalues greater than zero, meaning it is positive definite. These properties do not depend on the choice of grid or geometry, they are also valid for non-uniform grids and varying coefficients ε [28]. A is a sparse matrix, as for each row a maximum of five elements are unequal to zero, making matrix operations less expensive and reducing the amount of RAM necessary to store the matrix.

The boundary conditions of eq. (3.3) are applied to the matrix as well. Constant (Dirichlet) boundary conditions leave $\Phi_{i,j}$ constant

$$\Phi_{i,j} = \Phi^0_{i,j} \tag{3.24}$$

while the diagonal element in the matrix A is a one, with no other entries in the same row and column. The value of the right-hand-side is given as $\Phi^0_{i,j}$. The other possible boundary condition fixes the flux (Neumann boundary conditions). This is achieved by fixing the difference in potential between to neighbored grid points. Often a no-flux condition is chosen which leads to the condition (applied at a boundary in y direction)

$$\Phi_{i,j} - \Phi_{i,j-1} = 0\,. \tag{3.25}$$

The matrix entry then becomes a one in the diagonal and a -1 below it with no other entries in the same row or column.

PIC codes for plasma simulations are rarely full three-dimensional codes but rather use a radial symmetry, reducing the problem to two spatial dimensions. Thus, cylindrical polar coordinates can be used. Also in a real plasma the permittivity is not constant but can change from one grid cell to another. After transformation, Poisson's equation in cylindrical coordinates takes the form

$$\frac{1}{r}\varepsilon\frac{\partial\Phi(z,r)}{\partial r}+\frac{\partial}{\partial r}\left(\varepsilon\frac{\partial\Phi(z,r)}{\partial r}\right)+\frac{\partial}{\partial z}\varepsilon\frac{\partial\Phi(z,r)}{\partial z}=-\frac{n}{\varepsilon_0}\,, \tag{3.26}$$

where the angular derivative was already dropped. If the finite difference scheme is applied on a two-dimensional grid with spacing $(\Delta r, \Delta z)$ at the point (r_i, z_j) (for $r_i \neq 0$) a modified form of eq. (3.21) can be derived [29] and gives

$$\begin{aligned}
&\left(\frac{1}{4r_i\Delta r}\left(\varepsilon_{i+1,j-1}+\varepsilon_{i+1,j+1}\right)+\frac{1}{2\Delta r^2}\left(\varepsilon_{i+1,j-1}+\varepsilon_{i+1,j+1}\right)\right)\Phi_{i+1,j}+\\
&\left(\frac{1}{4r_i\Delta r}\left(\varepsilon_{i-1,j-1}+\varepsilon_{i-1,j+1}-\varepsilon_{i+1,j-1}-\varepsilon_{i+1,j+1}\right)-\right.\\
&\qquad\qquad\left.\frac{1}{\Delta r^2}\left(\varepsilon_{i-1,j-1}+\varepsilon_{i-1,j+1}+\varepsilon_{i+1,j-1}+\varepsilon_{i+1,j+1}\right)\right)\Phi_{i,j}+\\
&\left(\frac{1}{2\Delta r^2}\left(\varepsilon_{i-1,j-1}+\varepsilon_{i-1,j+1}\right)-\frac{1}{4r_i\Delta r}\left(\varepsilon_{i-1,j-1}+\varepsilon_{i-1,j+1}\right)\right)\Phi_{i-1,j}+\\
&\frac{1}{2\Delta r^2}\left(\varepsilon_{i+1,j+1}+\varepsilon_{i-1,j+1}\right)\Phi_{i,j+1}+\frac{1}{2\Delta r^2}\left(\varepsilon_{i+1,j-1}+\varepsilon_{i-1,j-1}\right)\Phi_{i,j-1}\\
&=-\frac{n}{\varepsilon_0 V_{i,j}}\,.
\end{aligned}\tag{3.27}$$

And for the axis points the formula

$$\begin{aligned}
&\frac{1}{2\Delta r^2}\left(\varepsilon_{i+1,j-1}+\varepsilon_{i+1,j+1}\right)\Phi_{i+1,j}-\frac{3}{\Delta r^2}\left(\varepsilon_{i+1,j-1}+\varepsilon_{i+1,j+1}\right)\Phi_{i,j}+\\
&\frac{1}{\Delta r^2}\varepsilon_{i+1,j+1}\Phi_{i,j+1}+\frac{1}{\Delta r^2}\varepsilon_{i+1,j-1}\Phi_{i,j-1}
\end{aligned}\tag{3.28}$$

$$=-\frac{n}{\varepsilon_0 V_{i,j}}\,. \tag{3.29}$$

is retrieved. Although the new difference formula is more complicated it can still be written in the form of a matrix equation as in eq. (3.22), with rational coefficients a_{ij}.

LU decomposition

An often used method to solve systems of linear equations is the LU decomposition, also known as Gauss algorithm. Consider the system of linear equations

$$\mathrm{A}\Phi = b \tag{3.30}$$

with a given right hand side b and a regular $\mathrm{N} \times \mathrm{N}-$matrix A. For a PIC scheme, the matrix A is chosen to be the matrix representing the finite difference scheme eq. (3.27) with the right hand side b consisting of the charge density on the grid according to Poisson's eq. (3.3).

The system can be solved by representing the matrix A as the matrix product of an upper triangular matrix U and a lower triangular matrix L, transforming the equation to

$$\mathrm{A}\Phi = \mathrm{LU}\Phi = \mathrm{L}\tilde{\Phi} = b\,.$$

If L and U are known, the solution is easy. $\tilde{\Phi}$ is obtained by simply substituting the result of the previous lines into the next one as L has a triangular structure. The next step is to obtain Φ by solving for $\mathrm{U}\Phi = \tilde{\Phi}$ analogously. This step is also known as back-solve. For each back-solve, the complexity is $\sim N^2/2$ [30], making it very efficient.

The problem lies in the computation of the decomposition $\mathrm{A} = \mathrm{LU}$ which shall not be discussed in detail here, but a good review can be found in [30]. It can be shown that a LU decomposition exists for every regular matrix, but pivoting, interchanging rows and columns of the matrix in order to move the matrix elements with the highest absolute value to the diagonal, might be necessary, thus further increasing computation time. The complexity of the decomposition is $\sim N^3/3$.

The Gauss algorithm is a very robust direct method to solve matrix equations. With the exception of rounding errors, which can be minimized by partial or full pivoting, it reliably delivers the right solution. Difficulties arise in analyzing the stability and rounding errors of the algorithm. There is a chance that rounding errors are very large in certain cases [30].

In PIC the LU decomposition offers a reliable and efficient solver for the field solving step, as the matrix structure is well known and not problematic. The decomposition is calculated at the beginning of code execution, as the matrix does not change throughout the execution of the code, and only the back-solve has to be computed every PIC cycle, hence giving a complexity of $\sim N^2$ per PIC cycle. The parallelization is problematic, as each line within a back-solve step depends on the results of the previous lines, limiting its application to a serial code structure.

3.2.3 Successive over-relaxation

Another frequently used method to solve eq. (3.30) is the use of a stationary iterative procedure. To formally obtain such procedures, eq. (3.30) is rearranged using a regular matrix B. The $(k+1)-$th iterate is then calculated as

$$\begin{aligned} \mathrm{A}\Phi &= \mathrm{B}\Phi + (\mathrm{A}-\mathrm{B})\,\Phi = b \\ \mathrm{B}\Phi^{k+1} &+ (\mathrm{A}-\mathrm{B})\,\Phi^{k} = b \\ \Phi^{k+1} &= \Phi^{k} - \mathrm{B}^{-1}\left(\mathrm{A}\Phi^{k} - b\right) = F\left(\Phi^{k}\right) \,, \end{aligned} \tag{3.31}$$

The iterative procedure can be broken down to four steps:

i) Choose a starting point Φ^0.

ii) Calculate $\mathrm{A}\Phi^k$.

iii) Solve $\mathrm{B}\Delta\Phi^k = b - \mathrm{A}\Phi^k$.

iv) $\Phi^{k+1} = \Phi^k + \Delta\Phi^k$.

B is chosen to have a simple form in order to reduce the necessary number of operations and defines the iterative procedure. Also B is often linked to the matrix A. If B is chosen to be the diagonal of A, the algorithm is known as Jacobi algorithm. If B is chosen to be the sum of the A's diagonal matrix D (with $a_{ii} \neq 0$ for all i) and its lower triangular matrix L (not to be confused with the matrix used in the LU decomposition), the Gauss-Seidel algorithm, with the element index i, is acquired:

$$\begin{aligned} \mathrm{A} &= \mathrm{D} + \mathrm{L} + \mathrm{R} \\ \mathrm{B} &= \mathrm{D} + \mathrm{L} \\ \Phi^{k+1} &= -\left(\mathrm{D}+\mathrm{L}\right)^{-1}\left(\mathrm{R}\Phi^{k} - b\right) \\ \Phi_i^{k+1} &= \frac{1}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij}\Phi_j^{k+1} - \sum_{j>i} a_{ij}\Phi_j^{k}\right) \,. \end{aligned} \tag{3.32}$$

This method is convergent if A is symmetric and positive definite [28]. It can be enhanced by introducing a relaxation parameter ω into the choice of B

$$\mathrm{B}(\omega) = \frac{1}{\omega}\left(\mathrm{D} + \omega\mathrm{L}\right) \,. \tag{3.33}$$

The algorithm is altered, giving

$$\Phi^{k+1} = (1-\omega)\,\Phi^k + \omega\tilde{\Phi}^{k+1} = \Phi^k + \omega\left(\tilde{\Phi}^{k+1} - \Phi^k\right) \tag{3.34}$$

$$\tilde{\Phi}_i^{k+1} - \Phi_i = \frac{1}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij}\Phi_j^{k+1} - \sum_{j>i} a_{ij}\Phi_j^k - a_{ii}\Phi_i^k\right) \tag{3.35}$$

where $\tilde{\Phi}_i^{k+1}$ is calculated via eq. (3.32). If $\omega < 1$ this is called under-relaxation and can be used to dampen divergent solutions. For $\omega > 1$ the algorithm is known as successive over-relaxation (SOR) which is an often applied method to solve the finite difference scheme for Poisson's equation.

The iteration continues until a termination criterion is met. A possible choice is

$$\frac{\|\Phi^{k+1} - \Phi^k\|}{\|\Phi^{k+1}\|} < \delta \tag{3.36}$$

in a given vector norm $\|\cdot\|$. Because this criterion is critical for $\Phi^{k+1} \to 0$ the condition

$$\|\Phi^{k+1} - \Phi^k\|_{max} < \varepsilon \tag{3.37}$$

may be used as well. The maximum norm is chosen to minimize the necessary computational cost.

For the solution to converge, as the Gauss-Seidel algorithm depends on the newly calculated iterates, the domain should be divided into small subdomains, each solved separately. A chess board pattern, solving first all even and then all uneven grid points, or vice versa, may also be used. The algorithm's structure allows for easy parallelization as the calculation of each point's iterate depends on only the surrounding points, delivering an advantage over LU decomposition.

It can be shown [28] that the SOR method is only convergent for $\omega \in (0,2)$ and that the optimal relaxation parameter can be found in the interval $\omega_{opt} \in (1,2)$. ω_{opt} can only be calculated for a uniform grid spaced by Δ, as found in [28], but a decent guess is provided by the approximation

$$\omega_{opt} \approx 2 - \Delta\,. \tag{3.38}$$

As ω_{opt} is found in only a narrow range as can be seen in fig. 3.6, the algorithm has to be fitted to the grid that is used.

