UiO : Department of Mathematics
University of Oslo

Photoemissive charging of the Mercury Magnetospheric Orbiter studied by means of 3D Particle-In-Cell simulations

Trym Erik Nielsen

Master's Thesis, Spring 2020

This master's thesis is submitted under the master's program *Mechanics*, with program option *Mechanics*, at the Department of Mathematics, University of Oslo. The scope of the thesis is 60 credits.

The front page depicts a section of the root system of the exceptional Lie group E_8, projected into the plane. Lie groups were invented by the Norwegian mathematician Sophus Lie (1842–1899) to express symmetries in differential equations and today they play a central role in various parts of mathematics.

Here Comes the Sun
Here comes the sun
Here comes the sun
And I say, it's all right

Little darling
It's been a long cold lonely winter
Little darling
It seems like years since it's been here

Here comes the sun
Here comes the sun
And I say, it's all right

The Beatles, 1969

Abstract

This thesis presents results from plasma charging simulations of the Mercury Magnetospheric Orbiter (MMO) spacecraft orbiting Mercury in direct sunlight. These simulations were carried out using the Particle-In-Cell program PINC. The object-plasma interaction module of the PINC framework was extended by including a photoemission current algorithm based on the work by Cartwright et al. (Cartwright et al. 2000) and Deca et al. (Deca et al. 2013). The new photoemission current model was tested rigorously by comparing the results of test cases run in other Particle-In-Cell codes, proving the models validity. The charging behaviour of the MMO spacecraft was then simulated for varying plasma conditions, and with the spacecraft booms extended and retracted. The results of these simulations are shown to agree with the spacecraft charging theories developed by Langmuir (Mott-Smith et al. 1926), Whipple (Whipple 1981) and (Garrett 1981). The MMO charging simulations are also compared with similar Particle-In-Cell simulations, and their differences and similarities are discussed.

Acknowledgements

There are a great deal of people I have to thank for being able to finish this thesis. I would first like to thank my supervisor Mikael and my co-supervisor Wojciech for taking me on as a Master student even when I've had a full time job. Your patience with panicked e-mails and on the spur of the moment requests for zoomm meetings is deeply appreciated. You have both taught me a lot, and I look back fondly on the summer I was lucky enough to join you both in Japan for the 4D space Numerical Workshop.

I would like to thank all the great scientists and programmers who contributed to PINC and made it easier to navigate a complex code base with helpful comments; Vigdis Holta, Gullik Killie, Sigvard Marholm and Jan Deca thank you for creating an excellent piece of simulation software and for helping me to contribute to it. I would like to extend my deepest thanks and appreciation to Steffen Brask, also a contributor to PINC, who went above and beyond anything that could be expected in helping me tackle difficult programming conundrums and in giving me ideas of how I could implement my own code in PINC. You have made me a better programmer.

Without my family, I would not have been able to suffer through a job on top being a Master student, I thank you for your endless support and for allowing me to go on long nerdy rants about physics and the Dune books.

Finally, to my girlfriend Hélène. You kept me sane through what has been one of the toughest years of my life. Thank you so much for encouraging me, and for believing in me.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

On January 20th 1994 Canadians experienced an unexpected interruption to their television program: The Canadian Telsat spacecrafts Anik E1 and Anik E2 had experienced sudden failures of their gyroscopic guidance systems and had begun to tumble out of control. The out of control Anik spacecraft had experienced electrostatic discharge in the circuitry, causing permanent damage to critical systems. Later, this critical systems failure was attributed to the phenomenom of spacecraft charging. Although engineers were able to restore the gyroscopes of Anik E1, the Anik E2 would never recover, representing a loss of several hundred million dollars (Leach et al. 1995).

Spacecraft charging has been studied as a disparate field of space physics since the mid twentieth century, with fundamental theories describing the phenomena developed by Irving Langmuir in in the early 1920's. Langmuir and his colleagues worked extensively with electrode probes of different geometries. They studied how sheaths of charged particles formed around the probe at its so called "floating potential", i.e, when there is no net current flowing between the probe and its surroundings (Mott-Smith et al. 1926), (Garrett 1981).

The theory developed by Langmuir and others helped to develop the plasma instrument called the Langmuir probe that is still in use on modern spacecraft today. In addition, the theory gave scientists and engineers the necessary theoretical framework to analyze the charging of spacecraft with the key insight of the spacecraft as a floating probe.

With the advent of the development of spaceflight in the middle of the twentieth century, these theories were no longer only used in analyzing the charging properties of interstellar dust. Engineers and scientists applied these theories for the practical purpose of protecting sensitive electronics onboard spacecraft and rockets. The investigation of these practical concerns cumulated in the launch of the "Spacecraft Charging At High Altitudes" (SCATHA) spacecraft in 1979 with the mission of gaining a better understanding of the development of the charging process due to the formation of a plasma sheet, and in testing strategies for controlling the potential of a spacecraft (NASA 2020).

The SCATHA mission measured spacecraft potential in the day and night side of the orbit and found that the difference between these points in orbit was several hundred volts(Mullen et al. 1986). This difference is due to the photoelectric effect; when the conducting surfaces of the spacecraft are exposed to sunlight, electrons gain enough energy from the photons to escape the surface, thereby causing an outgoing negative current. The photoemission current on a

spacecraft is often much larger than the charging caused by electrons in the ambient plasma impinging on the spacecraft, in the case of the SCATHA, the photoelectric current was greater by a factor of 20 at some points in the orbit (Lai 2019).

Modern spacecraft are designed with spacecraft charging in mind; both passive and active methods for controlling the potential of a spacecraft exist and a good overview of such methods can be found in the work by Lai (Lai 2003). Nevertheless, spacecraft that carry scientific instruments as part of their payload must still account for the effects that spacecraft charging has on the measurements of these instruments.

One such spacecraft, and the focus of this thesis, is the joint Japan Aerospace Exploration Agency (JAXA) and European Space Agency (ESA) mission BepiColombo. Launched in 2018, the BepiColombo consists of two orbiters that will begin to explore Mercury and its surrounding plasma environment at its arrival in 2025 (Benkhoff et al. 2009). This thesis will focus on one of the orbiters exclusively, the Mercury Magnetospheric Orbiter (MMO). The MMO carries as part of its payload, charged particle detectors, and magnetometers to measure the solar wind plasma around Mercury and its magnetic field respectively (Saito et al. 2010), (Benkhoff et al. 2009).

The main aim of this thesis is to simulate the charging of the MMO spacecraft in its polar orbit around Mercury when the spacecraft is exposed to direct sunlight. By simulating the charging of the spacecraft we hope to gain a better understanding how the the formation of a plasma sheet would influence the data collection of the scientific payload onboard. We further wish to investigate what effect the two diametrically opposed booms attached to the spacecraft have on the charging behaviour of the MMO.

In order to accomplish this, a series of simulations will be carried out using numerical methods. At the University of Oslo (UIO) for the past years, the 4DSpace Strategic Research Initiative has developed a new Particle-In-Cell (PIC) called PINC, with the capability of simulating the charging of objects in an ambient plasma. PINC will be used as a framework in this thesis to simulate the charging of the MMO.

Due to Mercury's close proximity to the sun, the MMO will experience a large outgoing photoelectron current. We therefore chose to integrate methods for including a photoemission current from the spacecraft surface into the PINC framework. We will later in this thesis see that selecting an appropriate method for injecting the photoelectrons into the computational domain plays a significant role in the charging characteristics of the spacecraft. The methods developed for photoemission must also be rigorously tested for accuracy and precision, as such, we compare the results obtained by PINC for a testcase previously simulated with other PIC codes. This testcase is outlined in full in the paper by Deca et al. (Deca et al. 2013).

When the new photoemission code has been thoroughly tested, we can apply the code to simulate the charging characteristics of the MMO. We do this by setting up a range of numerical experiments, comparing configurations of the MMO with and without deployed booms, for varying conditions of the ambient plasma and external magnetic field. These results will be presented and compared to theory with an emphasis on where our results align with established theory, and where they deviate from expected results.

CHAPTER 2

Theoretical background

2.1 Plasma modelling

In this section, we present a short overview of the basic mathematical framework used in the study of plasma dynamics. In section Section 2.2.1, we briefly present the equations of motion for single particle motion in electrical and magnetic fields. Then in Section 2.3 we give an overview of essential plasma parameters. Plasma is a collection of ionized gas, commonly referred to as the fourth state of matter. The solar wind, the polar aurorae, and lighting are some examples of plasmas that occur in nature (Chen 2018). Plasma shares many of the same properties that describe gases, but differ in being affected by magnetic and electrical fields: since plasma consists of charged particles, ions and free electrons, they are subject to the Lorentz force. The Lorentz force acting on a charged plasma particle, causes curvilinear motion further complicated by the influence of other nearby charged particles.

2.2 Basic equations

2.2.1 Single particle description

The single particle description of a plasma describes the motion of individual charged particles moving in imposed magnetic and electrical fields. Assuming the force of gravity is sufficiently small, and assuming constant electrical and magnetic fields, the single particle motion of a charge q moving at velocity v in the electrical field $\mathbf{E}$ and magnetic field $\mathbf{B}$ (where bolded letters denote vector fields) is described by the Lorentz force law

$$\mathbf{F} = q\mathbf{E} + q\mathbf{v} \times \mathbf{B}, \tag{2.1}$$

where the term $q\mathbf{E}$ is called the Electric force, and the term $q\mathbf{v} \times \mathbf{B}$ is called the magnetic force. When a particle moves in a static magnetic field, and no electric field is present, the particle will gyrate around magnetic field lines. Setting E to zero, we have

$$\mathbf{F} = q\mathbf{v} \times \mathbf{B}. \tag{2.2}$$

The cross product of the velocity and magnetic field vector means that the magnetic force always acts perpendicularly to the direction of motion of the particle, thereby causing the particle to gyrate. Setting the centripetal force

equal to the magnitude of the Lorentz force, we can derive an expression for the gyroradius r_g of the motion

$$\frac{m_s v_\perp^2}{r_g} = |q| v_\perp \|\mathbf{B}\|, \tag{2.3}$$

where the subscript on m_s denotes the specie of the particle, and $v_\perp$ denotes the perpendicular component of velocity to the plane of $\mathbf{B}$. We have also used the notation $\|\mathbf{B}\|$ to denote the magnitude of the vector. Upon rearranging, the expression for the gyroradius r_g becomes

$$r_g = \frac{m_s v_\perp}{|q| \|\mathbf{B}\|}. \tag{2.4}$$

The particle gyrates with an angular frequency, called the cyclotron frequency Ω_c, expressed as

$$\Omega_c = \frac{v_\perp}{r_g} = \frac{|q| \ \|\mathbf{B}\|}{m_s}. \tag{2.5}$$

The motion of a particle q_s is in practical cases often modelled as the drift of the centre of gyration of the particle. When also subjected to an isotropic electrical field, this motion is called E cross B drift, or Hall drift, and can be derived from the Lorentz force equation and Newtons' second law and solving for the acceleration of the particle: If we assume the drift velocity to be constant in time, the expression for the drift $\mathbf{V}_D$ becomes

$$\mathbf{V}_D = \frac{\mathbf{E} \times \mathbf{B}}{B^2}. \tag{2.6}$$

2.2.2 Kinetic description

In the previous section, we studied the individual motion of charged plasma particles. Although the single particle description of plasma is useful in gaining an understanding in how individual particles behave in isotropic magnetic and electrical fields; it is an impractical model for analyzing macroscopic phenomena of the plasma. The introductory text "Plasma Physics: An Introduction to Laboratory, Space, and Fusion Plasmas" (Piel 2017) contains a good discussion of when each plasma model should be used. The kinetic description of plasma begins with the assumption of a density distribution of charges in the six dimensional phase space varying over time. Let $f_s(\mathbf{x}, \mathbf{v}, t)$ be the continuous probability distribution, representing the probability of finding a charged particle of species s at time t in phase space. Multiplying the distribution of charges by the charge of the species q_s, and integrating over the velocity space, the charge density ρ_s can be found. Summing over all the species of charge in the plasma gives us the following expression for the total charge density ρ_c

$$\rho_c = \sum_s q_s \int f(\mathbf{x}, \mathbf{v}, t) d^3\mathbf{v}. \tag{2.7}$$

Similarly, an expression for the current density is obtained by multiplying the charge distribution by the velocity vector v and integrating the result in a similar fashion to (2.7)

$$\mathbf{j} = \sum_s q_s \int \mathbf{v} f(\mathbf{x}, \mathbf{v}, t) d^3\mathbf{v}, \tag{2.8}$$

equipped with a continuous distribution function of particles in phase-space, equations of motion that describe the flow of the charged particles can be derived from solving the Boltzmann equation. Or alternatively, when assuming a non-collisional plasma, the set of vector equations called the Vlasov equation can be used

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \boldsymbol{\nabla} f_s + \frac{q_s}{m_s}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \boldsymbol{\nabla}_v f = 0. \tag{2.9}$$

Equation (2.9) is coordinate system independent, in PINC the Cartesian coordinate system is used for all computations. The operator notation $\boldsymbol{\nabla}_v = (\frac{\partial}{\partial v_x}, \frac{\partial}{\partial v_y}, \frac{\partial}{\partial v_z})$ and $\boldsymbol{\nabla} = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$ has therefore been used. For a comprehensive derivation of the Vlasov equation see (Birdsall et al. 2004) or (Piel 2017).

2.2.3 Fluid description

In previous sections, the equations of motion for characterising plasma has been modelled by analyzing the forces acting on individual particles. While this approach can be helpful in gaining insight into the physics governing the plasma behaviour, it is difficult to apply these frameworks to practical computation models.