The complexity of each iteration step is $\sim N^2$ and the expected number of iteration steps is $\sim N$, giving the entire SOR method a complexity of $\sim N^3$ [28]. This is a lot more computation time compared to the back-solve of the LU decomposition which scaled quadratically, but it has much better potential for parallelization in a PIC code than the LU back-solve, because its parallelization fits nicely into the domain decomposition used also for the parallelization of the pusher and collisions.

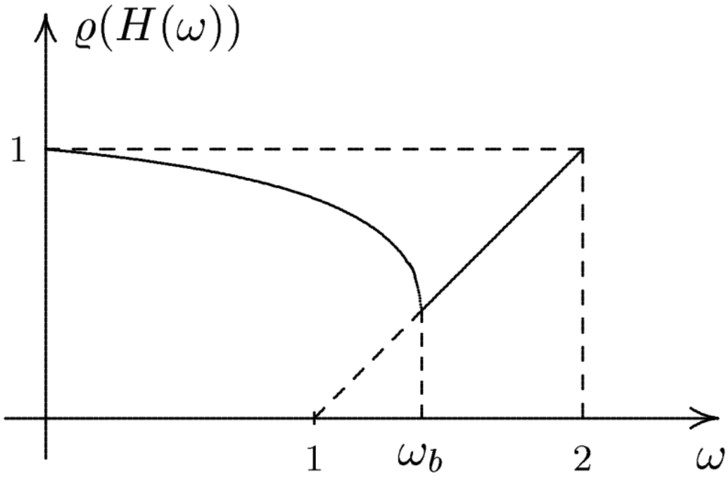


Figure 3.6: ω dependence of the error of the SOR algorithm for a fixed number of iterations. The optimal parameter is found in a narrow range.

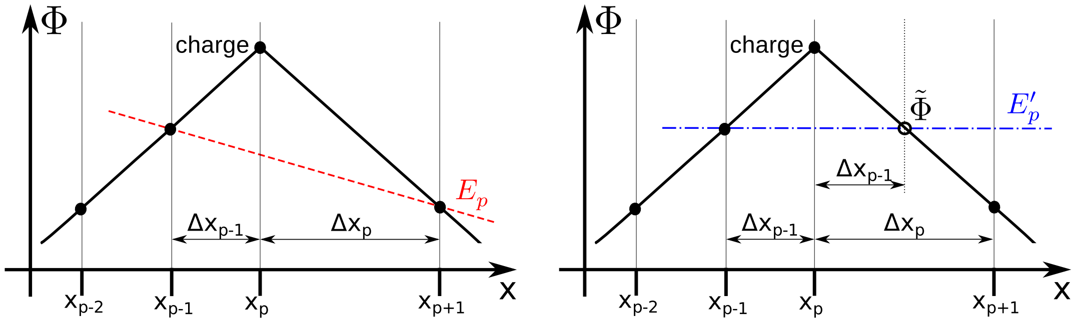
Calculation of the electric field

If the electrostatic potential $\Phi_{i,j}$ is obtained, the electric field is calculated by applying a simple symmetric difference scheme to eq. (3.2) for each component

$$\begin{aligned} E^r_{i,j} &= \frac{\Phi_{i-1,j} - \Phi_{i+1,j}}{2\Delta r} \\ E^z_{i,j} &= \frac{\Phi_{i,j-1} - \Phi_{i,j+1}}{2\Delta z}\,, \end{aligned} \tag{3.39}$$

if the grid is uniform. As the magnetic field is assumed to be externally defined in electrostatic PIC, the Lorentz force on the grid is then calculated via eq. (3.18) and then mapped back to the particles via eq. (3.20). This scheme assures that no self-forces are generated, assuring energy and momentum conservation [31].

If non-equidistant meshes are used, the scheme has to be adapted accordingly. The situation is sketched in fig. 3.7 and will be discussed for the general coordinate x_p with $\Delta x_p \neq \Delta x_{p+1}$, as the adjustment has the same form for each coordinate. If the central difference scheme in eq. (3.39) is used, the electric field will be calculated at the intermediate point of the interval $[x_{p-1}, x_{p+1}]$, which in this case does not coincide with the grid point x_p. This leads to a loss of energy



(a) General calculation scheme

(b) Modified calculation scheme

Figure 3.7: Electric field calculation for a charged particle located at x_i, in a non-equidistant grid with a change of cell size $\Delta x_{p-1} \neq \Delta x_p$. (a): Calculation scheme for an equidistant grid. (b): Modified field calculation.

and momentum conservation. Hence, an interpolation is used to calculate Φ at the point $x_p + \Delta x_p$

$$\Phi\left(x_p + \Delta x_p\right) = \Phi\left(x_p\right)\left(1 - \frac{\Delta x_{p-1}}{\Delta x_p}\right) + \Phi\left(x_{p+1}\right)\left(1 - \frac{\Delta x_p}{\Delta x_{p+1}}\right) ,$$

which would correspond to the value of Φ on the next grid point on a uniform mesh. The central difference scheme in eq. (3.39) can then be used, ensuring energy and momentum conservation [32].

3.2.4 Integration of equations of motion and particle mover

After obtaining the electric field and the force acting on each particle, the equations of motion eq. (3.18) have to be integrated for each particle. Since the number of simulated particles is very large, the main requirement for the integrator is low computational cost to go along with sufficient accuracy.

A scheme satisfying these requirements is the Boris-leapfrog algorithm first introduced in [33]. For each time derivative in eq. (3.18) a finite difference approximation is used with the same time step Δt as in the PIC code. As the second equation has to be evaluated at each complete timestep $t_k = t_0 + k\Delta t$ (because the positions and force acting on the particles are only known at each full timestep t_k), a central difference scheme for the time derivative of the velocities is introduced, thus evaluating the velocities have to be calculated at the half-steps $t_{k-1/2}$ and $t_{k+1/2}$. This leads to the following finite difference equations

(dropping the particle index for simplicity)

$$\begin{aligned}\frac{\vec{x}_{k+1}-\vec{x}_k}{\Delta t} &= \vec{v}_{k+1/2} \\ \frac{\vec{v}_{k+1/2}-\vec{v}_{k-1/2}}{\Delta t} &= \frac{q}{m}\left(\vec{E}_k(\vec{x}_k)+\frac{\vec{v}_{k+1/2}+\vec{v}_{k-1/2}}{2}\times\vec{B}_k(\vec{x}_k)\right)\end{aligned} \tag{3.40}$$

where the velocity at time k is approximated by the mean $\vec{v}_k = (\vec{v}_{k+1/2}+\vec{v}_{k-1/2})/2$. The accuracy of the leapfrog scheme is of order Δt^2 as is shown in [33].

To solve this set of equations an often used algorithm is the Boris algorithm giving the new particle positions and velocities as

$$\begin{aligned}\vec{x}_{k+1} &= \vec{x}_k + \Delta t \vec{v}_{k+1/2} \\ \vec{v}_{k+1/2} &= \vec{u}_+ + q\vec{E}_k(\vec{x}_k)\end{aligned} \tag{3.41}$$

using the definitions

$$\begin{aligned}\vec{u}_+ &= \vec{u}_- + \left(\vec{u}_- + \left(\vec{u}_- \times \vec{h}\right)\right) \times \vec{s} \\ \vec{u}_- &= \vec{v}_{k-1/2} + q\vec{E}_k(\vec{x}_k) \\ \vec{h} &= q\vec{B}_k(\vec{x}_k) \\ \vec{s} &= \frac{2\vec{h}}{1+h^2}\,,\end{aligned}$$

which requires a total of 39 operations.

The Boris-leapfrog method is an explicit scheme, as it requires only values from the previous timestep to calculate the new positions and velocities. While being numerically robust it restricts the timestep chosen for the PIC code. It can be shown [26] that the condition

$$\omega_0 \Delta t \lesssim 1 \tag{3.42}$$

must be met, with ω_0 being the highest frequency of any of the system's particle movements. In plasma simulations it is the electron plasma frequency that has to be resolved. To guarantee stability and minimize the systematic error a more strict condition

$$\omega_{p,e}\Delta t = 0.2$$

is used in the PIC code, assuring stability for at least 10^7 timesteps [26].

3.2.5 Particle collisions

So far the ansatz of calculating the macro-fields on the grid leads to the simulation of a collisionless plasma. In order to obtain a more realistic plasma description it is necessary to account for particle collisions. As there is a large number of possible collisions in a plasma, only the most important collisions can be modeled. These include Coulomb collisions between charged particles as well as elastic and inelastic collisions with neutral particles, including ionization and excitation.

It is still inconceivable to calculate the N^2 interactions between all charged particles on today's computers. Due to the Debye shielding of the charged particles in a plasma, each particle's Coulomb interaction is limited to an area of the size of the Debye sphere around this particle. For that reason only particles within the same cell are collided. For these a binary collision model can be used, as first suggested in [34]. Each particle in a cell is assigned a randomly chosen collision partner from the same cell to collide with. This is done by statistically scattering the relative velocity, rotating it(and hence assuring energy and momentum conservation) using a scattering angle χ and an azimuthal angle φ. While the azimuthal angle is chosen randomly in $[0, 2\pi]$, the scattering angle is sampled from a Gaussian distribution if the Fokker-Planck collision operator for binary collisions is used. This approach holds for sufficiently small collision frequencies and if the majority consists of small angle collisions [26].

The elastic and inelastic scattering collisions with neutral particles of density n_n can be calculated using an estimate for the collision probability as

$$P_c(\vec{v}) = 1 - \exp\left[-u n_n \sigma(E) \Delta t_c\right] ,$$

with the collision cross section $\sigma(\vec{E})$, relative velocity u and the collision time step Δt_c. The term $u n_n \sigma(E)$ presents the collision frequency for collisions with neutral particles, thus the exponential term is the probablility that a particle passes the timestep t_c without a collision.

To perform a collision, a uniform random number $u \in [0, 1]$ is chosen and compared to P_c. If $u \leq P_c$, a collision is performed, resulting in a rotation of the relative velocity vector for elastic collisions. When an ionizing or excitation collision occurs, the barycenter energy is reduced by the threshold energy E_{thr}. In the case of electron impact ionization, an electron-ion pair is created with position and velocity of the former neutral particle, and a Coulomb collision of the ionizing electron with the created particles is performed. An in depth discussion of Monte-Carlo collision models for the PIC method can be found in [27].

3.3 Simulation of the HEMP-T, code description

The first simulations of the HEMP-T were performed by K. Matyash et. al. [35] and more recent results can be found in [7]. A 2D PIC scheme with radial symmetry and a grid spacing of $\Delta x = \Delta r = 0.5\lambda_{D,e}$ on 890×240 grid points was used, with 3D simulation of particle velocities. The timestep was chosen to be $\Delta t = 0.2/\omega_{P,e} = 1.2 \cdot 10^{-12}\,\mathrm{s}$, as given in eq. (3.42), with about 10^6 timesteps necessary to reach a steady state.