Another method, that reduces the complexity of computing individual particle motions, is treating plasma as two continuous fluids. In this approach we are able to extract macroscopic properties of the plasma, such as the density, the velocity and the mean energy. The fluid equations are derived by taking the velocity moments of the Vlasov equation (2.9), where the generalized velocity moment is described as

$$\mathbf{M}^n \equiv \int \mathbf{v}\mathbf{v}\ldots\mathbf{v} f_s(\mathbf{x}, \mathbf{v}, t) d^3\mathbf{v}. \tag{2.10}$$

Where $\mathbf{M}^n$ is a tensor of rank n. The zeroth velocity moment, also called the continuity equation, can be found by multiplying equation (2.9) by the zeroth velocity moment

$$n_s(\mathbf{x}, t) = \int f_s(\mathbf{x}, \mathbf{v}, t) d^3\mathbf{v}, \tag{2.11}$$

the zeroth velocity moment for species s is simply the number density for the species. Multiplying equation (2.11) by the Vlasov equation we have

$$\int \frac{\partial f_s}{\partial t} d\mathbf{v} \int \mathbf{v} \cdot \boldsymbol{\nabla} f_s \mathbf{v} + \frac{q}{m} \int \mathbf{E} + \mathbf{v} \times \mathbf{B} \cdot \boldsymbol{\nabla}_v f_s \mathbf{v}\, d\mathbf{v} = 0. \tag{2.12}$$

With a considerable amount of manipulation (Chen 2018, ch 7), equation (2.12) reduces to

$$\frac{\partial n_s}{\partial t} + \boldsymbol{\nabla} \cdot (n_s \mathbf{v_s}) = 0. \tag{2.13}$$

In a similar fashion, the momentum equation can be derived from multiplying equation (2.9) by the first velocity moment

$$\mathbf{M}^1 = \int m_s \mathbf{v} f_s d^3\mathbf{v}. \tag{2.14}$$

This multiplication yields

$$\frac{\partial m_s n_s \mathbf{v_s}}{\partial t} + \boldsymbol{\nabla} \cdot \mathbf{P_s} - e_s n_s (\mathbf{E} + \mathbf{v_s} \times \mathbf{B}) = \mathbf{F_s}. \tag{2.15}$$

Where $\mathbf{P}_s$ denotes the pressure tensor field. The energy equation is derived from the second velocity moment

$$\mathbf{M}^2 = \int \frac{1}{2} m_s \mathbf{v}\mathbf{v} f_s d^3\mathbf{v}. \tag{2.16}$$

The second order moment described the flow of momentum, and is also called the stress tensor (Fitzpatrick 2015, Ch. 3). Applying the second velocity moment to equation (2.9) equation gives the partial differential equation

$$\frac{\partial}{\partial t}\left(\frac{3}{2} p_s + \frac{1}{2} m_s n_s v_s^2\right) + \boldsymbol{\nabla} \cdot \mathbf{Q_s} - e_s n_s \mathbf{E} \cdot \mathbf{v_s} = W_s + \mathbf{v_s} \cdot \mathbf{F_s}. \tag{2.17}$$

Where v_s is the thermal velocity of species s. The terms $\mathbf{Q}_s$ and W_s denote the energy flux density and total change of energy of species s respectively, and the notation $p_s = Tr(\mathbf{P})_s$ has been used for the scalar pressure, where Tr is the trace function. Equations (2.13), (2.15), and (2.17), are collectively known as the fluid equations, and written in their convective time-derivative form, can be reduced to

$$\frac{Dn_s}{Dt} + n_s \boldsymbol{\nabla} \cdot \mathbf{v_s} = 0 \tag{2.18a}$$

$$m_s n_s \frac{D\mathbf{v_s}}{Dt} + \boldsymbol{\nabla} \cdot p_s - e_s n_s (\mathbf{E} + \mathbf{v_s} \times \mathbf{B}) = \mathbf{F_s} \tag{2.18b}$$

$$\frac{3}{2}\frac{Dp_s}{Dt} + \frac{3}{2} p_s \boldsymbol{\nabla} \cdot v_s + \mathbf{p_s} \colon \boldsymbol{\nabla}\mathbf{v_s} + \boldsymbol{\nabla} \cdot \mathbf{q_s} = W_s. \tag{2.18c}$$

Formal derivations of the fluid equations, with some differences in notation, can be found in the fundamental works of Chen (Chen 2018), and Fitzpatrick (Fitzpatrick 2015).

2.2.4 Magnetohydrodynamics

Solving the equations of motion discussed in 2.2.1 coupled with Maxwell's equations is extremely computationally demanding. As such, two common approximations to these equations are used, the electrostatic model, and magnetohydrodynamics. In the electrostatic approximation of plasma, the plasma currents are presumed small, thereby simplifying maxwell's equations to solving poission's equation for the electrostatic potential.In the magnetohydrodynamics, plasma is treated as a neutral yet conducting fluid. Assuming a quasi-neutral, plasma at thermodynamic equilibrium Maxwell's equations (Hockney et al. 1988, Ch. 9-9-1) reduce to

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{j} \tag{2.19a}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{2.19b}$$

$$\nabla \cdot \mathbf{B} = \nabla \cdot \mathbf{E} = 0. \tag{2.19c}$$

The force equation can then be rewritten in terms of current density $\mathbf{j}$ and the conductivity σ as

$$\mathbf{j} = \sigma(\mathbf{E} + \mathbf{v} \times B). \tag{2.20}$$

From the assumption of a perfectly conducting fluid, and finite current density $\mathbf{j}$, then

$$\mathbf{E} + \mathbf{v} \times \mathbf{B} = 0. \tag{2.21}$$

From equation (2.21), equation (2.19b) reduces to

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B}). \tag{2.22}$$

The equations must then be closed using the fluid continuity equations and the momentum balance equations

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (\rho \mathbf{v}) \tag{2.23a}$$

$$\rho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \boldsymbol{\nabla} p + \mathbf{j} \times \mathbf{B}. \tag{2.23b}$$

Where ρ is the fluid density, p is the fluid pressure, and $\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}$ is the material derivative, i.e the rate of change of with respect to the moving fluid particle.

2.3 Plasma parameters

In this section, the basic parameters used in the analysis and simulation will be introduced. We discuss the notion of a plasma temperature, present the characteristic length and frequency scales the Debye length, and plasma frequency. Additionally, we discuss the principle of quasineutrality in relation to plasma.

2.3.1 Temperature

The temperature of a plasma is a measure of the average kinetic energy of the individual species that make up the plasma. For a mono-atomic gas, with a probability distribution $f(u)$, Chen (Chen 2018, Section 1.3) gives the following equation for the average kinetic energy

$$E_{av} = \frac{\int_{-\infty}^{\infty} \frac{1}{2} m u^2 f(u) du}{\int_{-\infty}^{\infty} f(u) du}. \tag{2.24}$$

By defining the variables

$$v_{th} \equiv (2KT/m)^{\frac{1}{2}} \quad \text{and} \quad y \equiv \frac{v_{th}}{u},$$

and integrating the numerator by parts, equation (2.24) reduces to

$$E_{av} = \frac{1}{2} KT. \tag{2.25}$$

Where K denotes the Boltzmann constant. Chen extends this argument for particles with three degrees of freedom. Integrating Maxwell's distribution in

three dimensions in a similar fashion to (2.24) and cancelling the integrals, the average kinetic energy is

$$E_{av} = \frac{3}{2}KT. \tag{2.26}$$

In plasma physics, it is common to use average energy rather than temperature as they are so closely linked. Calculations are often reduced in complexity by using the electron volt as the basic units of energy. It is defined as the amount of energy required to move a single electron across an electrical potential difference of 1 volt. One eV is therefore approximately equal 1.6x10^{-}19 J, with a conversion factor to temperature, in Kelvin

$$1eV \approx 11600K.$$

2.3.2 Debye length

The Debye length is a parameter that measures the net persistence of the electrostatic effect of a charged particle. This effect is also known as Debye shielding, (Hutchinson 2002, Section 3.1.2) presents a definition for the Debye length by analyzing the potential profile of a test charge placed in a cold plasma. In a non isothermal plasma ions can be assumed to be stationary in relation the more energetic electrons. The electrons density is then determined from the Boltzman factor

$$n_e = n_\infty \exp\left(\frac{e\phi}{T_e}\right). \tag{2.27}$$

Where T_e denotes the electron temperature, n_∞ is the density of electrons far away from the perturbing test charge, and where ϕ is the electrical potential as a function of the radial distance from the test function. Inserting equation (2.27) into Poisson's equation;

$$\nabla^2\phi = -\frac{\rho}{\epsilon_0} = \frac{-e}{\epsilon_0}(n_i - n_e) = \frac{-e}{\epsilon_0}n_\infty\left[1 - \exp\left(\frac{e\phi}{T_e}\right)\right]. \tag{2.28}$$

Assuming $e\phi \lll T_e$, then it is reasonable to only keep the linear terms of the Taylor expansion of the $\exp\left(\frac{e\phi}{T_e}\right)$ term on the right hand side of equation (2.28). Poisson's equation then takes the form of Helmholtz equation

$$\nabla^2\phi - \frac{1}{\lambda_D^2}\phi = 0, \tag{2.29}$$

where the Debye length λ_D has been defined as follows:

$$\lambda_d \equiv \sqrt{\frac{\epsilon_0 kT_e}{n_e e^2}}. \tag{2.30}$$

The solution of equation (2.29) takes the form

$$V \propto \exp\left(\frac{\pm x}{\lambda_D}\right).$$

The Debye length is then a measure of the shielding distance or thickness of the sheath that forms around a charged object embedded in a plasma (Chen 2018, Section 1.4).

2.3.3 quasineutrality

Now that we have presented an expression for the Debye length, the approximation of quasineutral plasma can be defined: When the length scale of interest is much smaller than the Debye length, i.e $\lambda_D \lll L$, the density of ions is approximately equal to the density of electrons (Chen 2018, Section 1.4). Or in other terms

$$n_i \approx n_e \approx n. \tag{2.31}$$

Where n is a common density called the plasma density. When this approximation holds, the plasma is said to be quasineutral. Note that this approximation still allows for variations in charge at smaller length scales than λ_D.

2.3.4 Electron plasma frequency

In plasma simulations of the solar wind an important physical phenomena is plasma oscillation, otherwise called Langmuir waves. In a neutral plasma, disturbances in the density of electrons (and ions) cause oscillations due to the restorative Coulomb force. For solar wind, and other cold plasmas, the frequency of the oscillations of electrons are expressed as

$$\omega_{pe} = \sqrt{\frac{n_e e^2}{\epsilon_0 m_e}}. \tag{2.32}$$

ω_{pe} is called the electron plasma frequency. A similar expression exists for the oscillations of ions, but due to their higher mass they oscillate much slower rate than the electrons. Equation (2.32) is formally derived from Maxwell's equations (Fitzpatrick 2015), (Hutchinson 2002) by assuming an infinite plasma not affected by an external magnetic field, no thermal motion i.e $KT = 0$, and ion fixed in space in a uniform distribution (Chen 2018, Section 4.3).

2.4 The solar wind plasma environment

In this section, a brief overview of the physics and basic plasma parameter values of the solar wind are presented. These data will be necessary in building the computational parameters required for the charging simulations of space probes moving in the interplanetary medium presented later in this thesis.

The solar wind is a non-isothermal plasma with a temperature averaging around 10 eV, that is primarily composed of free electrons, ions and alpha particles. The density of the solar wind plasma is relatively low as compared to other naturally occurring plasmas with a mean value of around 5 cm^{-3}. It flows with a drift velocity ranging from 400 km/s up to 900 km/s during periods of high solar activity (Lai 2019). In close proximity to the sun, the solar wind density increases significantly, potentially charging space probes to values as low as hundreds of volts (Lai 2019), (Deca et al. 2013), (Garrett 1981).

Interactions with solar system objects

The properties of the solar wind can change significantly in regions in close proximity to solar system objects. Planets with internal magnetic fields, will

cause magnetohydrodynamic (MHD) shock formation (Luhmann et al. 2004), (Benna et al. 2009), and bodies without an intrinsic magnetic field may still affect the flow of the solar wind plasma due to induced magnetic fields from solar wind charge collection or interaction with the ionosphere of the body.

This thesis is concerned with simulating the charging characteristics of the BepiColombo satellite, the MMO (Mercury Magnetospheric Orbiter). Mercury has an intrinsic, yet relatively weak, magnetic field. A bow-shock forms due to the compression of the solar wind from the obstruction of the magnetosphere of the body in question, the shocks location and properties changes dynamically depending on the pressure exerted by the solar wind. Immediately behind the bow-shock is the region called the magnetosheath. In this region the charged particles have a lower density yet higher average temperature than that of the solar wind (Benna et al. 2009).

Separating the magnetosphere of the earth, or Mercury, is the magnetopause. The magnetopause is defined as the region around an object where the dynamic pressure of the solar wind balances with the pressure from the intrinsic magnetic field of the object. Beyond the magnetopause, the intrinsic magnetic field of the planet dominates. In this region charged plasma particles move in spirals following magnetic field lines of the planet, the magnetosphere of Earth also contains a region of high density cold plasma called the plasmasphere. The composition of the plasma, its temperature and density in these regions vary significantly. The charging of a spacecraft will then vary significantly as a spacecraft orbits a planet, and it is therefore important to determine the location of these regions with respect to the planet.

The location of the magnetopause of a planet with an intrinsic field can be approximated by substituting the distance of the magnetopause r_M into the balance expression for the planets magnetic field strength. Balancing the dynamic pressure of the sun with the magnetic pressure (Meyer-Vernet 2007), we have

$$\rho_w v_w^2 \sim B_\mu^2 / 2\mu_0. \tag{2.33}$$

Where ρ_w and v_w is the solar wind density and velocity respectively, B_μ is the dipole magnetic field strength, and μ_0 is the vacuum permeability. The magnetic field strength is expressed as

$$B_\mu = (\mu_0/4\pi)\mu/r_M^3, \tag{2.34}$$

where μ is the permeability of the planet. However, the magnetopause must carry a current to confine the planets dipole magnetic field. This current produces a magnetic field on the order of magnitude to B_ν. Therefore, just inside the magnetopause, the magnetic field strength is approximately twice B_ν. The balance equation (2.33) then becomes

$$\rho_w v_w^2 \approx \frac{(2B_\mu)^2}{2\mu_0}. \tag{2.35}$$

Substituting in the magnetic field strength from equation (2.34) the distance of the magnetopause on the sun-ward side to the planet is approximately

$$r_M \approx \left(\frac{\mu}{4\pi} \sqrt{\frac{2\mu_0}{\rho_w v_w^2}} \right)^{\frac{1}{3}}. \tag{2.36}$$

This can be expressed as a relative distance by introducing the radius of the planet R and the magnetic field strength at the equator $B_0 = (\nu_0/4\pi)\nu/R^3$

$$\frac{r_m}{R} \approx \left(\frac{2B_0^2}{\nu_0 \rho_w v_w^2} \right)^{1/6}. \tag{2.37}$$

In her book (Meyer-Vernet 2007), Meyer-Vernet also gives an approximation for the distance from the planet to the bow-shock by assuming a conical function describing the radius r from the foci (the planet) to the bow shock. The radius r on the line extending from the planet to the sun is approximately

$$r \simeq (1 + \epsilon)\, r_M. \tag{2.38}$$

Where ϵ is the eccentricity of the conic section. Typically this value is around $\epsilon \simeq 0.5$.

2.5 Spacecraft charging in plasma

A spacecraft collects charged particles when moving relatively to ambient plasma, this change in net charge on the spacecraft causes an electrical field according to Gauss's Law (Lai 2019, Ch. 1). When the charging of the spacecraft reaches a non-oscillatory steady state, the sum of currents from all the species in the plasma is zero, and the net charging is constant. This is expressed as

$$\frac{\mathrm{d}Q}{\mathrm{d}t} = \sum_s I_s(V) = 0. \tag{2.39}$$

The solution of equation (2.39) for V is called the floating potential of the spacecraft, and is relative to the neutral plasma. The floating potential of a spacecraft in a drifting plasma is usually negative, since the electrons move at a higher speed than the ions.

For complex geometries it is non-trivial to find analytical expressions for the currents I_s charging the spacecraft; in this section, Langmuir probe theory and orbital motion-limited theory is discussed as the basic tools for analyzing spacecraft charging with simple geometries. The primary charging mechanisms that effect the overall potential of a spacecraft are then presented. Special attention is given to the photoelectric effect and the spacecraft photoelectric current, as the implementation of this current in a computational model is one of the main objectives of this thesis.

2.5.1 Langmuir probe theory

A Langmuir probe is a simple device used in the laboratory to measure the temperature, the density, and the electrical potential of a plasma. A Langmuir probe consists of one or more, often spherical, electrodes inserted into the ambient plasma. Either a time varying or constant potential is induced between the electrodes, or between the electrodes and some electrical ground, causing the probe to collect charged particles. By analysing the current $I(V)$ for different potentials, the properties of the surrounding plasma can be extracted (Garrett 1981), (Lai 2019), (Miloch et al. 2012). When immersed in a plasma, a spacecraft behaves in a similar fashion to a Langmuir probe. The difference is that in the case of a spacecraft a potential is not applied, but can be measured as a response to incoming and outgoing currents (Lai 2019, Ch. 2).

Orbital motion-limited theory

In orbital motion limited theory, or OML theory for short, the charge flux impinging on a spacecraft is determined from the conservation of energy and angular momentum of species travelling in proximity to a spacecraft (Lai 2019), (Garrett 1981). From conservation of energy, an expression for the impact velocity on a spherical spacecraft can be obtained, and from conservation of momentum an expression for the distance from the centre of the spacecraft to the straight line of travel can be found. In (Lai 2019, Ch. 2.1) these expressions are given as

$$v_a = v_{th}\left(1 - \frac{q\phi}{kT}\right) \tag{2.40a}$$

$$h = a\left(1 - \frac{q\phi}{kT}\right). \tag{2.40b}$$

Integrating the particle distribution $f_s(\mathbf{x}, \mathbf{v}, t)$ using equation (2.40a) and (2.40b) as limits of integration, we can find the flux of particles that impact the spacecraft. It is often convenient work with a normalized potential variable in the formulas for current, define

$$\eta_s \equiv -\frac{q_s\phi}{kT_s}. \tag{2.41}$$

Using the normalized potential, particles with $\eta_s < 0$ will be repelled, and particles with $\eta_s > 0$ will be attracted to the spacecraft (Marholm 2019). Expressions for the integration of the particle distribution has been found for some simple geometries, for species with $n_s > 0$ with the assumption of a plasma with no internal magnetic field and zero drift, by Mott-Smith and Langmuir (Mott-Smith et al. 1926)

$$I_s(\eta_s) = I_{th,s} \tag{2.42a}$$

$$I_s(\eta_s) = I_{th,s}\left(\frac{2}{\sqrt{\pi}}\sqrt{\eta_s} + \exp(\eta_s)\,\mathrm{erfc}(\sqrt{\eta_s})\right) \tag{2.42b}$$

$$\approx I_{th,s}\frac{2}{\sqrt{\pi}}\sqrt{1+\eta_s} \tag{2.42c}$$

$$I_s(\eta_s) = I_{th,s}(1+\eta_s). \tag{2.42d}$$

Where equations (2.42a), (2.42b) and (2.42d) are the functions of current for a plane, a cylinder, and a sphere respectively. These functions are derived based on the assumption that the cylinder is long compared to λ_D, the radius of the sphere and the cylinder to be small compared to λ_D, and the plane to be wider than λ_D. The notation I_{th} is used to denote the current collected when the object is neutrally charged, i.e $\eta_s = 0$, and $erfc(\sqrt{\eta_s})$ is the complementary error function defined as

$$\begin{aligned}\mathrm{erfc}(z) &\equiv 1 - \mathrm{erf}(z)\\ &= \frac{2}{\sqrt{\pi}}\int_z^\infty e^{-t^2}dt.\end{aligned}$$

Finally, for any repelled species, i.e. $\eta_s < 0$, the current is given by

$$I_s(\eta_s) = I_{th,s}\exp\{\eta_s\}. \tag{2.44}$$

Where equation (2.44) holds for any geometric shape.

The photoemission current

In 1921, Albert Einstein was awarded the Nobel price in physics for his discovery of the mathematical law governing the photoelectric effect (Nobel Media AB 2020). Qualitatively, the photoelectric effect occurs when a quantum of light, a photon, is absorbed by an electron in some material causing its energy to exceed the binding energy of that material and being ejected from the surface of the material. This binding energy of the material is called the work function of the material. Mathematically, with differences in notation (Einstein 1905), Einstein's law is expressed as

$$E = h\nu - W. \tag{2.45}$$

Where E denotes the maximum kinetic energy of the ejected electron, h is Planck's constant, ν is the wavelength of the light quanta, and W_f denotes the work function of the material. It is often useful in engineering to express equation (2.45) in terms of the threshold frequency ν_0 at which the photons have enough energy to cause photoemission. Using the relation $E = W_f = h\nu_0$, equation (2.45) becomes

$$E_k = h(\nu - \nu_0). \tag{2.46}$$

From equation (2.45), it is evident that the photoelectron yield is dependent on the material composition of the spacecraft. Table 2.1 shows the work functions for some of the materials commonly used in space applications, these data have been reproduced from (Feuerbacher et al. 1972) and (Taylor and Francis Group 2019).

Material	**Work function (eV)**
Aluminium	4.0
Gold	4.2
Silver	4.62
Stainless steel	4.4
Graphite	4.7

Table 2.1: Work function for common materials used in space applications

With an expression for the minimum frequency below which photoemission occurs it possible to compute the photoelectron current; defining a photoelectron yield $Y(\omega)$ (photoelectrons per photon at some energy ω) as

$$Y(w) = \begin{cases} 1 & if \quad \omega > h\nu_0 \\ 0 & if \quad \omega \leqslant h\nu_0. \end{cases} \tag{2.47}$$

Then, the photoelectron current flux J_{ph} can be found by integrating over the energy of the incoming photons (Chen 2018, Ch. 7)

$$J_{ph} = \int_0^\infty Y(\omega) f_s(\omega)\, d\omega. \tag{2.48}$$

Strictly speaking, equation (2.47) is only an approximation for the yield of photoelectrons. A more accurate description has to account for the amount of photons reflected from the surface, thereby not contributing to the total

photoelectron current flux. Defining the reflectance of the surface exposed to sunlight as $R(\omega)$, equation (2.48) becomes

$$J_{ph}(R) = \int_0^\infty Y(\omega, R) f_s(\omega)\, d\omega. \tag{2.49}$$

Both the photoelectron yield, and the reflectance of the spacecraft material is additionally dependent on the incident angle θ of the surface to the direction of travel of the photons. Lai gives the following equation for current flux $J_{ph}(\theta)$ when incidence angle is accounted for

$$J_{ph}(\theta) = \int_0^\infty f_s(\omega) Y^*[\omega, R(\omega, \theta)][1 - R(\omega, \theta)] \cos\theta\, d\omega. \tag{2.50}$$

Where $Y^*(\omega, R(\omega, \theta))$ is the photoelectron yield per *absorbed* photon. Using estimates for terms Y^* and $1 - R(\omega, \theta)$ and cancelling cosine terms in the product, equation (2.50) becomes

$$J_{ph}(\theta) = \int_0^\infty f_s(\omega) Y(\omega)(1 - R(\omega)) \cos\theta\, d\omega \tag{2.51a}$$

$$J_{ph}(\theta) = J_{ph}(0) \cos\theta. \tag{2.51b}$$

Finally, the photoelectron current $I_{ph}(\theta)$ can be expressed as

$$I_{ph}(\theta) = J_{ph}(0) A \cos\theta, \tag{2.52}$$

where A is the surface area of the spacecraft exposed to sunlight. Inserting equation (2.52) into equation (2.39), and with a known photon flux from blackbody radiation theory, the floating potential for a spacecraft subject to photoelectric and ambient plasma charging can be computed.

The sun as a blackbody radiation source

In the previous section, an expression for the photoelectron current was shown. If we are to compute the floating potential of a spacecraft, it is necessary to either know the value of the photoelectron current a priori or to compute its value based on the amount of photon flux through the spacecraft surface. In this section we discuss the theory of blackbody radiation as a method of computing solar photon flux.

A blackbody is an object in thermodynamic equilibrium that absorbs all frequencies of radiation, without reflecting any of the incoming radiation (Planck 2020). In Planck's seminal paper on the theory of heat radiation, Planck gives the following three conditions that a blackbody must satisfy (Planck 2020)

> First, the body must have a black surface in order to allow the incident rays to enter without reflection. [...] Second, the black body must have a certain minimum thickness depending on its absorbing power, in order to insure that the rays after passing into the body shall not leave it again at a different point of the surface. [...] Third, the black body must have a vanishingly small coefficient of scattering. Otherwise the rays received by it would be partly scattered in the interior and might leave again through the surface

The sun does not perfectly absorb all frequencies of electromagnetic radiation, and neither is it in perfect thermodynamic equilibrium; sunspots have a measurably lower temperature than the surrounding photosphere (Solanki 2003). However, energy absorbed and emitted by the sun is in steady state and can therefore be approximated as a black body radiator.

In "The theory of heat radiation" Planck derives the following expression for the energy density of the electromagnetic radiation emitted by a blackbody

$$B(\nu, T) = \frac{2h\nu^3}{c^2} \frac{1}{\exp\left(\frac{h\nu}{k_B T}\right) - 1}. \tag{2.53}$$

Where ν is the frequency of the electromagnetic radiation emitted, T is the temperature of the blackbody measured in Kelvin, c is the speed of light, h is the planck's constant, and k_b is the Boltzmann constant. Equation (2.53) is called Planck's law, and can be similarly expressed in terms of the wavelength of the radiation λ.

$$B(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{\exp\left(\frac{hc}{\lambda k_B T}\right) - 1}. \tag{2.54}$$

The spectral energy computed from the equations above can be expressed in many units depending on the problem at hand. Of particular interest in this thesis is expressing the spectral energy as the number of photons emitted at a particular wavelength. In the derivation of Planck's law; Planck is often credited as the father of modern quantum mechanics by historians for his idea that the radiation emitted by a blackbody at certain frequencies as packets of energy

Figure 2.1 plots the irradiance given by equation (2.54) for some values of T, the suns radiative output can be approximated by assuming a blackbody temperature $T \approx 5800K$, shown in the figure as a blue line.

The radiation calculated from equation (2.53) is commonly given in the units $Wm^{-2}Sr^{-1}nm^{-1}$, called spectral radiance, it is the radiance of a surface per unit wavelength. In figure 2.1, the radiance is multiplied by the ratio $\frac{R_\odot}{a_\oplus}$ (where $R_\odot$ is the radius of the Earth, and $a_\oplus$ is the solar radius) to give the spectral irradiance as measured on earth, the units then are given as $W^{-2}m^{-2}nm^{-1}$.

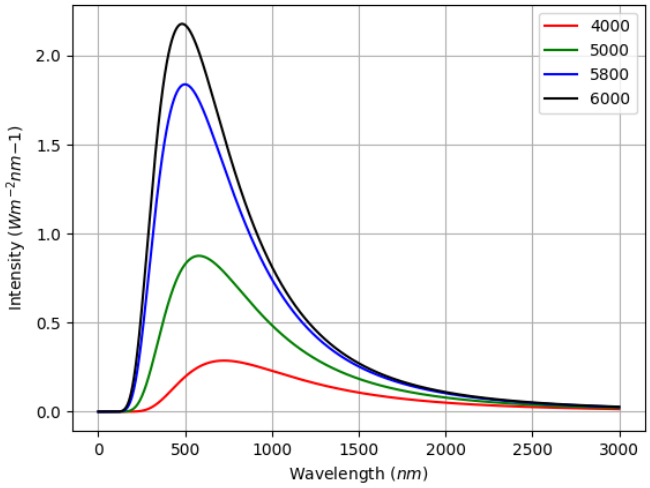


Figure 2.1: Irradiance of blackbody radiation for different temperatures

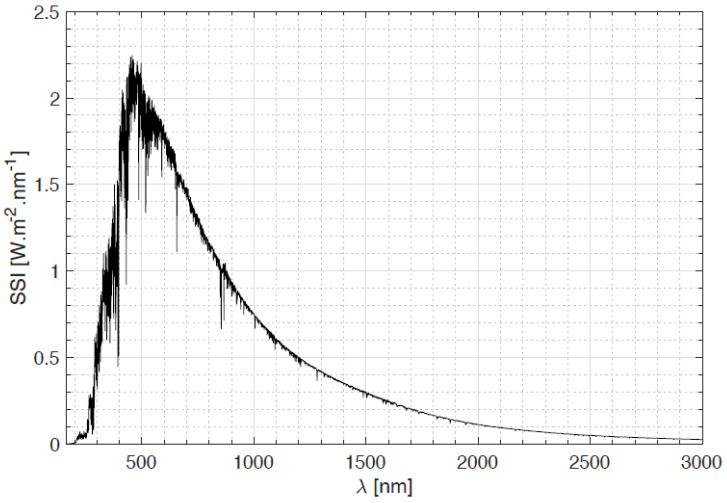


Figure 2.2: Solar irradiance as measured onboard the International Space Station, reused from ESA webpages (ESA 2020a)

Comparing figure 2.1 to figure 2.2, it can be seen that aside from smaller fluctuations in electromagnetic radiation, due to absorption of energy at certain frequencies in the solar atmosphere, our approximation of the sun as a blackbody emitter is a good one.

Given the work function (or more accurately, the cut-off wavelength or frequency) of the material of a spacecraft, and assuming a photoelectron yield of one photoelectron to one photon under the cut-off wavelength, integrating equation (2.54) gives the photoelectron flux emitted by the surface of a spacecraft

exposed to sunlight. Integrating planck's law for a given frequency band is non-trivial, in 3, the method for integrating Planck's law by Widger and Woodall (Widger et al. 1976) is presented in detail.