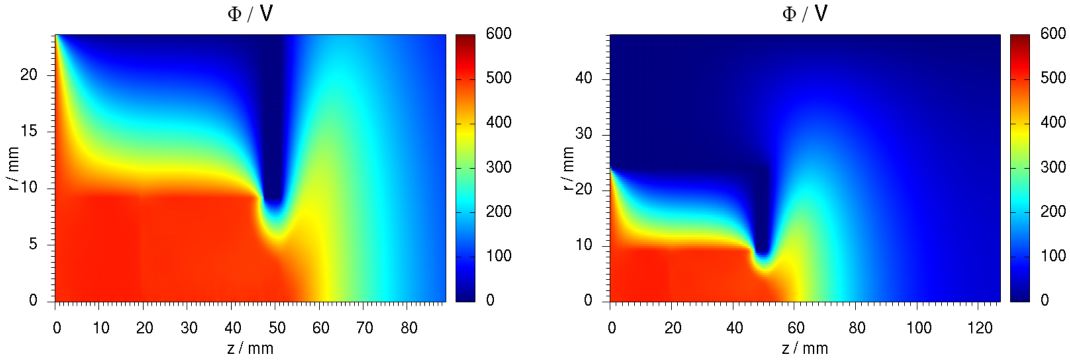
In this work, a 2D grid is used, covering the dimensions of the HEMP-DM3a with a mesh of 1272×480 grid points in the (z, r) plane. To reduce computational costs, similarity scaling as described in [5] is used, reducing the system size but keeping the physical laws intact as the mass-to-charge ration of each species is unchanged, unlike models that artificially raise electron mass. The timestep is chosen to be $\Delta t = 0.2/\omega_{P,e} = 1.2 \cdot 10^{-12}\,\mathrm{s}$ as in the previous works simulations conducted in [35] and [7]. A non-uniform mesh, further explained in [6], is applied to the simulation region. The ions are moved once per $400\Delta t$ and neutrals are moved once per $2000\,\Delta t$.

A multigrid method incorporating two nested grids, as described in [7], is used for the calculation of the electrostatic potential Φ. A coarse grid covers the entire domain, with a larger grid spacing of $\Delta x_{coarse} = 4\Delta x_{fine}$, while the finer grid only covers the thruster region with a mesh of 888×236 grid points. During the field solve phase, a solution for Φ is first obtained on the coarse grid, with the boundary conditions of the finer grid given by the interpolated values on the coarse grid. Then a solution is obtained for the finer grid. The anode voltage is set to 500 V with a zero potential boundary condition at the upper and a no flux condition at the right boundary. Poisson's equation is solved with the Gauss algorithm included in the Super_LU library [36], calculating the LU decomposition once, only using the backsolve during each timestep.

In fig. 3.8 the potential on both grids is plotted. It can be anticipated that the zero potential boundary conditions on the upper and the no-flux condition on the right boundary differ from the real situation, thus deviating the simulated potential, distorting simulation results. Thus, a large simulation region for the plume is desirable, but increases computational cost, which can be reduced by applying a hybrid electron model for plume simulation.

The thruster is fueled with Xenon gas, with electrons and Xe^+–ions included as charged particles. The injection of the neutral Xenon gas takes place in the bottom right corner of the simulation domain and Maxwellian electrons are injected at the position of the neutralizer source in the exit plume. These injections are performed once per $500\,\Delta t$.

The PIC code includes elastic neutral-neutral, electron-neutral and ion-neutral collisions as well as electron impact ionization collisions. The electron-electron



(a) Fine Grid

(b) Coarse Grid

Figure 3.8: Electrostatic potential of a converged run on both grids of the multigrid method.

and electron-ion Coulomb collisions are included also. The collisions are treated as described in section 3.2.5. The neutral-neutral collisions are performed once per each neutral timestep while all other collisions are performed during each ion timestep.

The first step to achieve a hybrid description of the plume is to implement and test a new Poisson solver with the SOR method, as this type of solver has beneficial properties, such as simple implementation and straight-forward parallelization, for fluid particle models, as described in [37]. Secondly, the electrons have to be simulated as a fluid, thus the fluid equation eq. (3.11) has to be solved for the electrons. The electrons will be assumed as massless, so that the Boltzmann relation eq. (3.13) can be considered instead. This leads to a non-linear Poisson's equation eq. (3.3), so that an iterative field solver becomes necessary. The fluid model will first be applied to the entire thruster and then restricted to the plume only.

4 Development of a Poisson solver

In this chapter, a new solver for the finite difference scheme of Poisson's eq. (3.22) is developed, since the LU decomposition of the SuperLU package [36] is not suitable for at parallelization of the PIC code. The method of choice is the SOR algorithm presented in section 3.2.3 as its parallelization is trivial and the exchange between separate domains calculated by different cores is very small. Furthermore, the solver is numerically robust with well known convergence properties, as discussed in [28]. The coordinate representation of the algorithm is given in eq. (3.35) and the calculation scheme for the coefficients of the finite difference scheme is found in eq. (3.27).

Within the PIC code the SOR method will be implemented as an alternative option to the SuperLU algorithm. The iteration procedure is executed until the termination condition

$$\|\Phi^{k+1} - \Phi^{k}\|_{max} < \varepsilon$$

is met for two subsequent iterates Φ^{k+1} and Φ^{k} in dimensionless form. As the domain covers large areas with $\Phi = 0\,\mathrm{V}$, using a relative termination condition as in eq. (3.36) is inappropriate. As the SOR method requires an initial guess at the start of the iteration, SuperLU is executed once at the start-up of the code, and the solution will be stored as the initial Φ during the first iteration. Alternatively, the SOR algorithm can also be used to obtain the initial guess, but usually costs more computational time than the SuperLU method, when no parallelization is used. The solution of each following iteration is then stored and used as guess during the next field solve. This method might lead to rounding errors being added up, so that stability tests over a number of PIC cycles are required.

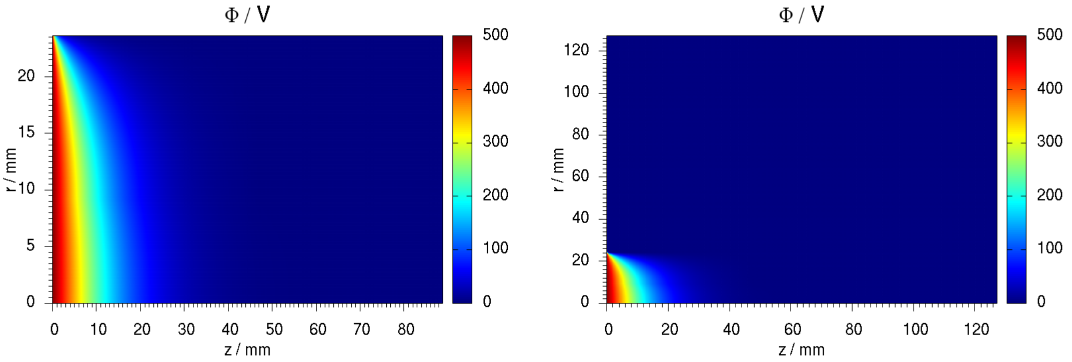
Tests of the SOR solver are needed to study the stability and computational cost of the method as well as the dependencies of solver parameters such as ε and ω. The tests are done for two cases, both being run on the simulation domain of the HEMP-T with the according boundary conditions described in section 3.3.

One test case is the execution within a converged, restarted run of the code, with charged particles covering the thruster channel and the exhaust region. This simulated HEMP-T is in a steady state after 14 516 000 timesteps were simulated. The (z, r)-domain covers an area of 1272×480 grid points, with a coarse grid of

338 × 120 and a fine grid of 888 × 236 grid points.

The second test case is a vacuum potential solution without plasma in a start-up run on a larger grid of 1272 × 1272 grid points. The size of the fine grid stays unchanged compared to the converged run, but the size of the coarse grid increases to 338 × 338 grid points, with the same choice of boundary conditions as before. This setup is chosen so that the solver's scaling to a larger grid and reaction to perturbations may be studied during execution of several PIC cycles.

For all benchmark tests, the potential solution of the SuperLU package will be considered as correct and all SOR solutions will be compared to it. The initial guess is $\Phi = 905\,\mathrm{V}$ on the entire domain. To ensure comparability, as not all benchmarks were run on the same machine, the execution time will be given in the number of iteration steps required to reach convergence. Firstly, a search for the best relaxation parameter ω_{opt} is conducted. Secondly, the influence of the termination condition is investigated, as the condition used does not directly correspond to the actual deviation from the correct solution. The third test will cover the dependence of execution time and the termination condition used. From these results suitable parameters for the iteration will be deduced and used in the PIC code. The test as a solver during the execution of the PIC code will be the last test conducted, in order to study the stability of the solver and its reactions to perturbations.



(a) Fine Grid (b) Coarse Grid

Figure 4.1: Electrostatic potential at the start-up of the PIC code for the larger grid.

4.1 Optimal relaxation parameter

The first test conducted is the determination of the best relaxation parameter for both grids, as the choice of ω not only influences the rate of convergence but also the correlation between the termination condition and the actual iteration error. An initial guess of $\Phi = 905\,\mathrm{V}$ and a termination condition of $\varepsilon = 10^{-6}$ were used. This value of ε was estimated to guarantee convergence, with estimated maximum errors of $1\,\mathrm{mV}$. This estimate is based on the fact that the scaling factor from the unscaled values to Volts amounts to 905.114 and that potential differences of $\sim 1\,\mathrm{mV}$ can be neglected as they have no effect on the electric field calculation. The number of iteration steps required to reach that convergence rate was recorded for each value of ω that was used. For the restarted runs, the results are presented in fig. 4.2.

As stated in section 3.2.3, the ideal parameter is to be found in the range of $\omega \in [1, 2)$, with a guess that $\omega \approx 2 - \Delta x$ as in eq. (3.38). As the grid contains of the order of 10^3 grid points in each direction, a rough estimate of $\omega_{opt} \approx 1.9$ can be made. Since the ideal parameter depends on the grid spacing, a different value has to be obtained for both grids used in the multigrid approach.

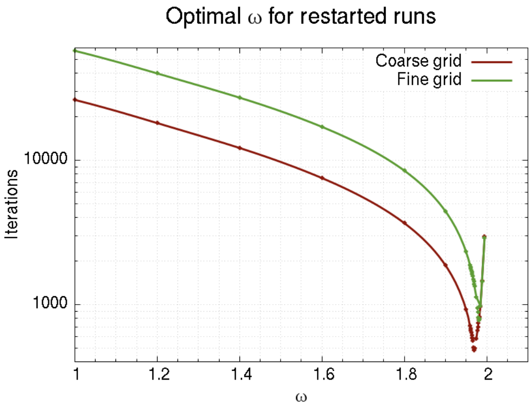


Figure 4.2: Number of Iteration steps required to reach a convergence of $\varepsilon < 10^{-6}$ for restarted runs.

In fig. 4.2 the results for the restarted runs are shown. The dependence of ω from the required number of iteration steps matches the expected behavior as shown in fig. 3.6 on each grid. The optimum is found in a very narrow range with parameters of $\omega_{c,r} = 1.969$ for the coarse and $\omega_{f,r} = 1.981$ for the fine grid. The ideal choice decreases the required number of iterations by a factor of $\sim 10^2$ on each grid compared to the Gauss-Seidel method with $\omega = 1$.

The situation is similar for the vacuum tests as shown in fig. 4.3. The best parameter for the fine grid is the same as for the restarted runs, $\omega_{f,s} = \omega_{f,r} =$

1.981. This is due to the fact that ω depends only on the chosen grid, not on the right hand side of the matrix equation eq. (3.22). For the coarse grid, a value of $\omega_{f,s} = 1.986$ is obtained. The number of iterations decrease by approximately the same factor as before.