Secondary and backscattered electrons

Secondary and backscattered electrons can, depending on the ambient plasma environment and the material properties of the spacecraft in question, contribute significantly to the overall charge of a spacecraft. Both secondary and backscattered electrons provide a source of outgoing electrons from surfaces (Lai 2019), (Lai 2005), (Miloch 2015). At higher electron temperatures, the electrons impinging on a spacecraft may have high enough energy to interact with the electrons embedded in the surface material thereby releasing one or more electrons from the surface. The yield of secondary electron emission, or probability of emission, is a function of energy and is denoted as $\delta(E)$.

In a finite range ambient electron temperature, certain materials $\delta(E)$ can exceed one; gold, aluminum oxide, and Kapton are some common materials used in spacecraft for which $\delta(E)$ exceeds one (Lai and Tautz 2006), (Lai 2019). When $\delta(E) > 1$ there is a net total flux of outgoing electrons thereby causing the spacecraft to charge positively. Since the outgoing flux of electrons are dependent on the incoming electrons, one can calculate the secondary electron flux based on the incoming electron distribution $f(E)$. The current flux equation for secondary electrons is then

$$j_s = \int_0^\infty \delta(E) f(E) dE. \tag{2.55}$$

Backscattered electron emission is similar to secondary electron emission in that the mechanisms are dependent on incoming electrons from the ambient plasma. An electron is backscattered due to an inelastic collision with some surface material. The outgoing electron is the same electron that collided with the surface material, backscattering of electrons can then be thought of as a form of reflection. Since the outgoing and incoming electron is the same particle, the probability of backscattering, $\eta(E)$, cannot exceed 1.

Similarly to the secondary electron emission mechanism, the backscattering effect is dependent on the distribution of the electrons in the ambient plasma, so a similar integral to equation (2.55) can be formed for the backscattered electron current flux

$$j_b = \int_0^\infty \eta(E) f(E) dE. \tag{2.56}$$

Artificial sources of spacecraft charging

Artificial sources of charging can occur both as a method for charge mitigation on spacecraft, as well as an adverse side effect of charged particle emitting equipment onboard. The SCATHA spacecraft launched by NASA was one of the first flown mission that successfully proved that spacecraft could be discharged safely without damaging scientific instruments (Wade 2020).

Charge mitigation methods can be subdivided into active and passive methods, the difference being passive methods require no control mechanisms and are automatic, whereas active mitigation requires some form of control

system. These systems can be further subdivided into where the discharging occurs, whether it is on the spacecraft frame or dielectric surfaces.

One of the simplest methods for discharging a spacecraft is by using a simple sharp spike protruding from the spacecraft surface. Since the radius of curvature is small at the tip of the spike, a large electrical field is generated causing electron emission from the tip. The flux of outgoing electrons reduces the negative potential the spike is connected to. Lai (Lai 2003) gives the following equation, with differences in notation, for the current flux of a sharp spike

$$j_{spike} = AE^2 \exp\left(-\frac{BW^{\frac{3}{2}}}{E}\right). \tag{2.57}$$

Where A and B are some constant, E is electric field strength and W is the surface material work function.

Not all sources of artificial charging are beneficial however, ion engines expel high density plasma that may return to the already charged surfaces of the spacecraft. Differential charging between conducting and dielectric surfaces can occur in this case, causing damage to onboard electrical equipment. Deriving analytical expressions for charging due to electrical thrusters is impractical, and is typically simulated; (*ESA Science & Technology - BepiColombo plasma simulation* 2020) shows one such simulation of the mercury transfer module thruster on the BepiColombo spacecraft.

CHAPTER 3

Numerical Method

In the previous chapter, the basics of plasma physics and the mathematical tools required to analyze plasma dynamics were introduced. Solving the equations of motion for the millions of plasma particles analytically is impractical, this section will outline the Particle-in-Cell (PiC) algorithm as a framework for computer simulation of plasma. A method for implementing photoelectron emission from conducting surfaces is emphasized. First, a general description of the PiC algorithm is introduced as well as the stability criteria of the algorithm. Then the implementation of the PiC algorithm in the PINC framework is discussed with emphasis on the implementation of object charging in plasma. Finally, a method for implementing photoelectron emission into the PINC object module is presented.

3.1 Particle In Cell algorithm

The particle in cell algorithm is a method used in computational plasma physics to analyze large systems of many particles in a computationally efficient manner. This is achieved by the introduction of super particles, computational particles, that represent many real particles, then interpolating the forces acting on these super particles to a spatial grid.
There are three main ways of simulating the forces acting on a system particles. The particle-particle method (PP) where forces are computed between individual particles. The particle-mesh method (PM) where the forces between the particles are computed as field quantities on the spatial mesh. And the particle-particle-mesh method (PPPM or P^3M), which is a combination of the two earlier methods (Birdsall et al. 2004).
By far the simplest method computationally is the PP method. However, since all forces between each individual pair of particles are computed, the method is also the most computationally expensive. If the system of interest contains N_p particles, then the number of operations scale as $\mathcal{O}(N_p^2)$ (Hockney et al. 1988, p.20). It is therefore impractical to use the PP method for all but the simplest systems, even on highly parallel High Performance Computers (HPC) available today.
The PM method computes the forces of a system as field quantities, first by assigning the charges in the system to the mesh by some method, then solving Poisson's equation on the mesh, then computing the forces on the mesh points and interpolating to the individual particles. This method is therefore faster, but usually not as accurate as computing the forces on all particle pairs directly.

With N_g grid points, the complexity of this method scales as $\mathcal{O}(N_g \log N_g)$ (Hockney et al. 1988) thus making this method much more applicable to larger systems than the PP method.

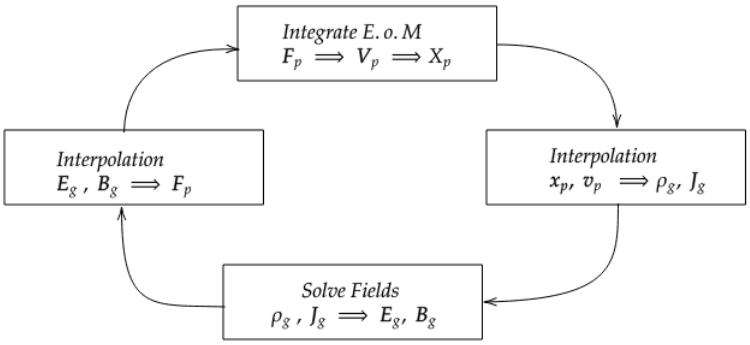


Figure 3.1: Particle in cell compute cycle

The PiC algorithm is an example of the PM method, figure 3.1 shows an overview of a computational cycle for each timestep in the PiC algorithm. Beginning at the box on the right in the figure, a distribution of N_p particles with position $\mathbf{x}_p$ is interpolated to get the charge ρ_g and current density $\mathbf{j}_g$ at the surrounding grid points. The two most common methods for this interpolation is the Nearest Grid Point (NGP) scheme, and Cloud in Cell (CIC) scheme.
Once the charge and current densities are known on the grid, the next step is to solve for the **E** and **B** fields. This is accomplished by solving Maxwell's equations at the grid points

$$Gauss's\ Law : \nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \tag{3.1a}$$

$$Gauss's\ Law\ for\ magnetism : \nabla \cdot \mathbf{B} = 0 \tag{3.1b}$$

$$Maxwell - Faraday's\ equation : \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{3.1c}$$

$$Ampere's\ Law : \nabla \times \mathbf{B} = \mu_0 \left(\mathbf{J} + \epsilon_0 \frac{\partial \mathbf{E}}{\partial t} \right) \tag{3.1d}$$

Where ρ is the charge density, ϵ_0 is the vacuum permittivity, μ_0 is the vacuum permeability, and **E** and **B** are the electric and magnetic fields respectively. With the electric and magnetic field computed at the grid nodes, the force on each super particle is computed from Lorentz's force equation and interpolating back to the position of the super particles. Using some numerical integrator the new position and velocity of the super particle is updated from the computed forces, and the cycle can then be repeated for the next timestep.

3.2 Particle in Cell implementation in PINC

In this section, we discuss the methods implemented in PINC that are required for the main PiC compute cycle described in figure 3.1 with focus on the particular schemes used in this thesis. The scheme used in PINC for integrating the equations of motion are presented, followed by the multigrid field solver, and finally the particle weighting scheme is discussed. PINC is the work of many researchers, but have been mainly implemented by Sigvald Marholm, Gullik Killie (Killie 2016), Vigdis Holta (Holta 2018), and Steffen Brask (Brask 2018). Significant contributions have also been made by Jan Deca, who implemented the module for object charging in PINC. This module will be discussed in more detail later in this chapter.

3.2.1 Integration of the equations of motion

Planet Mercury possesses and internal magnetic field, as such the plasma surrounding the planet is magnetized. To solve for the motion of magnetized plasma, the Boris algorithm is used to integrate the equations of motion. The Boris algorithm is a variant of the well known leapfrog method, in which the position and velocity of a particle is updated at half timesteps in staggered fashion. Like the leapfrog method, the Boris algorithm is an energy conserving integrator; the merits of the Boris algorithm as the de facto particle mover is further expanded upon in the publication (Qin et al. 2013).
Using the same notation as defined in (Birdsall et al. 2004) the Lorentz force is discretized as

$$\frac{\mathbf{x}_p^{t+\Delta t} - \mathbf{x}_p^t}{\Delta t} = \mathbf{v}^{t+\frac{\Delta t}{2}} \tag{3.2a}$$

$$\frac{\mathbf{v}_p^{t+\Delta t} - \mathbf{v}_p^{t-\Delta t}}{\Delta t} = \frac{q_s}{m_s}\left(\mathbf{E}_p + \frac{\mathbf{v}_p^{t+\Delta t} - \mathbf{v}_p^{t-\Delta t}}{2} \times \mathbf{B}_p\right) \tag{3.2b}$$

In the Boris algorithm equation (3.2b) is decomposed into a series of updating steps. First, half the acceleration is added, then the intermediary velocity vector is rotated due to the external magnetic field $\mathbf{B}$, and finally the second half of the acceleration is added.

$$\mathbf{v}^- = \mathbf{v}_p^{t-\Delta t} + \frac{q_s}{m_s}\mathbf{E}_p\frac{\Delta t}{2} \tag{3.3a}$$

$$\mathbf{v}_p^{'} = \mathbf{v}_p^- + \mathbf{v}_p^- \times \mathbf{T} \tag{3.3b}$$

$$\mathbf{v}_p^+ = \mathbf{v}_p^- + \mathbf{v}_p^{'} \times \mathbf{S} \tag{3.3c}$$

$$\mathbf{x}_p^{t+\Delta t} = \mathbf{v}_p^+ + \frac{q_s}{m_s}\mathbf{E}_p\frac{\Delta t}{2} \tag{3.3d}$$

Where the rotational parameters $\mathbf{T}$ and $\mathbf{S}$ are expressed as

$$\mathbf{T} = \hat{\mathbf{B}}_p \cdot \tan\left(\frac{q_s \Delta t}{2m} B_p\right) \tag{3.4a}$$

$$\mathbf{S} = \frac{2\mathbf{T}}{1 + \|\mathbf{T}\|^2}. \tag{3.4b}$$

Where $\hat{\mathbf{B}}_p$ is the magnetic unit vector acting on particle p. Equations (3.3) are equally suited for plasmas with time varying magnetic fields as for electrostatic plasma with a constant external magnetic field. Since PINC is an electrostatic model, equations (3.4) is solved once before the main time loop, then applied to each call to the mover method.

3.2.2 Field Solver

Several field solvers exists in PINC, in this thesis the multigrid solver developed by Killie (Killie 2016) has been used. The multigrid solver is an iterative method; the basic principle of the method in a PiC context is to solve Poisson's equation first on a coarse grid, then using the solution on the course grid as a guess, solve the equation again for a finer grid. The solution for the coarse grid speeds up the solution for the finer grid, reducing the total computational time required to converge to an accurate solution.

There are three main types of multigrid solver, divided into the so called V-cycle, W-cycle and F-cycle, defined by when the algorithm should use a coarser or finer mesh than used in the previous iteration. Multigrid solvers are highly flexible, and many researchers have spent considerable effort in order to optimize the number of iterations (Trottenberg et al. 2000). Highly optimized multigrid solvers have a local complexity given by $\mathcal{O}(N_g)$, with a global complexity of $\mathcal{O}(N_g log(N_g))$ when using domain decomposition like in the case of PINC (Trottenberg et al. 2000).

As described earlier, the multigrid method solves Poisson's equation in PINC. Strictly speaking, the solution to Gauss's law, equation (3.1a) is solved. In PINC however, electrostatic plasma is assumed, in this case the electric field is irrotational, i.e. $\nabla \times \mathbf{E} = 0$ and the electric field can then be represented as the gradient of a scalar potential field $\mathbf{E} = \nabla\phi$. Substituting the potential field back into Gauss's law, equation (3.1a) we have Poisson's equation

$$\nabla^2\phi = \frac{\rho}{\epsilon}. \tag{3.5}$$

In PINC this equation is solved iteratively using the Gauss-seidel method. Gauss-Seidel discretizes (3.5) using the Forward-Time Central-Space (FTCS) finite difference scheme. In one spatial dimension, the electric field in terms of the potential becomes

$$\mathbf{E}_g = \frac{\phi_g^{n+1} - \phi_g^{n-1}}{2\Delta x}. \tag{3.6}$$

Poisson's equation, equation (3.5), becomes

$$\frac{\phi_g^{n+1} - 2\phi_g^n + \phi_g^{n-1}}{\Delta x^2} = -\frac{\rho_g}{\epsilon}. \tag{3.7}$$

Where the subscript g implies the evaluation of ϕ at grid points, and the superscript p denotes the grid node index. Analogous expressions can be formed in two and three spatial dimensions.

3.2.3 Particle weighting

In the particle in cell method particles can exist anywhere in the continuous spatial computational domain, but forces and charge densities are calculated at discrete grid points (Birdsall et al. 2004, chapter 2.6). Historically, the NGP method and CIC method are used to weight computational particles to the grid. Higher order schemes exists, such as Quadratic Splines (QS) and Cubic Splines (CS), see (Okuda et al. 1979) for an overview on the usage of higher order weighting schemes in plasma simulation. In PINC the CIC method is used to weight particle parameters to the grid. Using similar notation as Verboncoeur (Verboncoeur 2005) the weighting function is defined as

$$\mathbf{w}_{i,j,k} = \mathbf{x}_p - \mathbf{X}_{i,j,k}. \tag{3.8}$$

Where $\mathbf{x}_p$ is the position of particle p, and $\mathbf{X}_{i,j,k}$ is the position of the nearest grid point closest to the origin. Using equation (3.8), the charge distribution of particle p to a two dimensional grid can be found as

$$\begin{aligned} Q_{i,j} &= q_p\,(1 - w_i)\,(1 - w_j) \\ Q_{i+1,j} &= q_p\,w_i\,(1 - w_j) \\ Q_{i,j+1} &= q_p\,(1 - w_i)\,w_j \\ Q_{i+1,j+1} &= q_p\,w_i\,w_j. \end{aligned}$$

With similar equations for the three dimensional case can be found from equation (3.8). From the charge distribution found above, the charge density can be found directly by dividing the charge distribution by the volume of a computational cell, i.e

$$\rho_{i,j,k} = \frac{Q_{i,j,k}}{V_{i,j,k}}.$$

From the charge density and current density, PINC computes the electric and magnetic field using the multigrid field solver module.