The figures show that the search for the best parameters is crucial, as a reduction of ω by 0.02 from the ideal value approximately doubles the required number of iterations to reach convergence. If ω is chosen even closer to two, the computational expense dramatically increases. It can be observed that ω increases for an increased number of grid points, making the search more difficult and erratic, as it has to be obtained to a very high order of accuracy as it gets closer to two. It is recommended to include an automatic search for ω when implementing such a method.

Despite the optimization, the SOR-method is a lot slower than the SuperLU solving routine. On the same hardware and with the combination ideal relaxation parameters for each grid, the iteration procedure had a runtime of 29.38 s for the restarted runs and 44.41 s for the start-up runs, respectively. The full SuperLU method needed a time of 7.6 s for restarted runs and 12.7 s for start-up runs, being faster by a factor of about 4 in both cases. This increase is not unexpected, as the complexity of the SOR solver scales as $\sim N^3$, with the overall number of grid points N, and the Gauss algorithm scales as $\sim N^3/3$, which in the symmetric case can be improved to $N^3/6$ [30]. The optimal relaxation parameters for each grid will be used from this point forward.

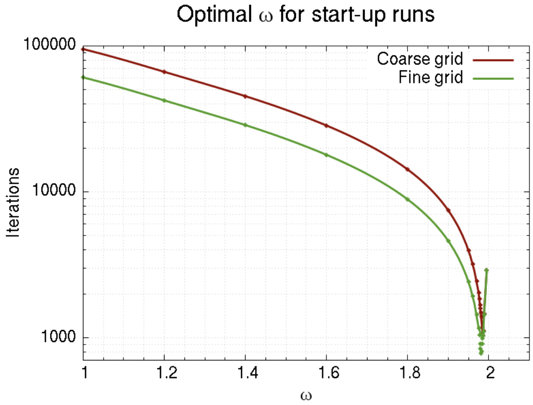


Figure 4.3: Number of Iteration steps required to reach a convergence of $\varepsilon < 10^{-6}$ for start-up runs.

4.2 Iteration error and termination condition

The next step is to investigate the correlation between the chosen termination condition ε in eq. (3.37) and the actual deviation from the SuperLU solution. Again, an initial value of $\Phi = 905\,\mathrm{V}$ was used. The solution is considered to be convergent if the maximum deviation from the SuperLU solution is below $10^{-2}\,\mathrm{V}$. As this is the maximum difference, this value is estimated to have a negligible effect on the resulting electric field, not influencing charged particle movement.

The situation for the restarted runs is plotted in fig. 4.4. The double logarithmic plot shows that the actual calculation error scales almost linearly with the chosen termination condition, but is of about three orders of magnitude higher, which can be attributed to the scaling factor of $905.114\,\mathrm{V}$. Therefore the plot shows the expected dependence, as the maximum deviation in dimensionless units is of the order of the termination condition. The linear curve flattens for conditions $\varepsilon < 10^{-8}$.

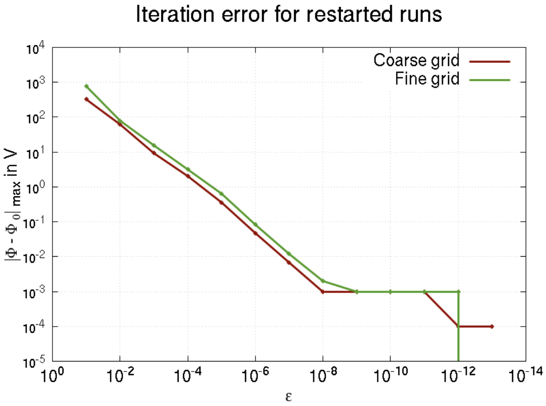
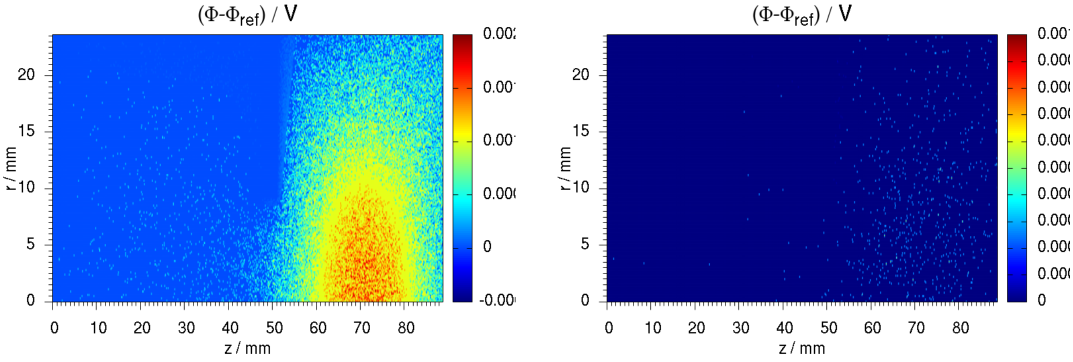


Figure 4.4: Different values for the termination condition ε and corresponding maximum deviation from the real solution for restarted run.

This behavior does not indicate directly that the accuracy of the solution is not increasing for more restrictive ε. fig. 4.5 shows that the area with the largest error is "permeated" by the more accurate solution. This trend continues for more accurate solutions until only isolated points with a high error remain. That is the reason for the error to hit zero on the fine grid in a very steep drop, pictured in fig. 4.4. For restarts, no results for $\varepsilon = 10^{-14}$ or lower could be acquired due to persistent rounding errors.

The situation is similar for the start-up runs. The error scales almost linearly with the chosen termination condition, with higher accuracy achieved for $\varepsilon \geq 10^{-6}$ when compared to the restarted case. This can be attributed to the potential vanishing on most parts of the grid.



(a) $\varepsilon = 10^{-8}$ (b) $\varepsilon = 10^{-10}$

Figure 4.5: Absolute difference in electrostatic potential of the SuperLU and SOR for the given termination conditions on a restarted run.

It can be concluded that termination conditions of $\varepsilon \leq 10^{-6}$ ensure good accuracy of the potential calculation with an error $\leq 10^{-3}$ V, leaving the physics of the system unchanged. Values of $\varepsilon \geq 10^{-5}$ should not be considered due to deviations of ≥ 0.1 V.

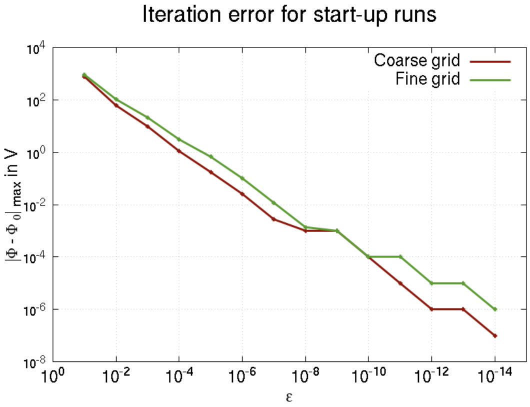


Figure 4.6: Different values for the termination condition ε and corresponding maximum deviation from the real solution for start-up runs.

4.3 Solver runtime

The last benchmark is the solver runtime, i.e. number of iteration steps, as a function of the termination condition and therefore the error ε, with the initial value being $\Phi = 905\,\mathrm{V}$ again. It is expected that the grids containing a large number of grid points require more iterations to reach convergence and that the computation time increases linearly with each higher order of accuracy. This is due to the complexity $\sim N^3$ of the SOR algorithm, with about N iterations to reach a converged solution without earlier termination. As this number should not be surpassed, a linear dependency for each higher order of magnitude is expected.

The dependency for the restarted runs is presented in fig. 4.7. The calculation time shows the expected dependency, with a linear rise of the number of iterations with each higher order of accuracy. The number of iterations required is higher on the fine grid than on the coarse one, which was also expected.

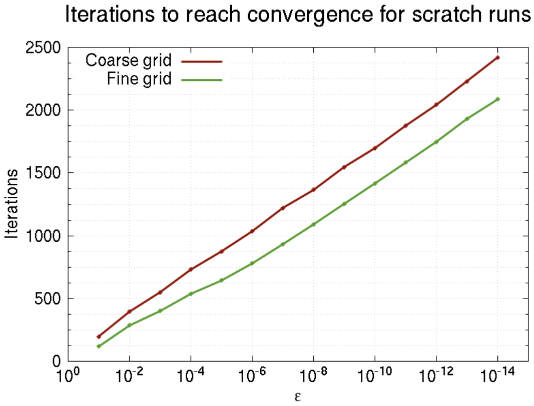


Figure 4.7: Number of iterations required to reach the termination condition ε for restarted runs.

The situation is similar for the start-up runs in fig. 4.8, as the calculation time scales linearly as well. Here, the number of iterations required is higher for the coarse grid, which can be attributed to the larger domain covered by the grid and the high initial value chosen for Φ, as it takes a higher number of iterations for the boundary conditions to affect the inner grid points. It is observed that the areas with high potential gradients correspond to large errors in the iteration procedure, as apparent when comparing fig. 4.5 and fig. 3.8. The next section further discusses this observation.

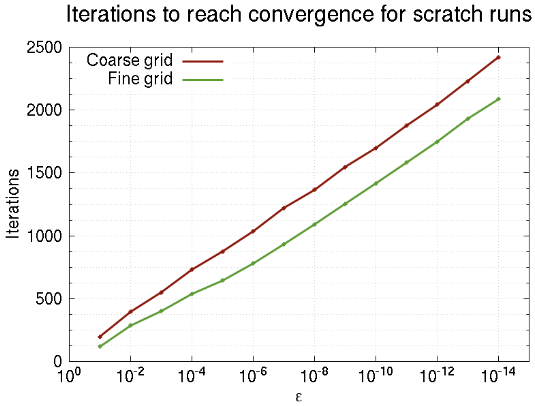


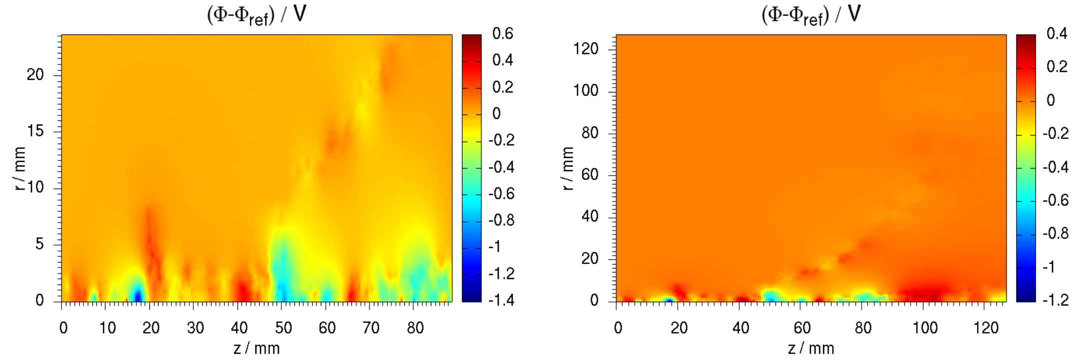
Figure 4.8: Number of iterations required to reach the termination condition ε for start-up runs.

From the observed calculation time and error dependence in section 4.2, one can conclude that termination conditions in the range of 10^{-10} to 10^{-6} combine good solver accuracy with an acceptable solver runtime. It has to be taken into account that this benchmark setup is not the one used in the actual PIC cycle, as the guess used there will be much closer to the actual solution than a condition of $\Phi = 905\,\mathrm{V}$ on the entire domain, such that the number of iterations is expected to decrease significantly within the real application.

4.4 Inclusion in the PIC code, stability test

In this section, the behavior of the SOR solver in the PIC code will be investigated, testing the stability and reaction to perturbations within the system. Here, a solver setup used in the PIC scheme is used is applied, with the first initial Φ provided by the SuperLU solver that is executed at the startup, and each solution of the SOR solver stored as the initial value for the following iteration.

The first test was conducted for the vacuum case. The duration of the test was 9100 PIC cycles, with the potential averaged over 100 steps with $\varepsilon = 10^{-8}$. It is expected that large differences in the solutions will be observed, due to the Monte-Carlo scheme used for the ionization collision routines. This test will serve as a measure of the stability of the SOR solver, along with a benchmark in calculation time. The absolute differences between the SOR and SuperLU solutions on both grids are shown in fig. 4.9.



(a) Fine grid

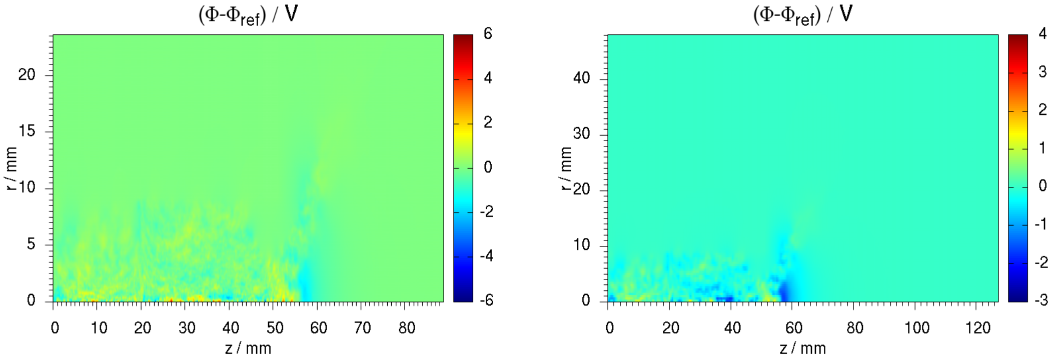
(b) Coarse grid

Figure 4.9: Difference in electrostatic potential of the SuperLU and SOR solver with a termination condition of $\varepsilon < 10^{-8}$ after 9100 PIC steps, averaged over 100 steps during a start-up run.

As expected large differences in potential within the channel and the near-field plume are observed due to ionization processes. Still, the SOR method shows stability toward sudden changes in potential, and solver errors are not expected to add up during a large number of PIC cycles. On comparable machines, the code with the SuperLU package executed this test in 93 min, compared to 3553 min with the SOR method, which is slower by a factor of about 40. This is expected, since only the SLU back-solve with a complexity of $\sim N^2$ has to be computed during each PIC cycle, with each PIC cycle executed in about 1 s. With the SOR method, each PIC cycle is executed in approximately 23 s.

This test was also performed for restart runs, with the resulting differences given in fig. 4.10 after the same number of time steps and with the same termination condition as in the start-up run in fig. 4.9. The differences between both solutions are observed almost exclusively in the thruster channel and the near field plume, the areas where the highest amount of ionization takes place. Thus, the deviations from the SuperLU run can be attributed to the statistics of the system, as a Monte-Carlo routine is used to perform collisions which take place mostly within the thruster channel. One can expect that the same solution is reached when averaging Φ over a large number of timesteps, reducing statistical effects observed.

In fig. 4.11 the results for a longer averaged restarted run are presented. The potential was averaged over $3 \cdot 10^4$ timesteps, with $\varepsilon = 10^{-6}$ in order to reduce the execution time of the code. As expected it is observed that the difference in potential is reduced, with no pattern of deviation. The deviation is further



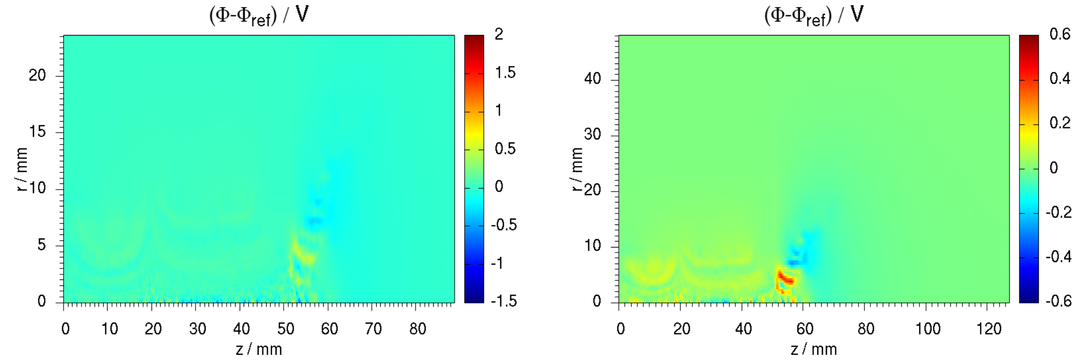
(a) Fine grid

(b) Coarse grid

Figure 4.10: Difference in electrostatic potential of the SuperLU and SOR solver with a termination condition of $\varepsilon < 10^{-8}$ after 9100 PIC steps, averaged over 100 steps during a restarted run.

expected to decrease for even longer time averages of the order of 10^5 to 10^6 timesteps, such that a statistical mean is then observed.

It can be concluded that the SOR method offers an alternative method to solve the finite difference scheme eq. (3.22) in the PIC method. Although its execution time is much larger than that of the SuperLU back-solve, it offers some advantages as well, as its parallelization is simple, with each core calculating its own domain, with only little information exchange between neighboring domains necessary. There are further methods to reduce the execution time of the SOR method, such as the Chebyshev acceleration as described in [28]. In this thesis the focus is on the application of the SOR method to the hybrid description of the electrons that is discussed in the next chapter.



(a) Fine grid (b) Coarse grid

Figure 4.11: Difference in electrostatic potential of the SuperLU and SOR solver with a termination condition of $\varepsilon < 10^{-6}$ averaged over 30000 PIC steps during a restarted run.

5 Hybrid plume model

In this chapter the development of a hybrid PIC model for the simulation of the HEMP-T is presented. Firstly, a theoretical introduction is given along with the numerical implementation in the PIC code. Secondly, this method is applied to the entire thruster and results are presented. In the last section the application of the method to the thruster plume is discussed.

5.1 Implementation of the hybrid model

Besides the fully kinetic PIC method, hybrid methods are often applied in plasma simulations. In plume simulations hybrid plasma models have been applied successfully [8]. The advantage of such a plume model is that no longer the electron movement has to be resolved, with the restrictions to length scales of $\sim 1.49 \cdot 10^{-3}\,\mathrm{m}$ and time scales of $\sim 1/564\,\mathrm{MHz}$, as given by the electron Debye length and the electron plasma frequency within the thruster channel, respectively. Only the ion movement has to be resolved, with the a Debye length in the range of $10^{-3}\,\mathrm{m}$ up to $1\,\mathrm{m}$, and a plasma frequency of $\sim 10\,\mathrm{kHz}$ within the plume. This allows plume simulations on length scales of several meters, in order to assess possible sputtering damages of surrounding parts caused by exhausted ions.

The condition to achieve such a description is that electrons are no longer treated fully kinetic, but rather as a fluid described by the fluid equation of motion eq. (3.11), the governing continuity equation eq. (3.12) and the equations for the electric field eq. (3.3).

In order to drop the electron time scale, the particles need to be treated as a massless fluid, following changes in the electric field instantaneously. This way, effects on the time scale of electron motion, such as electron excitations, are lost. The electron fluid is assumed to have a Maxwellian energy distribution in a local thermodynamic equilibrium. This way, the calculation of electron charge density n_e simplifies tremendously, as the Boltzmann relation

$$n_e\left(\Phi\right) = n_{e0} \exp\left[\frac{e\left(\Phi - \Phi_0\right)}{k_B T_e}\right] ,$$

with the electrostatic potential Φ, electron temperature T_e and the Boltzmann constant k_B, is used instead of solving the entire set of fluid equations eq. (3.11). Φ_0 and n_{e0} are reference values for the charge electrostatic potential and electron charge density, respectively.

As a consequence, Poisson's equation eq. (3.3) is converted from a linear partial differential equation to a nonlinear one, as the electron density is now coupled to the electrostatic potential Φ, resulting in the new equation

$$\Delta\Phi = \frac{n_{e0}}{\varepsilon\varepsilon_0}\exp\left[\frac{e\left(\Phi - \Phi_0\right)}{k_b T}\right] - \frac{n_i}{\varepsilon\varepsilon_0} \tag{5.1}$$

for the potential.

In the PIC scheme, the finite difference scheme, as discussed in eq. (3.21) and eq. (3.27), can still be applied to discretize the equation. The non-linearity in eq. (5.1) is then linearized and a solution can be obtained. As the electron density changes with the electrostatic potential, an iterative procedure needs to be used to reach a fixed point of the iteration. Hence, even direct solvers, such as the SuperLU algorithm used in the PIC code, have to be iterated to reach a solution of eq. (5.1).

The SOR method in eq. (3.34) allows for adjustment to incorporate the changed electron potential, which is linearized within the scheme given in eq. (3.35). Depending on wether $n_e\left(\Phi\right)$ is chosen as a function of the previous iterate Φ^k or the new one $\tilde{\Phi}^{k+1}$, two different schemes can be derived. In the first case, no explicit linearization is required, as the potential from the previous iteration is already known, resulting in an explicit scheme

$$\begin{aligned}\tilde{\Phi}_i^{k+1} - \Phi_i^k = \frac{1}{a_{ii}}\Bigg(& n_{i,i} - n_{e0,i}\exp\left[\frac{e\left(\Phi_i^k - \Phi_{0,i}\right)}{k_B T_{e,i}}\right] - \\ & \sum_{j<i} a_{ij}\Phi_j^{k+1} - \sum_{j>i} a_{ij}\Phi_j^k - a_{ii}\Phi_i^k\Bigg)\ . \end{aligned} \tag{5.2}$$

with the new iterate calculated $\Phi^{k+1} = \Phi^k + \omega\left(\tilde{\Phi}^{k+1} - \Phi^k\right)$ as derived in eq. (3.34).

In the second case the electron density eq. (3.13) is first linearized

$$n_e\left(\tilde{\Phi}^{k+1}\right) = n_{e0}\exp\left[\frac{e\left(\tilde{\Phi}^{k+1} - \Phi_0\right)}{k_B T_e}\right] \approx \left(1 + \left(\tilde{\Phi}^{k+1} - \Phi^k\right)\right)\exp\left[\frac{e\left(\Phi^k - \Phi_0\right)}{k_B T_e}\right]$$

and a semi-implicit scheme is derived by substitution into Poisson's eq. (5.1), resulting in the equation

$$\tilde{\Phi}_i^{k+1} - \Phi_i^k = \frac{1}{a_{ii} + \exp\left[\frac{e\left(\Phi_i^k - \Phi_{0,i}\right)}{k_B T_{e,i}}\right]} \left(n_{i,i} - n_{e0,i} \exp\left[\frac{e\left(\Phi_i^k - \Phi_{0,i}\right)}{k_b T_{e,i}}\right] - \right.$$
$$\left. \sum_{j<i} a_{ij} \Phi_j^{k+1} - \sum_{j>i} a_{ij} \Phi_j^k - a_{ii} \Phi_i^k \right) , \tag{5.3}$$

After the solution is obtained, the PIC method can be continued as in the fully kinetic case, without the need to move electrons.