3.3 Simulation stability and constraints

3.3.1 Spatial resolution

In PINC, and in other particle in cell codes, particles move in continuous space, but their macroscopic properties are projected to a discrete grid. Representing continuous variables on a discrete grid leads to numerical instability called finite grid instability (Lapenta 2011, section 5.1.). The analysis of finite grid instability is beyond the scope of this thesis, a rigorous mathematical description of finite grid instability can be found in for example *Plasma physics via computer simulation* (Birdsall et al. 2004). The most important result of these analyses is that the grid spacing Δx must satisfy the condition

$$\frac{\Delta x}{\lambda_D} < C. \tag{3.10}$$

Where C is some constant dependent on the discretization scheme used. In the case of the CIC scheme, the constant C is approximately equal to π. Failure to meet this condition in a PIC simulation leads to unphysical heating of the plasma. Equation (3.10) must therefore be satisfied in all directions of the full computational domain to ensure conservation of energy.

3.3.2 Temporal resolution

In PINC, the boris algorithm is used for particle pushing. The boris algorithm is an explicit forward time integration scheme, and as such simulations run with PINC must satisfy temporal stability constraints associated with such schemes. A Von Neumann stability analysis of a harmonic oscillator without an external magnetic field given by the following equation (Hockney et al. 1988), (Birdsall et al. 2004), (Lapenta 2011)

$$\frac{q_s}{m_s}\mathbf{E}_p(\mathbf{x}_p) = -\Omega^2\mathbf{x}_p, \tag{3.11}$$

leads to the equation

$$\left(\frac{\Omega\Delta t}{2}\right)^2 = \sin^2\left(\frac{\omega_N\Delta t}{2}\right). \tag{3.12}$$

Where ω_N is the numerical oscillation frequency. For values outside the range [-1,1] the sine function has only complex solutions, thus for timesteps Δt where $\Omega\Delta t > 2$ is true, the numerical oscillation frequency will be complex. Any practical simulation of the harmonic oscillator will therefore become unstable due to unbounded numerical heating. Thus, the finite time stability criteria can be expressed as

$$\Omega\Delta t < 2. \tag{3.13}$$

In this thesis, for solar wind simulations, the frequency Ω that needs to be resolved is typically the electron oscillation frequency of the cold solar wind plasma.

3.3.3 The CFL condition

The Courant–Friedrichs–Lewy condition, or CFL condition, is a stability criteria linking the finite timestep and grid step, Δt and Δx respectively. The condition must be met everywhere in the computational domain if the explicit integrator in PINC is to converge to a solution.A common formulation of the condition is given as (Marholm 2019)

$$\frac{\Delta x}{\Delta t} > C. \tag{3.14}$$

Where C is some characteristic speed. The constant C is solver scheme dependent, and is often set as $C = 1$ for explicit schemes; in PINC this qualitatively means that a particle is restricted to move by maximum one computational cell per timestep as the length scale is normalized to the grid step size.

3.4 The PINC Object module

The object module in PINC contains a set of functions and data structures necessary for simulating objects immersed in plasma. It was developed primarily by Jan Deca and Sigvald Marholm as a part of a collaboration between the University of Oslo and University of Colorado Boulder. In the object module, spacecraft-plasma interations are simulated using the same Capacitance Matrix method outlined by Miyake and Usui in developing the PIC code EMSES (the ElectroMagnetic Spacecraft Environment Simulator) (Miyake and Usui 2009).

In this section the method used in PINC for defining objects on a structured Cartesian grid is discussed in brief; a description of the *Grid* data structure and associated functions in PINC is also introduced as all scalar and vector fields are represented in PINC using this data structure, including conducting objects which are defined as scalar fields. The capacitance matrix method is then outlined, with an emphasis on the equations that have been implemented in PINC.

3.4.1 Representing objects on a grid

In the PINC object module, conducting objects are represented as points on the Cartesian computational grid, and stored in an input HDF5 (Hierarchical Data Format) file. Internally in PINC, this file is converted to a PINC *Grid* structure instance and is stored as a subset of the *Object* data structure which additionally contains data on which nodes are internal to the object, which nodes are on the surface, and data required in photoemission computations. Required data for photoemission in the *Object* structure will be described in the later section on implementation of photoemission in PINC. The PINC *Grid* structure is a generalized data structure that stores N-dimensional data in a flat manner. The values on the grid is stored in a lexicographical manner.

3.4.2 The capacitance matrix method

The general idea of the capacitance matrix method is to pre-calculate a capacitance matrix for each object immersed in the plasma in question. Applying the computed capacitance matrix, the charges due to super-particles impinging the object can be redistributed to the surface of the object, after redistribution of charges a charge density and electric potential correction is computed and spacecraft potential can be updated. super-particles located within the object are then removed from the computational domain after their charges have been redistributed. This modifies the PIC computational cycle in figure Figure 3.1 slightly, adding steps for redistributing charges that impinge on the spacecraft and another iteration of the Multigrid solver to update the fields.

The capacitance matrix relates the potential ϕ and charge distribution ρ as follows (Miyake and Usui 2009)

$$\rho_i = \sum_{j=1}^{N_G} A_{ij}\phi_j. \tag{3.15}$$

Where the matrix A represents the capacitance matrix, where i and j are the indices of the grid, and N_G is the total number of grid points. Inverting matrix

A, and defining $B \equiv A^{-1}$, the potential ϕ can be calculated as

$$\phi_i = \sum_{j=1}^{N_G} B_{ij}\rho_j. \tag{3.16}$$

After redistribution of charges, the charge density on the conducting body surface ρ_s changes. The correction in potential on the spacecraft surface $\delta\phi_s$ can then be found from the charge density correction $\delta\rho_s$ as

$$\delta\phi_{s,i} = \sum_{j=1}^{N_B} B_{ij}\delta\rho_, \tag{3.17}$$

where N_B is the total number of surface nodes, and $N_B < N_G$. By forming a subset of matrix B with the values associated with the surface nodes, with N_B rows and columns, and inverting this matrix the charge density correction on the surface can be found

$$\delta\rho_{s,i} = \sum_{j=1}^{N_B} C_{ij}\delta\phi_{s,j}. \tag{3.18}$$

Where the matrix C is the inverted sub matrix of B. Qualitatively C is the capacitance matrix of the conducting surface. In the PINC object module, matrix C is constructed and stored before the main simulation loop begins by placing a unit charge on each grid point defining the conducting object surface, setting all other grid points to zero charge, and then solving for the potential: This forms the sub matrix of matrix B which can then be inverted to find the surface capacitance matrix C, details of this implementation are found in appendix **??**.

In the simulation time-loop of PINC, after the potential has been updated, but before charges have been distributed to the surface, the surface potential of the conducting body $\phi_{s,j}$ is not the same value for all values of j and is therefore not at an equipotential. The correction in potential in terms of the equipotential ϕ_c of the surface is given by

$$\delta\phi_{s,j} = \phi_c - \phi_{s,j}, \tag{3.19}$$

substituting equation (3.19) into equation (3.18), the charge density correction equation becomes

$$\delta\rho_{s,i} = \sum_{j=1}^{N_B} C_{ij}(\phi_c - \phi_{s,j}). \tag{3.20}$$

The equipotential ϕ_c is unknown in equation (3.20); by using the fact that charges must be conserved when impinged charges are redistributed to the object surface, the equipotential can be calculated. Conservation of charge takes the form

$$\sum_{i=1}^{N_B} \delta\rho_{s,i} = 0, \tag{3.21}$$

substituting equation (3.20) into equation (3.21), ϕ_c can be computed as

$$\phi_c = \frac{\sum_i \sum_j C_{ij}\phi_{s,j}}{\sum_i \sum_j C_{ij}}. \tag{3.22}$$

In the conducting object run mode of PINC, after a first pass of the multigrid Poisson solver, equations (3.20) and equation (3.22) are solved to redistribute absorbed charged particles, a new pass with the Poisson solver then solves the new potential field using the correction to the charge density field. Details on the implementation of these equations in PINC can be referenced by contacting the 4DSpace group at the University of Oslo for access to the PINC code repository.

3.5 Implementing photoemission in PINC

Photoelectric emission by sunlit spacecraft surfaces contribute significantly to the overall charge of a spacecraft. In fact, in normal conditions in geosynchronous orbits, the photoelectron flux exceeds that of the ambient electrons (Lai and Tautz 2006). When there is a net outgoing current flux, spacecraft charge to to a positive voltage. In order to simulate spacecraft charging using PINC, it is then necessary to implement photoelectric emission. In this section the photoemission method implemented as part of this thesis is presented.

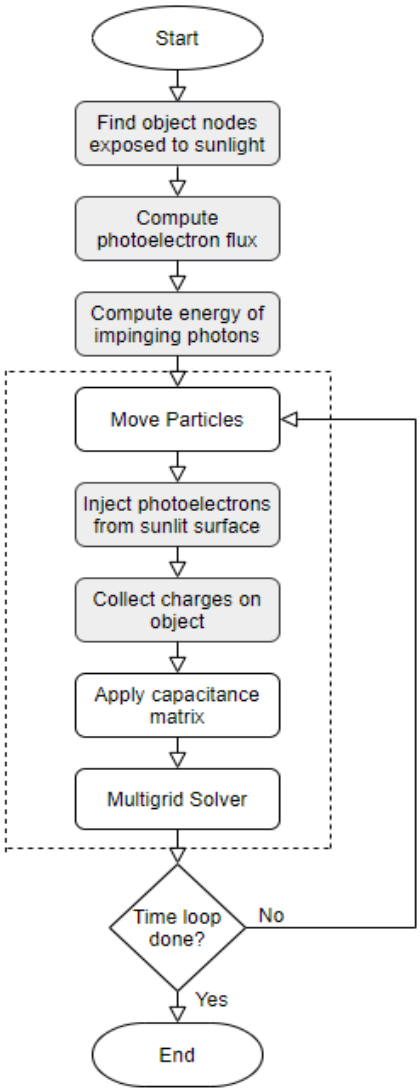


Figure 3.2: Flowchart describing the photoemission algorithm

The flowchart shown in figure 3.2 describes the general flow of PIC method as implemented in PINC when the photoemission modules developed for this

thesis are included. Processes shown in the diagram are on a function level in the code, with processes colored grey describing functionality that needed to be implemented in PINC. The details of the implementation of these processes are shown later in this section. The functions placed within the stippled box are carried out for each iteration of the PIC time loop; since these functions are called with a greater frequency, it is vital that they are computationally efficient.

In addition to the functions described in the figure 3.2, the initialization function of the object structure in PINC was expanded to read input parameters required for the new photoemission functions. To facilitate ease of use, and to make the code more easily readable, function pointers were added such that the functions for computing photoelectron flux and photoelectron injection could be specified as part of an input file.

3.5.1 Identifying sunlit surfaces

To be able to simulate photoemission, our code needs to know which surfaces the photoelectrons are supposed to be emitted from. As discussed in the previous section on the PINC object module, PINC has no "knowledge" on what faces make up the surface of an object embedded in the computational domain. Thus, it is necessary to locate the object nodes that are exposed to direct sunlight.

Two different functions were developed to find these surface nodes, which function to run is dependent on which algorithm selected for injecting photoelectrons into the computational domain.

The method for finding the surface nodes directly exposed to sunlight has been programmed with the assumption that photons travel in the same direction as the x axis of the domain to reduce the complexity of the code.

Beginning at the node with index 0 of the domain, the code steps along the x direction checking whether each node index is an element of the array containing all object surface nodes. When the node index matches the index of a surface node, that index is stored in the object structure, and the loop stepping along the x axis is exited. The loop of stepping down the x axis is then repeated for each node along the y and z axis until the function has iterated through each "column" normal to the YZ plane. If a surface node is not located in a column parallel to the x axis, the function simply continues to the next column.

This function can be visualized as analogous to mapping the topology of the bottom of a lake by sinking a weight attached to a rope from the surface of the lake, and measuring the wetted length of the rope when the weight reaches the bottom. Listing A.1 in Appendix A shows the basic structure of the function described above, and details how the code loops through the structured grid. The surface node array is initialized by a function that computes the object capacitance matrix, which is called before the sunlit surface nodes are found. In the photoelectron injection function, the indices of the sunlit nodes can then be converted into Cartesian coordinates with a helper function using the "sizeProd" array.

A similar function to listing A.1 finds the subset of object surface nodes that are used for photoelectron injection by filling cells adjacent to the emitting surfaces. Since PINC does not store data defining cells, and each node is surrounded by 8 computational cells, it is necessary to consistently pick which

cell to fill with photoelectrons. For a surface node with linear index p located at grid points $[i, j, k]$ we choose the cell to be filled by the nodes $[i-1, j, k], [i-1, j+1, k], [i-1, j, k+1], [i-1, j+1, k+1], [i, j, k], [i, j+1, k], [i, j, k+1], [i, j+1, k+1]$. In PINC we have defined the sunlit surface of the object by the projection of the object onto the YZ plane, therefore for a cell to be "on" the surface of the object we also require that the vertices $[i, j, k], [i, j+1, k], [i, j, k+1], [i, j+1, k+1]$ of the cell are also surface nodes.