One difficulty remaining are the collisions that still need to be performed, as electron impact ionization is the only source of electrons within the bulk plasma. In this work, the kinetic collision routines were retained within the hybrid model by a steady state approximation. As the plasma within the thruster channel is not expected to change dramatically from a converged run, the electron positions are retained and a new Maxwellian velocity distribution is sampled for the electrons within each cell. The collisions are then performed between these and the other particles, with the benefit of using the proven collision routines that guarantee a good description of the HEMP-T, as described in [35], with the disadvantage of only being able to use these routines in a steady state assumption. This assumption is expected to work well in plume modeling, since particle densities and collision probabilities are small within the plume.

5.2 Results

5.2.1 Application to the entire thruster

Firstly, the hybrid model described above is applied to the simulation of the entire HEMP-T. This approach is not expected to work very well, as in section 3.1.3 and section 3.1.4 it was stated that the plasma within the thruster channel is dominated by kinetic effects, such as electron heating in the magnetic cusps, and sheath effects due to the channel walls. Furthermore, the transport of magnetized electrons along the magnetic field lines close to the thruster axis further reduces the applicability of such a hybrid electron model for the entire thruster, as discussed in section 3.1.2. This case is still of interest, as the results can be compared to that of the fully kinetic PIC code, and may present some limitations of the hybrid electron model.

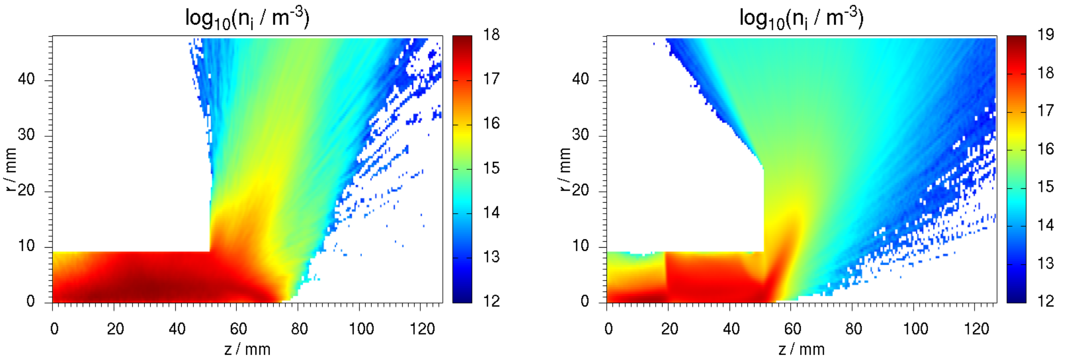
To test the model, the same converged run as in chapter 4 was used, starting at a time step of 14 516 000 electron timesteps on a mesh of 1272 × 480 grid points with the multigrid approach implemented as previously discussed. The

PIC scheme is mostly unchanged, with the difference that electron density is calculated within the solver and no further electron motion needs to be calculated. The collisions are performed using the steady state approximation described in the previous section.

At the beginning of code execution, the SuperLU solver is executed, providing the initial Φ_0 used in the SOR method, which is implemented as discussed in chapter 4. The initial density n_{e0} is obtained by calculation of the initial electron particle distribution given in the restart file, and connected to the initially calculated potential Φ_0, such that $n_e(\Phi_0) = n_{e0}$ in eq. (3.13). For the solution of the nonlinear eq. (5.1), the semi-implicit scheme eq. (5.3) is used, to avoid timestep restrictions as given by an explicit scheme. After each field solve, the electron temperature is calculated using the approximation

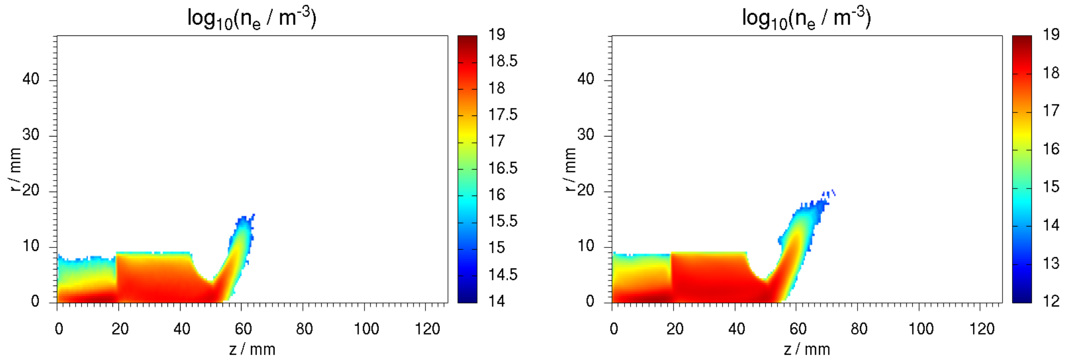
$$T_e = T_{ref}\left(\frac{n_e}{n_{ref}}\right)^{\alpha-1}$$

from eq. (3.14). The reference temperature was set to $T_{e0} = 10\,\mathrm{eV}$ and with the parameter $\alpha = 4/3$ which were suggested to be good estimates for plume modeling at the thruster exit [8]. The simulations were performed for 496 000 timesteps, with the ion pusher executed once every 400 timesteps, so that the simulation amounts to 1240 ion steps.



(a) Hybrid (b) Kinetic

Figure 5.1: Logarithmic plot of the ion density from the hybrid and the kinetic run after 496 000 timesteps, with values averaged over 100 000 timesteps. Here, the hybrid model is applied to the entire domain including the thruster channel.



(a) Hybrid

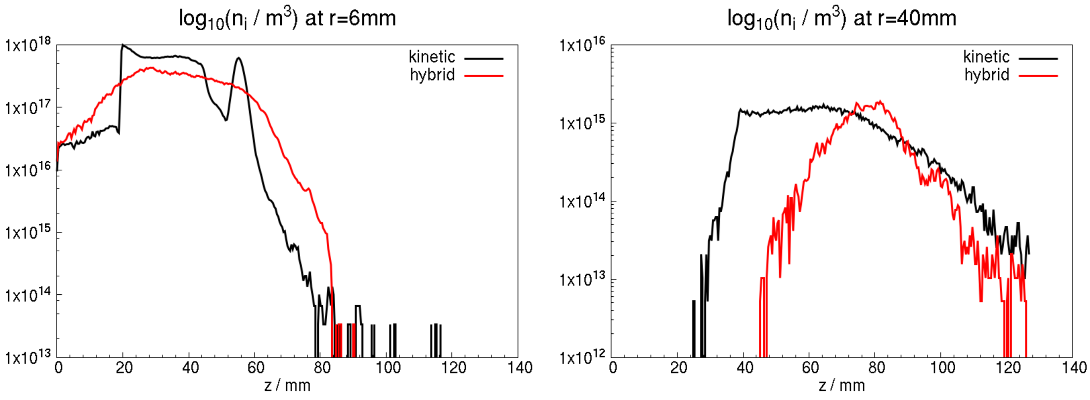
(b) Kinetic

Figure 5.2: Logarithmic plot of the electron density from the hybrid and kinetic runs after 496 000 timesteps, with values averaged over 100 000 timesteps. Here, the hybrid model is applied to the entire domain including the thruster channel.

In plume simulations, the ion motion is of interest since the ions generate the thrust and lead to sputtering of surrounding parts. The resulting ion density after application of the hybrid electron model, as well as the kinetic run are found in fig. 5.1. Furthermore, the axial ion distribution for $r = 6\,\mathrm{mm}$ within the channel and $r = 40\,\mathrm{mm}$ in the plume are given in fig. 5.3. Within the thruster channel the ion density of the hybrid run is higher near the wall and slightly lower near the axis when compared to the kinetic run. The ion movement resembles a diffusion within the channel when compared to the kinetic run, thus resulting in a higher ion density near the thruster exit. Figure 5.3 (a) further supports this, as the kinetic ion density (black) is higher within the thruster and lower in the exit region than the hybrid ion density (red).

In the plume, the ion density is much more focused, with only a little fraction of the ions having an negative axial velocity component, but also with lower density in the region near the right boundary. Thus is shown in fig. 5.3 (b), as the hybrid ion density only surpasses the kinetic density near $z = 80\,\mathrm{mm}$.

In order to understand this behaviour, the electrostatic potential has to be considered, which is given in fig. 5.4, as well as its deviation from the kinetic run. The potential difference within the thruster is reduced just slightly by a maximum of around 10 V. The largest differences are found in the thruster exit region with an increase of up to 50 V, where the ion density of the hybrid run was very large, thus resulting in the observed deviation. Within the plume, this results in the reduced ion density for small beam angles, as the potential gradient



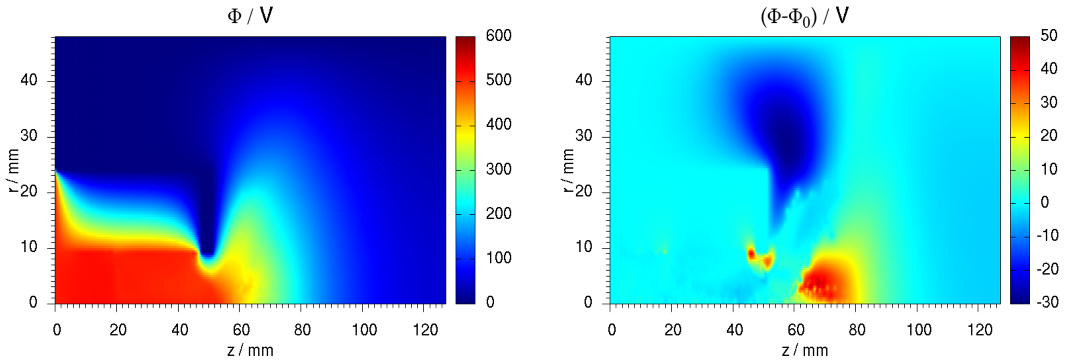
(a) $r = 6\,\mathrm{mm}$ (b) $r = 40\,\mathrm{mm}$

Figure 5.3: Axial distribution of the ion density from the hybrid and the kinetic run at different radial positions after 496 000 timesteps, with values averaged over 100 000 timesteps. Here, the hybrid model is applied to the entire domain including the thruster channel.

is changed and ions are accelerated towards larger beam angles. The reduced electrostatic potential above the thruster leads to a smaller potential gradient in negative $z-$direction, resulting in only a very small number of ions being accelerated into that direction.

The electron density of the hybrid and kinetic runs are shown in fig. 5.2. One can see that the electron density slightly decreased throughout the channel, as the ion density and therefore the electrostatic potential showed a slight diminishing in that area as well, as comparison with fig. 5.4 and fig. 5.1 shows. In the thruster exit region the limitations of the used hybrid model are visible. Here, the electrostatic potential increased the most. As stated above the electron density was initialized with the starting density, thus not allowing no real electron dynamics in areas where $n_{e,0} = 0$. An attempt was made to introduce such an electron dynamics, but resulting in diverging solutions.