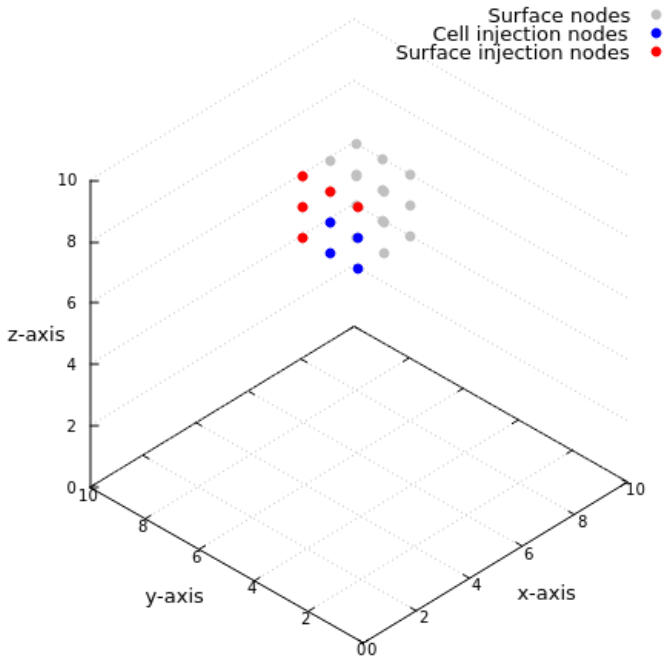


Figure 3.3: Surface nodes on a box exposed to sunlight

Figure 3.3 is a visual representation of why this condition must be met for nodes used for the cell filling injection algorithm. It shows the surface nodes of a 2 unit by 2 unit box inside a computational box of 10 x 10 x 10 nodes where the sunward direction is along the negative X axis. Nodes marked in blue are nodes used for photoelectron injection by cell filling, the surface injection algorithm uses the nodes colored **both** red and blue. Cells defined by nodes colored red are not adjacent to the sunlit surface in the XY plane.

Another condition is necessary to avoid selecting cells that are part of the interior of the object; we further stipulate that for nodes $[i-1, j, k], [i-1, j+1, k], [i-1, j, k+1], [i-1, j+1, k+1]$ a maximum of three are either surface nodes **or** interior nodes of the object. A C style pseudocode implementation of these conditions are given in listing A.2 in Appendix A; the "surfaceOffset" stores the number of surface nodes for object "a", then the defined cell vertices are stored. The linear index of these vertices are then compared against the "surface" array and "interior" array. We then check whether the cell has a face on the surface of the object, and that the cell is **not** an interior cell.

3.5.2 Photoemission current and photoelectron energy

Two methods have been developed in PINC for finding the temperature, and flux of photoelectrons from sunlit object surfaces. The first method has been previously discussed in chapter Chapter 2; integrating Planck's law over a frequency band, allows both the photoelectron flux and temperature to be calculated. For some objects, especially spacecraft, the photoelectric current, and the temperature of the emitted electrons are known a priori to our simulation. These values can then be specified as a input parameters to PINC. The number of super particles to inject per timestep into each cell adjacent to the sunlit surface is directly computed from the photoelectron current flux (Deca et al. 2013)

$$N_{inject} = j_{ph} \frac{dA \cdot dt}{q_{part}}. \tag{3.23}$$

Where N_{inject} is the number of super particles to inject into each photoemitting cell, dA is the area of a computational cell, dt is the simulation timestep, and q_{part} is the charge of one computational particle. The charge on one particle is calculated as

$$q_{part} = \frac{\rho \, e \, V_{cell}}{N_{p,cell}}, \tag{3.24}$$

where ρ is the electron density, V_{cell} is the volume of the cell, and $N_{p,cell}$ is an input parameter specifying the number of particles per computational cell. Equation (3.23) can be rewritten to calculate the total number of photoelectrons ejected per timestep. Since several photoelectron injection methods have been analyzed for this thesis, the total photoelectron flux per timestep is the form that has been implemented into PINC

$$N_{total} = \frac{j_{ph} \cdot A \cdot dt}{q_{part}}. \tag{3.25}$$

Where the total area of the sunlit object surface A has been used instead of the cell surface area. These particles are then injected into the computational domain with velocity sampled from a Maxwellian plasma with an average temperature given by the input parameters. Which method PINC uses for finding photoelectron flux is specified as a flag in the simulation input file.

3.5.3 Integrating Planck's law

The photoelectron flux of a sunlit object is dependent on the material composition of the sunlit surface. Surfaces with a lower work function tend to eject more photoelectrons. Similarly, the photoelectron flux of a sunlit surface is proportional to the distance from the surface to the sun. Since the power output of the sun follows an inverse square law with the square of the distance from the sun. The closer the spacecraft is to the sun in its orbit the higher the number of photons with energy greater than the work function strike the surface of the spacecraft. Both these factors need to be accounted for in order to make simulating spacecraft charging in sunlight as general as possible.

The number of photoelectrons emitted is however not equal to the number of impinging photons with energy higher than the surface work function. Both the reflectance of the surface, and the photoelectron yield of the surface material

significantly reduce the photoelectron flux. Reflectance and photoelectron yield are both dependent on the energy of incoming photons, but are assumed to be constant in this thesis. Thus, to compute the photoelectron flux for a particular spacecraft, Planck's law must be integrated in the frequency band from the work function frequency, to infinity before the main particle-in-cell loop is started. The photoelectron flux can then be directly computed from the power received by the spacecraft, as long as the surface area exposed to the sun and work function of the spacecraft material surface is known. These variables, in addition to the material photoelectron yield and reflectance, can be given as inputs to our simulation.

Integrating Planck's additionally allows the average kinetic energy of the emitted photoelectron to be calculated, dividing the total radiation energy absorbed per timestep by the total photoelectron flux. Several methods for numerically integrating Planck's law efficiently exists. In this thesis, the method settled on is based on the paper by Widger and Woodall (Widger et al. 1976). A overview of the method presented in that paper, and the implementation in PINC as C pseudocode will now be given.

First, to simplify the integrals, Planck's law is recast in terms of the wavenumber (cm^{-1}) ν rather than the frequency of radiation. Planck's law then takes the form

$$B(T) = C_1 \int_{\nu}^{\infty} \nu^3 \left(\exp\left\{ \frac{C_2 \nu}{T} \right\} - 1 \right) d\nu. \tag{3.26}$$

Where B(T) is the radiance, given in units of $Wcm^{-2}Sr^{-1}$, T is the absolute temperature in Kelvins, and C_1 and C_2 are the first and second radiation constants respectively. Letting $x = C_2 T^{-1} \nu$, equation (3.26) can be rewritten as

$$B(T) = C_1 C_2^{-4} T^4 \int_{x}^{\infty} \frac{x^3}{\exp\{x\} - 1} dx. \tag{3.27}$$

The integrand in equation (3.27) can be expanded in a series form, substituting that series into equation (3.27) and integrating by parts, equation (**??**) becomes

$$\int_{x}^{\infty} \frac{x^3}{\exp\{x\} - 1} dx = \sum_{n=1}^{\infty} \exp\{-nx\} \left(x^2 n^{-1} + 3x^2 n^{-2} + 6x n^{-3} + 6n^{-4} \right). \tag{3.28}$$

C_1, C_2 and T are known constants in equation (3.27), and the infinite sum in equation (3.28) is straight forward to compute to some finite tolerance using a C function. The integral in (3.28) is computed in units of radiance Wm^2Sr^{-1}, the solid angle Ω of the photoemitting object with respect to the sun can be calculated as

$$\Omega = \frac{A_s}{r^2}.$$

Where A_s is the area of the sunlit surface, and r is the radial distance from the center of the sun to the object. Multiplying the result of (3.28) by the solid angle of the object with respect to the sun, and the surface area of the sun, the power output in the frequency band above the work function of interest can be found. A similar integral to the one found in equation (3.28) can be constructed in terms of photons per second instead of Watts.

Computing the photons impinging on the surface per timestep does not however correspond to the amount of photoelectrons emitted by the sunlit surface. To compute the total photoelectron flux, the photon impinging rate must be multiplied by the surface material reflectance, and the photoelectron yield (number of electrons emitted per incoming photon above the work function of the material) as described in chapter 2.

Listing A.3 in Appendix A shows a numeric implementation of equation (3.27) and equation (3.28). The original algorithm (*Calculating Balckbody Radiance* 2020) has been converted from C++ code to C for compatibility with the PINC framework.

3.5.4 Injection of photoelectrons

The method selected for injecting a Maxwellian velocity distribution of electrons affect the accuracy of the results from a particle in cell simulation. Once the average kinetic energy of the photoelectrons has been calculated as seen in previous sections, a Maxwellian velocity distribution can be constructed from the average velocity.

The position of the particles must also be distributed into the computational domain in such a way that the particles remain Maxwellian in their velocity distribution. Describing an accurate particle distribution function $f(\mathbf{x}, \mathbf{v}, t)$ is especially important for simulations with timescales longer than that that of the ion transit time, and for simulations that include Monte Carlo collision modules (Cartwright et al. 2000).

In the code developed for this thesis, three methods for injecting photoelectrons were analyzed; two methods based on injecting electrons directly from the surface nodes exposed to sunlight, and one method in which electrons were uniformly distributed in the computational cells adjacent to the emitting surfaces.

Later in this chapter, these injection algorithms are tested against the charging simulations of the Solar Probe plus presented in (Deca et al. 2013).

Surface node injection

The simplest method for injecting the photoelectrons from a surface, is by defining all object surface nodes exposed to sunlight as particle sources. The position of each emitted particle is computed as an "push" based on the Maxwellian velocity distribution from the position of each node:

$$x_i = x_{\mathrm{node},i} + R\, v_i \tag{3.29}$$

Where $x_{i,j}$ denotes the position component in the i direction of the injected particle, $x_{node,i}$ denotes the position of some node in the i direction, R is some random number on the interval $[0, 1]$, and v_i is the sampled velocity component in the i direction of the injected particle.

Two methods for sampling the Maximilian velocity distribution were analyzed; the first was based on the Box-Mueller algorithm (Deca et al. 2013).

In the Box-Mueller algorithm, the tangential velocities are sampled from the bivariate Maxwell distribution:

$$f(v_{t1}, v_{t2}) = \left(\frac{m}{2\pi kT}\right) \exp\left(-\frac{m(v_{t1}^2 + v_{t1}^2)}{2kT}\right) \tag{3.30}$$

Whereas the normal component of the velocity is sampled from a half-Maxwellian distribution, described by the distribution function

$$f(v_n) = \left(\frac{m}{2\pi kT}\right)^{1/2} \exp\left(-\frac{mv_n^2}{2kT}\right) \tag{3.31}$$

The bivariate distribution is implemented in PINC by sampling a bivariate Gaussian distribution with a variation equal to the average thermal velocity of the photoelectrons. The normal component of the photoelectron velocity is sampled from a univariate Gaussian distribution, also with a variation equal to the average thermal velocity, and rejecting the negative velocities (pointing into the object).

The second node injection method implemented similarly distributed emitted particles as a partial push using equation (3.29). The method differs in how the velocity distribution of the particles were sampled. First, the speed of the electron is sampled from a univariate Gaussian distribution and the velocity of the particle is initialized as parallel and opposite to the direction of the sun. Then, a rotation matrix is formed from subsequent rotations about each axis perpendicular to the sunward direction. Each rotation angle is randomly sampled from the range $[-90°, 90°]$ to reduce the probability of the particle being injected into the object.

In PINC, the sunward unit vector is always defined as being in the negative x direction. Thus, the rotation matrix in PINC is implemented as two elemental rotations about the y and z axis in either order. The rotation tensor $\boldsymbol{R}$ is then expressed as

$$\boldsymbol{R} = \boldsymbol{Y}_\beta \boldsymbol{Z}_\gamma \tag{3.32}$$

and in the matrix form

$$\boldsymbol{R} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.33}$$

Where β, γ are the rotation angles about the y and z axis, and $\boldsymbol{Y}_\beta$, $\boldsymbol{Z}_\gamma$ are the elemental rotation tensors about the Y and Z axis respectively. The initial velocity vector is then multiplied by the rotation matrix, and the particles are placed according to equation (3.29).

The disadvantage of using this method is that it requires calling the function that generates the rotation matrix every time a photoelectron is emitted. This makes this method more computationally expensive than taking samples from the univariate and bivariate Gaussian distributions. However, the photoelectrons produced by the first method could potentially have higher average kinetic energy than what is physically correct, since the two Gaussian distributions are sampled independently in the first method.

Filling cells adjacent to sunlit surfaces

This method is based on the same method utilized in the particle-in-cell code iPic3D by Deca et al. (Deca et al. 2013); photoelectrons are distributed in a uniform random location in each computational cell adjacent to the emitting surface of the object. Based on the results from verification simulations, this cell filling algorithm was the method ultimately selected when simulating charging of the BepiColombo spacecraft MMO. In PINC, unlike iPic3D, objects are defined by the nodes on the computational grid rather than the computational cells. The first step in PINC is then to identify the corner nodes of the cells adjacent to the emitting surface, the pseudocode found in listing A.2 in Appendix A shows how these corner nodes are identified. Each photoelectron is placed in a cell by adding a random value in the range $[0, 1]$ to each position component of the corner node.

$$x_p = x_n - R \tag{3.34a}$$

$$y_p = y_n + R \tag{3.34b}$$

$$z_p = z_n + R. \tag{3.34c}$$

Where the subscript p denotes the p'th photoelectron and the subscript n is used for node positions. The random number R is sampled independently for each position component, and the range $[0, 1]$ is used since distance is normalized by stepsize in PINC. The negative sign in subequation (3.34a) stems from the fact that in PINC the sunward direction is always along the negative x axis.

The total flux of photoelectrons is divided by the number of cells adjacent to the sunlit surfaces of the object, each cell is then filled iteratively with this amount. Listing A.4 in Appendix A shows how this method was implemented in C style pseudocode. The velocity of a particle is sampled in the surface normal direction and the tangential direction. Then a random position within the cell is computed as an offset of the position of the corner node that defines the cell, the electron is then stored in the particle "population" structure with its new velocity and position. The next cell corner is then selected and the process repeats until the total amount of photoelectrons in the timestep has been created.

3.6 Photoemission verification simulation

Before we can run any analysis on the charging behavior of MMO in its orbit around mercury, it is necessary to verify the accuracy of our implementation of photoemission. The simulations described in the paper (Deca et al. 2013) provides an excellent test case for our model. In their paper Deca et al. analyze different charging mechanisms, including photoemission, affecting a cubesat. The charging of the cubesat is simulated using the PIC codes iPic3D and PTetra.

Using a computational setup similar to Deca et al. I ran the experiments that included ambient plasma charging and photoemission, and used the floating potential of the cubesat found using iPic3D and PTetra as benchmarks for the accuracy of the code I had developed. The formation of wakes downstream from the cubesat, as well as the formation of a photoelectron cloud in front of the sunlit surface of the cubesat affects the floating potential of spacecraft (Miloch 2015) (Sjögren et al. 2012). The presence of both of these phenomena are therefore quantitatively compared with results found in the iPic3D and PTetra simulations to further verify my photoemission implementation in PINC.

The simulations were carried out for all three of the photoelectron injection methods described in section 3.5.4, the total photoelectron flux was computed using the current density method described in section 3.5.2 as the current density is a known variable in (Deca et al. 2013).

3.6.1 simulation setup

The simulation parameters described in (Deca et al. 2013) were used mostly as presented. However, the verification test case was also used in the development stage for testing smaller parts of the new photoemission code: In the PINC verification simulations then, slightly larger computational cells were used, and the number of computational super-particles were reduced. This greatly reduced the total time required for each simulation to complete, and allowed for rapid iteration on the development of the new code.

All verification tests were run on the supercomputer "Saga" using domain decomposition to run PINC in parallel on 16 cores. 16 cores were used since at the time of writing, PINC suffered from segmentation errors when running the code with 64 or more cores.

The cubesat was modeled as a cube measuring 1 m x 1 m x 1 m placed in the center of a 10 m x 10 m x 10 m computational box. The resolution of the grid was 0.15625 m in each direction of the computational box.

Δt	5e-9 s
timesteps	10,000
Computational box	10 m x 10 m x 10 m
Resolution	$\frac{10}{64}$ m x $\frac{10}{64}$ m x $\frac{10}{64}$ m
Particles per cell	10 per species
Cubesat size	1 m x 1 m x 1 m

Table 3.1: Numerical parameters used in verification simulations

$\frac{m_i}{m_e}$	1836
$T_e(eV)$	85
$T_i(eV)$	82
$v_{th,e}(m/s)$	3.86×10^6
$v_{th,i}(m/s)$	8.87×10^4
$n_e(m^{-3})$	7×10^9
$n_i(m^{-3})$	7×10^9
$v_{sw}(m/s)$	3×10^5
$j_{ph}(A/m^2)$	1.6×10^{-2}

Table 3.2: Plasma parameters used in verification simulations

The complete list of numerical parameters and plasma parameters used in the verification simulations are found in table 3.1 and table 3.2 respectively. The plasma parameters found in table 3.2 are the exact same as used in (Deca et al. 2013), and are reproduced here for ease of reference.

The numerical parameters were changed to make simulation run in PINC numerically stable. The timestep Δt used in our verification simulation is smaller, this allowed for better convergence at small tolerances for the multigrid solver used in PINC. The number of timesteps used in the PINC simulations was selected to allow enough time for the cubesat to reach a steady-state potential. Furthermore, iPic3D uses an implicit solver (Deca et al. 2013) which is numerically stable over larger timesteps as compared to the explicit Boris solver used in PINC.

The direction of plasma drift and the direction of the sun is also different in our case; in PINC the sunward direction is always pointed along the negative X axis. The solar wind drift was then picked to flow in the positive Y direction to maintain the 90° angle between the drift direction and the sun as selected by Deca et al.

3.6.2 Floating potential of the cubesat

In PINC the potential of the cubesat is computed each timestep using equation Equation (3.22) and output to a log file, plotting this potential versus time allows us to see the development of the floating potential of the cubesat

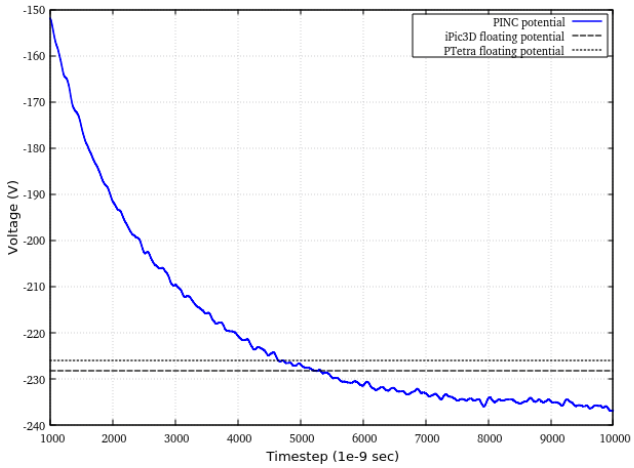


Figure 3.4: Plot of the cubesat potential over time, in this case no photoelectrons are included and the cubesat is only charged by the ambient flowing plasma. The stippled horizontal lines show the floating potentials obtained by iPic3D and PTetra for the same simulation setup

Figure 3.4 shows a time series plot of the cubesat potential relative to the upstream plasma defined as ground. As the object charges over time, and the potential becomes more and more negative, more electrons are diverted away from the cubesat thereby slowing the rate of charging. The two horizontal lines in the figure show the computed floating potential found by iPic3D and PTetra; the floating potential found using PINC is more negative, and accurate to both results found by iPic3D and PTetra within a 4% margin. Increasing the number of computational particles would reduce this inaccuracy, additionally due to the dimensions of the computational cells, the cubesat simulated in PINC has a smaller volume than the $1m^3$ cubesat simulated by Deca et al. leading to further inaccuracies.

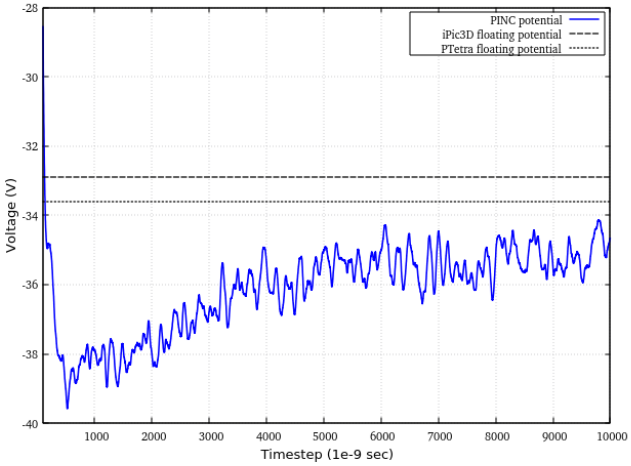


Figure 3.5: Plot of the cubesat potential over time, in this case photoelectrons are injected into the domain using the adjacent cell filling method. The stippled horizontal lines show the floating potentials obtained by iPic3D and PTetra for the same simulation setup

Figure 3.5 shows the timeseries of the cubesat potential when photoemission using the cell filling algorithm is included. The potential stabilizes after approximately 5,000 timesteps before it begins to oscillate steadily with an amplitude of approximately 1 Volt around a steady state value. The first hundred timesteps are cut from the plot, to better show this oscillation. Comparisons of the cubesat potentials for the different injection algorithms are averaged over time after 5,000 timesteps to remove these fluctuations and give a better sense of the accuracy of each algorithm relative to the floating potentials found with iPic3D and PTetra.

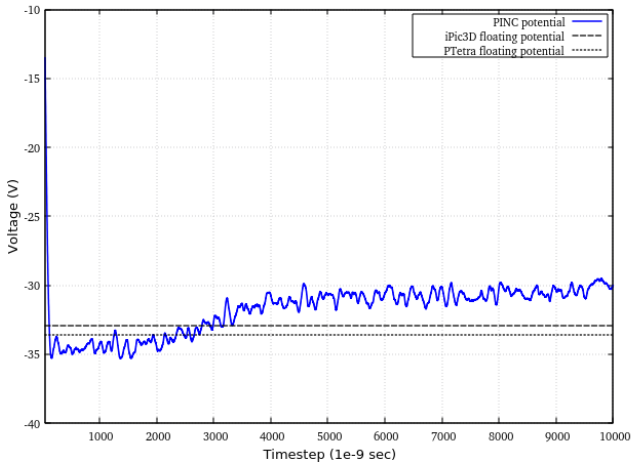


Figure 3.6: Plot of the cubesat potential over time, in this case photoelectrons are injected into the domain using equation (3.29) with velocity sampled from the univariate Maxwellian distribution and rotated using equation (3.33). The stippled horizontal lines show the floating potentials obtained by iPic3D and PTetra for the same simulation setup

The timeseries plot of the cubesat potential in Figure 3.6 includes ambient plasma charging, and photoelectron charging where photoelectrons are emitted using the rotation matrix method. The oscillations around a steady state value are of the same amplitude as in figure 3.5, but a less negative floating potential is predicted. Since the rotation matrix injection algorithm is based on a particle push of the sampled particle velocity distribution, this might be explained due to a higher concentration of photoelectrons near the cubesat surface as opposed to in the cell filling case. With a higher local density of electrons, the forces between the photoelectrons will be higher, thus imparting enough acceleration on a large enough mass of photoelectrons such that they pass the potential barrier formed by the photoelectron cloud. The higher number of "escaped" photoelectrons that are not reabsorbed by the cubesat then cause a less negative floating potential.

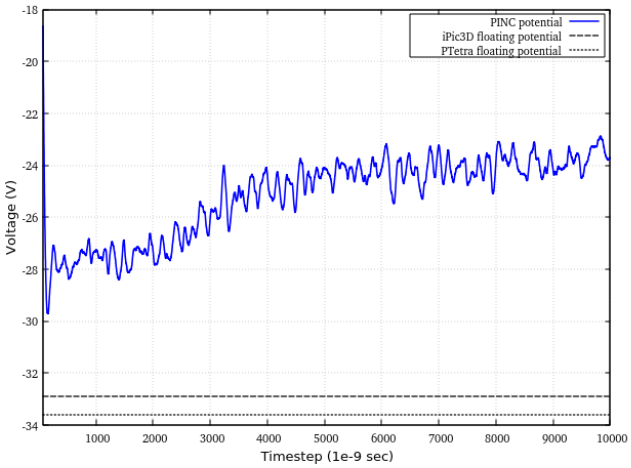


Figure 3.7: Plot of the cubesat potential over time, in this case photoelectrons are injected into the domain using equation (3.29) and sampling the surface normal and tangential velocities separately. The stippled horizontal lines show the floating potentials obtained by iPic3D and PTetra for the same simulation setup

Figure 3.7 shows the timeseries plot of the cubesat potential where photoelectrons are injected into the computational domain by a particle push based on sampling a univariate and bivariate Maxwellian velocity distribution. This injection algorithm diverges by far by the largest amount from the floating potentials found using the iPic3D and PTetra codes. The large difference in floating potential, and the floating potential being the least negative, suggests photoelectrons injected with this algorithm have higher kinetic energy than photoelectrons injected using the two other algorithms. It could also be the case that higher densities of photoelectrons resulting from the particle push tangentially away from the surface, force more electrons away from the cubesat that are then not reabsorbed.

Photoelectron injection method	Floating potential	Percentage difference from iPic3D	Percentage difference from PTetra
No photoelectrons	-233.242	2.21 %	3.20%
Cell filling	-35.31	7.33 %	5.09 %
Rotation matrix	-30.61	6.96 %	8.90 %
Double gaussian sample	-24.03	26.96 %	28.48 %

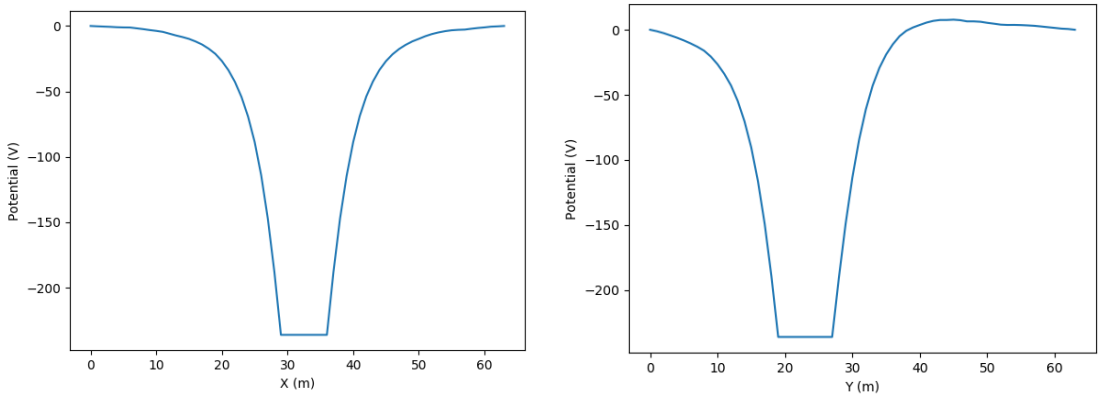
Table 3.3: Comparison of floating potentials found with iPic3D and PTetra, with the average potential of the cubesat from timestep 5,000 to timestep 10,000

Table 3.3 is a summary of the floating potentials computed using the injection algorithms I developed for this thesis. The "No photoelectrons" case give a baseline for differences in object charging when comparing PINC to iPic3D and PTetra. There is overall good agreement with the three codes in this simplest case, with only a 2.21 % absolute difference in the floating potentials found between PINC and iPic3D.

These errors can be reduced by decreasing the tolerance of the solver used to compute the capacitance matrix in PINC, and reducing particle noise by increasing the number of computational particles introduced in the domain, at the cost of longer simulation times.

Both the cell filling and rotation matrix injection algorithms are fairly close to values found by deca et al. when ambient plasma charging and photoemission are both included. The difference in computed floating potentials could again be reduced by setting a stricter tolerance for the multigrid solver in PINC, and increasing the number of computational particles used in the simulations.

3.6.3 Photoelectron cloud and wake formation



(a) Potential along x axis

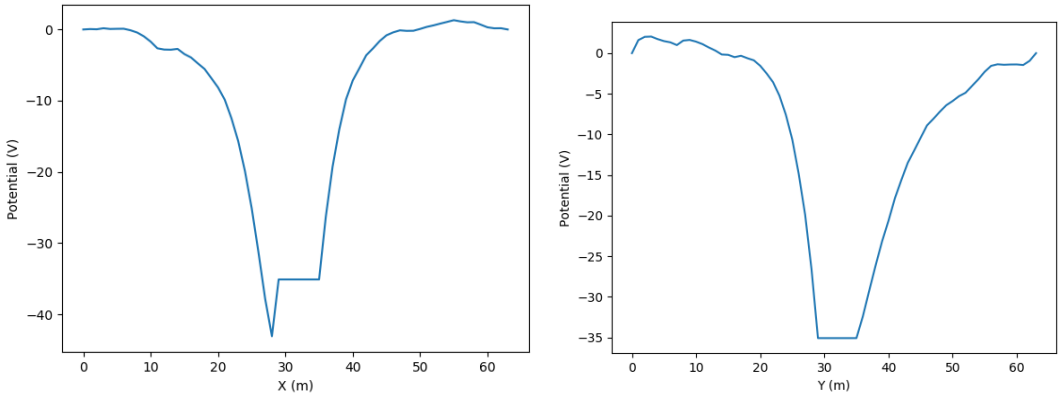
(b) Potential along y axis

Figure 3.8: Potential profile along direction of flow (+Y axis), and along direction of light travelling from the sun (+X axis) when photoemission is omitted from the simulation, each line passes through the centre of the cubesat

In figures 3.8a, and 3.8b we have plotted the potential in Volts along the axis of drift, and along the axis of the sun through the centre of the cubesat at timestep 10,000. The spatial X-axis of each plot is multiples of the computational cell step size, in this case 0.15625 meters.