The application of the hybrid electron model to the entire thruster develops into a different solution compared to the fully kinetic PIC case and does not describe the thruster very well, as the ion density and the potential differ from that of the kinetic runs. This resembles the expectations stated in the fluid description of plasmas section 3.1.2, since the fluid description is best suited to describe collision dominated plasmas, with negligible kinetic effects and only small influence of the plasma sheath. But as the electron dynamics in the HEMP-T is dominated by kinetic effects, the hybrid model does not offer a suitable description of this type of ion thruster. In order to obtain better results, the



(a) Potential

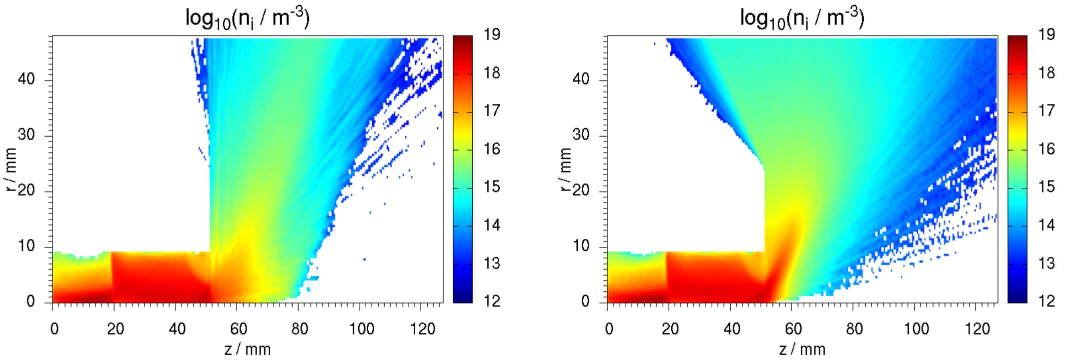
(b) Difference to the kinetic run

Figure 5.4: Plot of the electrostatic potential, and the difference to the kinetic run after 496 000 timesteps, with values averaged over 100 000 timesteps. Here, the hybrid model is applied to the entire domain including the thruster channel.

hybrid electron model was then restricted to the plume region.

5.2.2 Application to the plume

In this section, the plume simulation using the hybrid electron model is discussed. To restrict the hybrid model to this area, a boundary surrounding the thruster is introduced. Within this boundary, the ions are neither pushed nor collided and the system there remains in the initial state. Outside the boundary, the code is executed as described in section 5.2.1 and the electrostatic potential is solved on the entire domain. In order to include new ions into the system, the average number of ions, as well as their spatial and velocity distribution in the boundary cells were measured using a kinetic simulation of 500 000 timesteps. This information is used to sample a number of ions missing from the average number during each ion timestep at the boundary via a Metropolis-Hastings algorithm [38]. Two boundaries were introduced, one at the thruster exit and one behind the electron source. In order to assess the results, a comparison with a Monte-Carlo plume model is discussed, sampling the ions at the boundary and pushing them in a straight line motion toward the domain boundaries without the influence of an electric field. As the change in electron density is rather low within the plume region, as shown in fig. 5.2, the discussion of the results is concentrating on the ions.

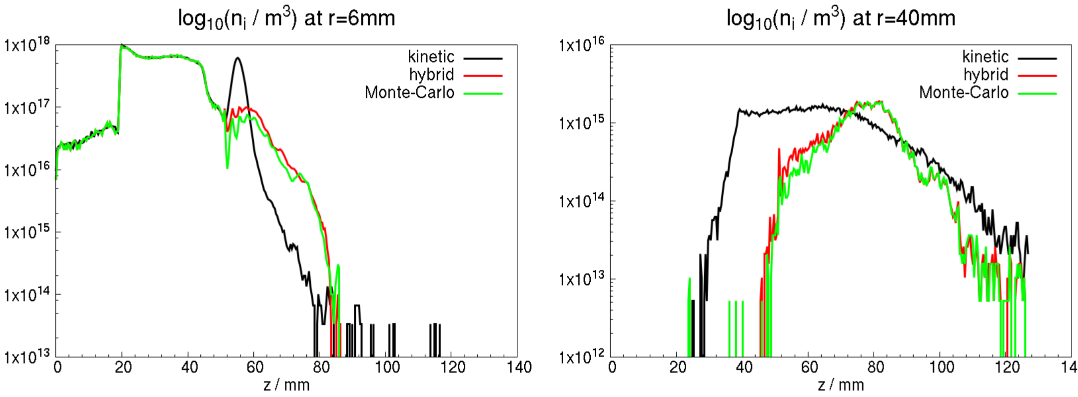


(a) Hybrid

(b) Kinetic

Figure 5.5: Logarithmic plot of the ion density from the hybrid and the kinetic run after 496 000 timesteps, with values averaged over 100 000 timesteps. The boundary is set at the thruster exit.

The case with the boundary located near the thruster exit is discussed first. The ion density is similar to the one included in fig. 5.1, as the hybrid ion density is much more focused within the plume region when compared to the kinetic results. Figure 5.6 (a) further supports this observation. Up to the thruster exit at $r = 51\,\mathrm{mm}$ the densities are identical as no ions inside the boundary are pushed. At the thruster exit, the kinetic ion density is larger at first, but then reduced when compared to the Monte-Carlo model as well as the hybrid model, which can be attributed to the increased electrostatic potential in that area as observed in fig. 5.7, resulting in an ion acceleration towards higher beam angles.



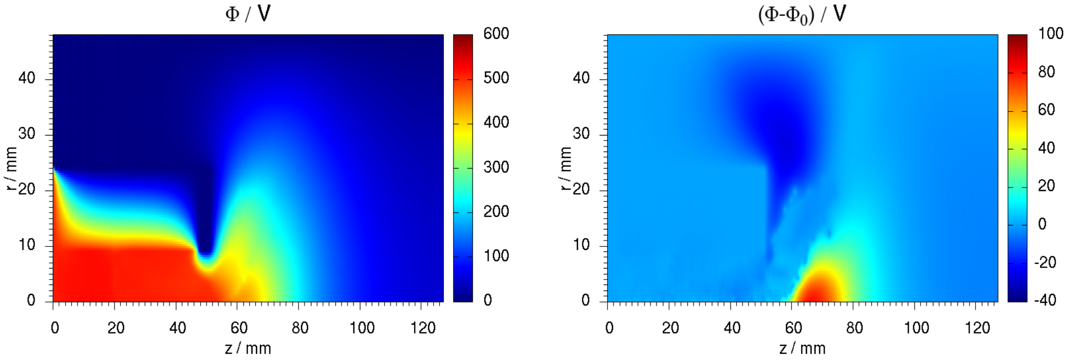
(a) $r = 6\,\mathrm{mm}$ (b) $r = 40\,\mathrm{mm}$

Figure 5.6: Axial distribution of the ion density from the hybrid, kinetic and a Monte-Carlo run at different radial positions after 496 000 timesteps, with values averaged over 100 000 timesteps. The boundary is set at the thruster exit.

The reduction in ion density for high beam angles can be explained by the decreased potential gradient in negative $z-$direction above the thruster, overall leading to a more focused ion beam, as shown in fig. 5.6. Both figures show furthermore, that the results of the hybrid model do not differ significantly from the Monte-Carlo model, as the difference in axial density distribution is negligible for both radial positions.

The second boundary introduced was set behind the electron source at 78 mm. The ion density of the hybrid run shows much better resemblance of the density with the kinetic run, as given in fig. 5.8. The ion density is slightly higher for low beam angles, as shown in fig. 5.9(a), where the ion density drops much faster after the boundary introduced at $z = 78\,\mathrm{mm}$. In that region the hybrid model results in lower ion density than the Monte-Carlo model, which can be attributed to further acceleration of ions in the exit region when using the hybrid model. The higher ion density in that area results in an increased potential as well, as seen in fig. 5.10, where a potential increase of up to 20 V occurs.

In the outer plume region the hybrid ion density is slightly smaller for low z, but surpasses the density of the kinetic run in regions closer to the right boundary



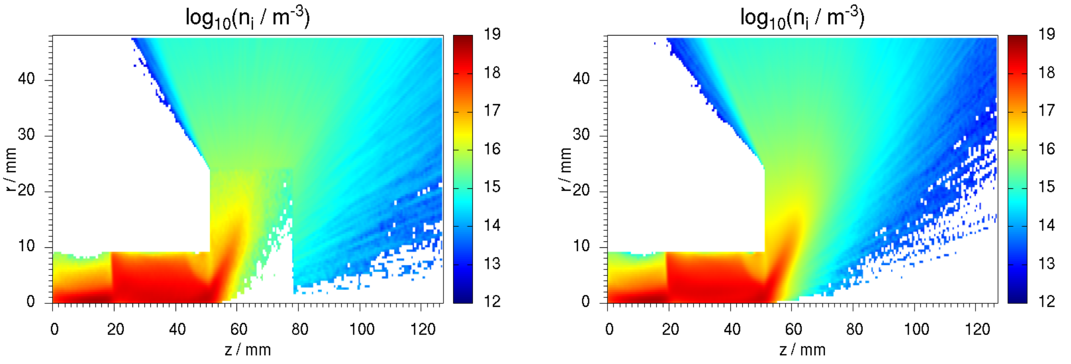
(a) Potential

(b) Difference

Figure 5.7: Plot of the electrostatic potential, and the difference to the kinetic run after 496 000 timesteps, with values averaged over 100 000 timesteps. The boundary is set at the thruster exit.

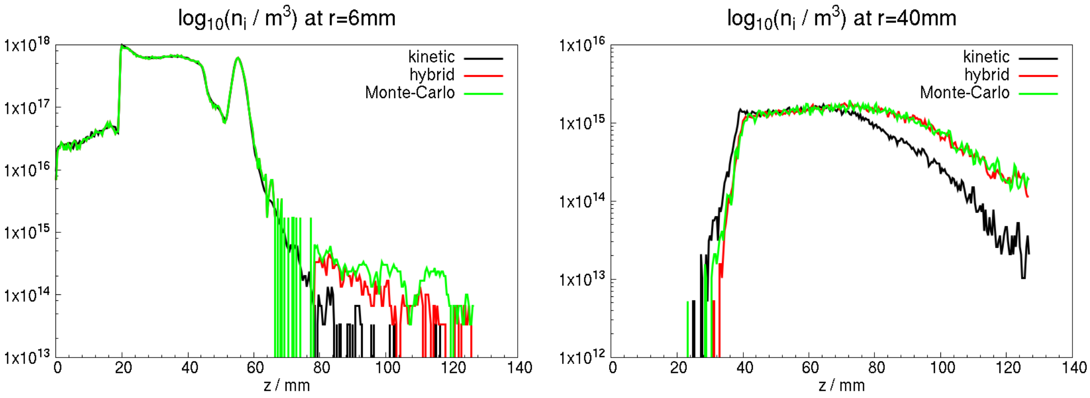
as fig. 5.9 (b) shows. From the graph one can also observe that the hybrid and the Monte-Carlo model get the same results with negligible differences between both methods. From these simulations one can conclude that the hybrid electron model as introduced in section 5.1 does not represent the kinetic results of the simulation of HEMP-T very well. In the application of the entire thruster, the model resulted in a diffused ion density, which then directly influenced the plume. This behavior continued to a lesser extend in a plume simulation starting at the thruster exit, where a more focused beam could be observed as well. Only a simulation with the boundary set behind the electron source could reproduce the kinetic results. But as physical effects are small in that area, a Monte-Carlo plume model offers nearly the same results.

Thus, the only advantage of the hybrid model over a Monte-Carlo approach for the plume may be the collection of additional information such as the ion energy distribution, which is strongly influenced by the electrostatic potential that is not considered within the Monte-Carlo model. The advantage of the hybrid plume model over the kinetic model is determined by the execution time of the code, as the simulation of 496 000 timesteps on comparable machines was finished in about 68 h for the fully kinetic code, but only needed close to 11 h for each hybrid run, where the execution time is dominated by the field solve with the adjusted SOR method in eq. (5.3). The Monte-Carlo method was executed in slightly under 12 min, or 0.2 h. Hence, the Monte-Carlo method is preferred over the hybrid model that was applied in this work.



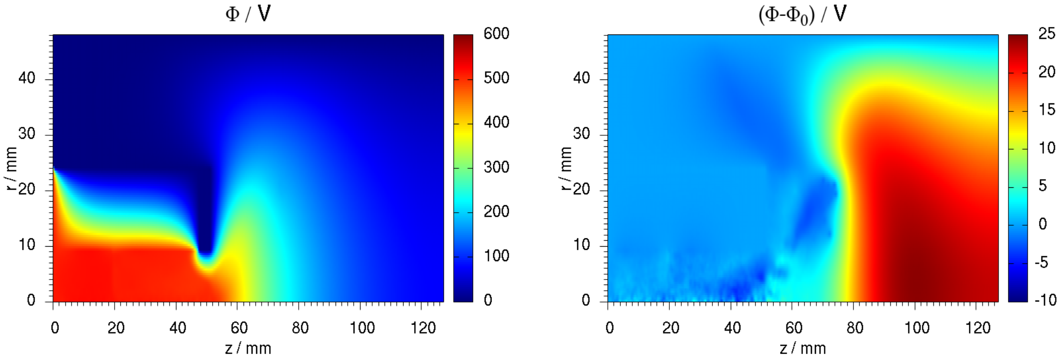
(a) Hybrid (b) Kinetic

Figure 5.8: Logarithmic plot of the ion density from the hybrid and the kinetic run after 496 000 timesteps, with values averaged over 100 000 timesteps. The boundary is set behind the electron source.



(a) $r = 6\,\mathrm{mm}$ (b) $r = 40\,\mathrm{mm}$

Figure 5.9: Axial distribution of the ion density from the hybrid, kinetic and a Monte-Carlo run at different radial positions after 496 000 timesteps, with values averaged over 100 000 timesteps. The boundary is set behind the electron source.



(a) Potential (b) Difference

Figure 5.10: Plot of the electrostatic potential, and the difference to the kinetic run after 496 000 timesteps, with values averaged over 100 000 timesteps. The boundary is set behind the electron source.

6 Conclusion

Aim of this work was the development and evaluation of a hybrid plume model for ion thruster modeling. This was achieved using the assumption of massless electrons, thus obtaining the electron density through the Boltzmann relation and a heuristic adiabatic approximation for temperature calculation, resulting in a simplified fluid electron model. For the numerical development of the model a new solver for Poisson's equation was implemented, which is also of interest for the fully kinetic PIC method due to its beneficial properties concerning a parallelization of the PIC code.

The results showed that the model was not suitable to be applied to the entire thruster, as the simulation results were vastly different from the kinetic model. The results were similar when the hybrid model was only incorporated for simulation of the plume, with two different boundaries introduced defining the simulation domain. For a simulation domain close to the thruster, the hybrid model showed a behavior similar to the hybrid model for the entire thruster. If the simulation domain starts outside of the electron source the results improved, but could also be obtained using a simplified ray-tracing Monte-Carlo method with almost no visible difference in results. Furthermore, the Monte-Carlo method is executed much faster than the hybrid model and hence should be preferred.

Hybrid models are very popular in ion thruster modeling because they need much less runtime than fully kinetic models. In the case of high-density collision dominated plasmas they offer a good approximation and deliver reliable results even when applied to the development and optimization of ion thrusters. This work showed that, in the case of HEMP-T, which is dominated by kinetic effects and non-Maxwellian distribution functions, the limits of the hybrid model become evident, as it delivers the same results as a simple Monte-Carlo model treating the ions unaffected by electric and magnetic fields in a ray-tracing limit. Therefore, a critical evaluation of the popular hybrid model should be done, because in case of plasmas with strong kinetic effects it can fail miserably.

Bibliography

[1] Dan A. Goebel and Ira Katz. *Fundamentals of Electric Propulsion: Ion and Hall Thrusters* . JPL SPACE SCIENCE AND TECHNOLOGY SERIES. NASA, 2008.

[2] esa. ESA's gravity mission GOCE. Retreved from `http://www.esa.int/Our_Activities/Observing_the_Earth/GOCE` (Jan. 2015).

[3] Marc Rayman. Deep Space 1. Retreved from `http://nmp.jpl.nasa.gov/ds1` (Jan. 2015).

[4] Joe Wise. Dawn - A Journey to the Beginning of the Solar System. Retreved from `http://dawn.jpl.nasa.gov` (Jan. 2015).

[5] F. Taccogna, S. Longo, M. Capitelli, and R. Schneider. Self-similarity in Hall plasma discharges: Applications to particle models. *Phys. Plasmas*, 12:053502, 2005.

[6] Karl Felix Lüskow, Julia Duras, Oleksander Kalentev, Konstantin Matyash, David Tskhakaya, Jürgen Geiser, and Ralf Schneider. Non-equidistant Particle-In-Cell for Ion Thruster Plumes. In *Proceedings of the 33rd International Electric Propulsion Conference*, October 2013.

[7] Karl Felix Lüskow. Physics of Ion Thrusters' Plumes . Master's thesis, University of Greifswald, July 2013.

[8] A. Vicini, A. Passaro, and L. Biagioni. Hall Thruster 3D Plume Modeling and Comparison with SMART-1 Flight Data, 2004.

[9] George P. Sutton and Oscar Biblarz. *Rocket Propulsion Elements.* John Wiley & Sons, Inc., 7th edition, 2001.

[10] Martin J.L. Turner. *Rocket and Spacecraft Propulsion.* Springer-Verlag, 2 edition, 2006.

[11] L.B. King. *Transport-property and mass spectral measurements in the plasma exhaust plume of a Hall-effect space propulsion system.* PhD thesis, University of Michigan, 1998.

[12] Francesco Taccogna. *Plasma-Surface Interaction inside a Hall Thruster.* PhD thesis, University of Bari, 2003.

[13] Norbert Koch, Martin Schirra, Stefan Weis, Alexey Lazurenko, Benjamin van Reijen, Jens Haderspeck, Angelo Genovese, Peter Holtmann, Ralf Schneider, Konstantin Matyash, and Oleksandr Kalentev. The HEMPT Concept - A Survey on Theoretical Considerations and Experimental Evidences . In *Proceedings of the 32nd International Electric Propulsion Conference*, number IEPC-2011-236, September 2011.

[14] Erfahrungsaustausch Oberflächentechnologie mit Plasma- und Ionenstrahlprozessen. *The HEMP Thruster - An Alternative to Conventional Ion Sources?*, volume X., march 2003.

[15] G. Kornfeld, Norbert Koch, and Hans-Peter Harmann. Physics and Evolution of HEMP-Thrusters . In *Proceedings of the 30th International Electric Propulsion Conference*, number IEPC-2007-108, 2007.

[16] F. F. Chen. *Introduction to Plasma Physics and Controlled Fusion, Volume 1: Plasma Physics*, volume 1. Springer Verlag, 1984.

[17] D. Kremp, M. Schlanges, and W-D. Kraeft. *Quantum statistics of nonideal plasmas.* Springer Verlag, 2005.

[18] Julia Duras. Instabilities in Ion Thrusters by plasma-wall interaction. Master's thesis, Brandenburg University of Technology, August 2011.

[19] Ralf Schneider. Plasma edge physics for tokamaks. 2001.

[20] G D Hobbs and J A Wesson. Heat flow through a Langmuir sheath in the presence of electron emission. *Plasma Physics*, 9(1):85, 1967.

[21] Francesco Taccogna, Savino Longo, Mario Capitelli, and Ralf Schneider. Anomalous transport induced by sheath instability in Hall effect thrusters. *Applied Physics Letters*, 94(25), JUN 22 2009.

[22] Arthur V. Phelps, April 2002. Retreved from `http://jila.colorado.edu/~avp/collision_data/ionneutral/IONATOM.TXT` (Jan. 2015).

[23] Oleksander Kalentev, Konstantin Matyash, Julia Duras, Karl Felix Lüskow, Ralf Erwin Schneider, Norbert Koch, and Martin Schirra. Electrostatic Ion Thrusters - Towards Predictive Modeling . *Contributions to Plasma Physics*, 54:235–248, February 2014.

[24] C. K. Birdsall and A. B. Langdon. *Plasma physics via computer simulation.* 1985.

[25] John M. Dawson. Particle simulation of plasmas. *Rev. Mod. Phys.*, 55:403–447, Apr 1983.

[26] D. Tskhakaya, K. Matyash, R. Schneider, and F. Taccogna. The Particle-In-Cell Method. *Contrib. Plasma Phys.*, 47(8-9):563–594, 2007.

[27] Konstantin Matyash. *Kinetic Modeling of Multi-Component Edge Plasmas.* PhD thesis, Ernst-Moritz-Arndt-Universität Greifswald, 2003.

[28] J. Stoer and R. Bulirsch. *Numerische Mathematik 2.* Springer Verlag, 5th edition, 2005.

[29] Konstantin Matyash, Ralf Schneider, Andreas Mutzke, Oleksandr Kalentev, Francesco Taccogna, Norbert Koch, and Martin Schirra. Kinetic simulations of SPT and HEMP thrusters including the near-field plume region, December 2009. First presented at the ICNSP'09.

[30] J. Stoer. *Numerische Mathematik 1.* Springer Verlag, 9th edition, 2005.

[31] R. Hockney and J. Eastwood. *Computer simulation using particles.* Adam Hilger, 1981.

[32] Christopher A. Fichtl. *An Arbitrary Curvilinear Coordinate Particle In Cell Method.* PhD thesis, Nuclear Engineering, University of New Mexico, July 2010.

[33] J. P. Boris and R. A. Shanny, editors. *Proceedings of the Fourth Conference on Numerical Simulation of Plasmas.* Naval Research Laboratory, 1971.

[34] T. Takizuka and H. Abe. A binary collision model for plasma simulation with a particle code. *J. Comp. Phys.*, 25:205–219, 1977.

[35] Konstantin Matyash, Ralf Schneider, Andreas Mutzke, Oleksandr Kalentev, Francesco Taccogna, Norbert Koch, and Martin Schirra. Kinetic Simulations of SPT and HEMP Thrusters Including the Near-Field Plume Region. *IEEE Transactions on Plasma Since*, 38(9, Part 1):2274–2280, SEP 2010.

[36] Xiaoye S. Li. An overview of SuperLU: Algorithms, implementation, and user in- terface . *ACM Trans. Mathematical Software*, 31:302–325, September 2005.

[37] Leonardo Patacchini. Physics of Ion Thrusters' Plumes . Master's thesis, Massachusetts Institute of Technology, May 2007.

[38] M. E. Newman and G.T. Barkema. *Monte Carlo Methods in Statistical Physics.* Oxford University Press, 1st edition, 1999.

List of Figures

Acknowledgments

I wish to thank the whole work group of Prof. Ralf Schneider, who put me in touch with this Master Thesis and helped me with good advice and support during this thesis.

They always had time to discuss current topics and to help me, wether it was about my thesis work or just for fun. A special thanks goes to Jennifer Arndt who always offered great support and motivation during the creation of this thesis and the entire study. Furthermore, I would like to thank my entire family for all their support throughout my study and beyond.