The only charging mechanism included is the charging due to the ambient flowing plasma and is included as a baseline test to check the implementation of the capacitance matrix method in PINC and so that differences due to the photoemission implemented in this thesis may be singled out from the rest of the implementation of PINC.

No potential barrier can be seen in figure 3.8a, and the plot is symmetric about the centre of the cubesat. In figure 3.8b the potential relative to the upstream plasma rises above 0 behind the cubesat as expected due to the formation of an ion wake.

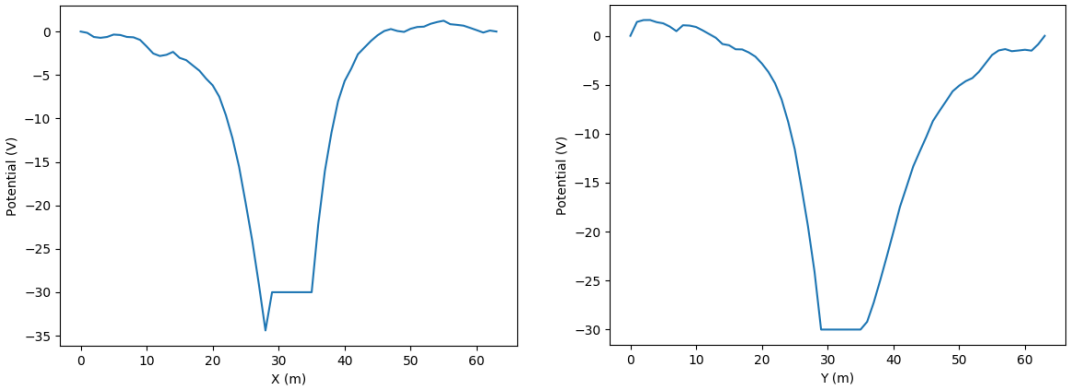


(a) Potential along X axis

(b) Potential along Y axis

Figure 3.9: Potential profile along direction of flow (+Y axis), and along direction of light travelling from the sun (+X axis) where photoelectrons are injected into the domain by randomly distributing the electrons uniformly into the cells adjacent to the emitting object surfaces, each line passes through the centre of the cubesat

Figure 3.9a and 3.9b include photoemission as a charging mechanism, where the cell filling algorithm in section Section 3.5.4 has been used to inject photoelectrons. Figure 3.9a clearly shows a potential barrier forming in front of the object where the photoelectron cloud has formed. Figure 3.9b, unlike figure 3.8b, shows no ion wake forming behind the cubesat along the drift axis, likely due to photoelectrons reflecting off the potential barrier and drifting behind the cubesat.

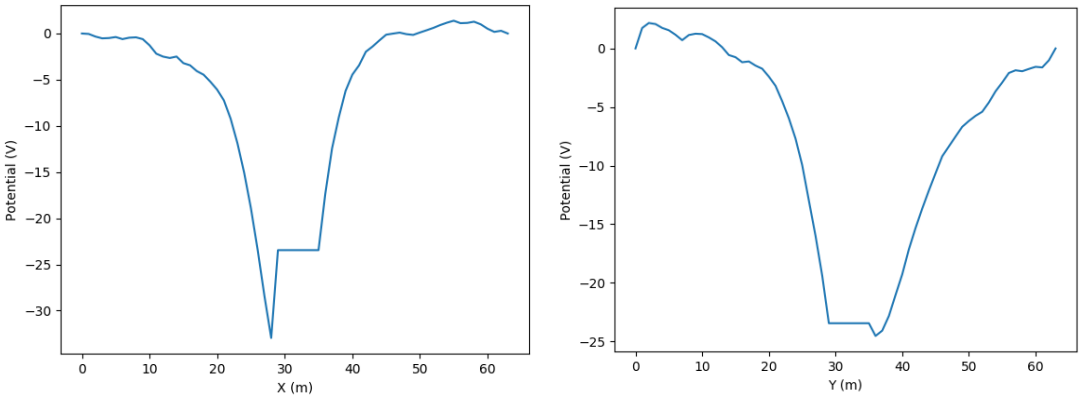


(a) Potential along X axis

(b) Potential along Y axis

Figure 3.10: Potential profile along direction of flow (+Y axis), and along direction of light travelling from the sun (+X axis) where photoelectrons are emitted using the rotation matrix algorithm and a partial particle push from object surface, each line passes through the centre of the cubesat

In figures 3.10a and figure 3.10b photoelectrons are injected by a partial particle push. The surface normal velocity is sampled from a univariate Maxwellian velocity distribution, then rotated by a random rotation tensor before the particle is pushed from the surface based on the computed velocity vector. The potential barrier in figure 3.10a is smaller than found in figure 3.9a, possibly due to more photoelectrons managing to escape the forming photoelectron cloud based on the random particle push, or due to the size of the computational cell: Refining the computational grid, and reducing the volume of the computational cells would likely decrease the difference in the potential barrier formed for the two cases.



(a) Potential along X axis

(b) Potential along Y axis

Figure 3.11: Potential profile along direction of flow (+Y axis), and along direction of light travelling from the sun (+X axis) where photoelectrons are injected into the domain by a partial, random, particle push. Surface normal velocity and tangential surface velocity is sampled separately. Each line passes through the centre of the cubesat

Figures 3.11a and 3.11b plot potential profiles where photoelectrons are emitted similarly to the rotation matrix algorithm, but where the three dimensional particle velocity vectors are sampled in both the surface normal and tangential directions separately. Figure 3.11a shows the deepest potential well out of all the injection algorithms, but the least negative floating potential.

A potential reason for this, is that the double sampling method increases the average kinetic energy of the emitted photoelectrons to a point where not all the photoelectrons return to the surface of the object and become reabsorbed. Figure 3.11b also shows another small barrier directly behind the cubesat along the direction of drift.

This suggest that particles pushed based on the two samplings of velocity have higher tangential velocities in the photoelectrons than when injected using the other two algorithms, causing electrons to accumulate downstream of the object. Based on the relative accuracy as shown in table 3.2 when compared to the two other PIC codes, an accurate reproduction of the expected physics, and the lower computational complexity as compared to the rotation matrix

algorithm, I selected the cell filling algorithm as the injection algorithm to use when simulating MMO in its orbit around Mercury.

3.7 Mercury Magnetospheric Orbiter simulation setup

The MMO is one of the two spacecraft making up the joint European Space Agency (ESA) and Japan Aerospace Exploration Agency (JAXA) BepiColombo mission. The principal scientific goals of the MMO spacecraft is to investigate the plasma environment around Mercury as well as the interaction between the planets intrinsic magnetic field and the solar wind (Benkhoff et al. 2009) (Saito et al. 2010).

Among the scientific instruments carried onboard the MMO, of special relevance to this thesis is the Mercury Plasma Particle Experiment (MPPE): The MPPE instrument package carries sensors for measuring plasma electrons and ions, as well as high energy particle sensors (Saito et al. 2010). These sensors are highly sensitive to variations in the plasma density around the device, thus simulating how charging effects of the MMO spacecraft affects density measurements of these sensors is an important aspect of analyzing the data MMO will send back to Earth.

In this section, I outline aspects of the design of MMO relevant to an accurate simulation of charging phenomena, the plasma and numerical parameters used in the simulation, as well as outline the different numerical experiments that were run as part of this thesis to investigate the charging behaviour of the spacecraft while in orbit around Mercury.

3.7.1 MMO design

The basic geometric shape of the MMO spacecraft is that of an octagon that can be surrounded by a 1.8 meter circle. The height of the octagon is 0.9 meters, with an upper section covered in both solar cells and second surface mirror (SSM), and the lower section only covered by SSM. The plasma particle instruments are situated inside this octagon, with their sensors placed on the surface panels of the octagon. Two 5 meter long booms extend from opposite sides of the spacecraft on which flux gate magnetometers are placed (Yamakawa et al. 2008).

With these dimensions in mind, the MMO can be approximated by stacking voxels. A trade off must be made between the number of voxels used. A high number of voxels increases the accuracy of the object model, but requires longer compute times as the computational cell size decreases. For plasmas with relatively low densities, like the plasma found within Mercury's magnetosphere, coarser approximations with lower number of voxels is possible since the debye length is often significantly longer than the characteristic length of the spacecraft.

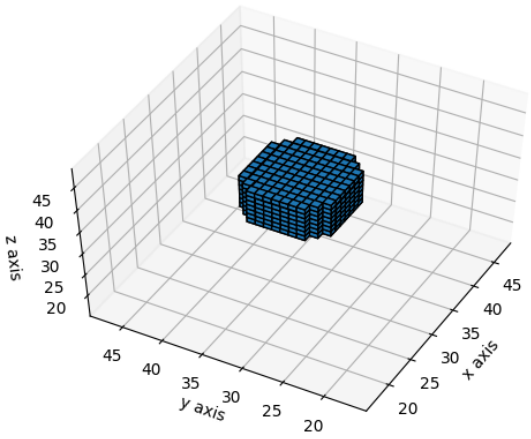


Figure 3.12: Voxelized approximation of MMO without booms in a 10 meter cubical computational domain with 64 grid points along each dimension. The plot ranges have been reduced to better display individual voxels

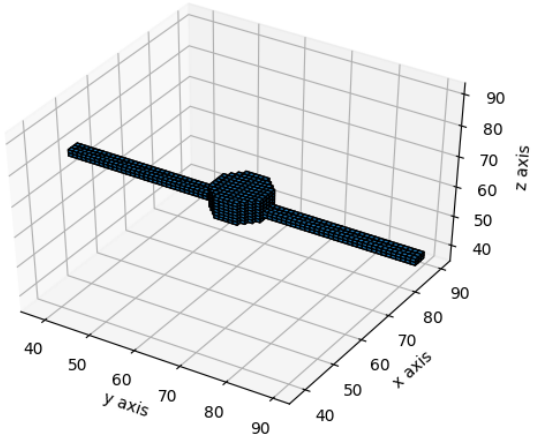


Figure 3.13: Voxelized approximation of MMO with booms in a 10 meter cubical computational domain with 128 grid points along each dimension. The plot ranges have been reduced to better display individual voxels

Figures 3.12 and figure 3.13 show two voxelized approximations of the MMO spacecraft, with and without the booms carrying the flux gate magnetometers. The high gain antenna and the wire antenna used to measure the electrical field have been discarded. These features are very thin, and would therefore increase the number of computational cells in the global domain to an untenable size. Both voxelizations were created by first constructing the objects in the CAD software suite Fusion 360, meshing the resulting object, and finding the voxels of the resulting mesh using a flood filling algorithm. The final voxelized version of both versions of the MMO used 8 cells over the diameter of the octagon shape of the MMO body. This allowed the shape of the spacecraft and the thickness of the booms to be resolved adequately, and allowed for a computational domain of several Debye lengths while keeping the total number of computational cells low enough for a full simulation to complete within a few days with parallelization.

3.7.2 Magnetospheric and numerical parameters

Two spacecraft have visited Mercury at the time of writing. The first spacecraft to do so was the Mariner 10 spacecraft, performing three fly-bys of the planet, equipped both with plasma detectors and magnetometers to study the properties of Mercury's magnetic field and its interaction with the solar wind plasma (Slavin et al. 2012). A longer observation mission, the MESSENGER spacecraft, entered into orbit around Mercury as the first spacecraft to do so. It orbited the planet from 2011 to 2015 in a highly eccentric orbit, collecting data all the while on the plasma conditions of the planets magnetosphere, magnetotail and the intrinsic magnetic field of the planet.

In this thesis, I have used a combination of data gathered from the MESSENGER mission, as well as MHD simulations of solar wind interaction with the magnetosphere of Mercury, to simulate the charging of the MMO spacecraft. Several assumptions were made in carrying out this analysis; first of all, the MMO spacecraft is spin-stabilized (Yamakawa et al. 2008) to reduce thermal load with its spin axis aligned along the magnetic dipole axis of Mercury. Because of the small timescales of charging relative to the rotational rate of the spacecraft, I have assumed an irrotational body fixed reference frame for the PINC simulations. Furthermore, MMO will orbit Mercury in a polar orbit with a period of 9.3 hours (ESA 2020b). As such, I chose to simulate the charging of the spacecraft close to the periapsis of MMO's orbit. Specifically, when the spacecraft is over Mercury's magnetic south pole where the plasma density is relatively high and exposed to direct sunlight. The distance of Mercury to the sun was also assumed to be the same as used in (Benna et al. 2009) such that the photon flux experienced by the MMO can be computed accurately.

Parameter	Value
$Timesteps$	10,000
dt	$5 \times 10^{-9} (10^{-8})$
dx, dy, dz	0.225 m
n_x, n_y, n_z	128
n_0	$1 \times 10^8 m^{-3}$
m_e	$9.10938356 \times 10^{-31}$ kg
m_i	$1.67262192 \times 10^{-27}$ kg
$\mathbf{B}_0$	[-8.8, 0, -99.6] nT
v_{drift}	1×10^5 m/s
$v_{th,e}$	4193011.62 m/s
$v_{th,i}$	87538.91 m/s
$n_{p,com}$	50 pc
r_{sc}	$5.6847 \times 10^{10} m$
$\overline{Y}_{ph}$	1×10^{-3}
A_{ph}	1.62 (3.87)
W	4.2 eV

Table 3.4: Magnetospheric plasma parameters, MMO specific parameters, and numerical parameters used in MMO simulations

Table 3.4 summarizes all relevant parameters used in the charging simulation of the MMO spacecraft. The computational box measures $28.125m$ along each axis. With the plasma parameters given above, applying equation (2.30) gives a debye length of $\lambda_D = 7.43m$. This means the computational box is approximately 3.79 Debye lengths in each direction. Ideally, we would use a larger computational domain, around λ_D, to avoid disruptive effects in the electric field along the boundaries. The size of the computational domain used was a tradeoff between the ability to resolve the MMO to sufficient accuracy, the total computational time available to us on the Saga supercomputer, and reduction of undesirable boundary effects.

The plasma density and thermal velocity values were extracted from the MHD simulation plots given in the paper by Benna et al. modelling the magnetosphere of Mercury at the first MESSENGER flyby. The intrinsic magnetic field of Mercury is dominantly dipolar, with a dipole tilt to the rotational axis constrained to being less than 5° (Anderson et al. 2010). Mercury's rotational axis is tilted 2 minutes of arc with respect to its orbital plane (Rothery 2015), and has a external field strength of about $100nT$ at the north cusp (Anderson et al. 2010). Using the maximum dipole axis tilt angle, the value for the magnetic field $\mathbf{B}_0$ in table 3.4 is then given in the Mercury solar orbital (MSO) coordinates.

The distance from the MMO to the sun r_{sc} was computed using the Jet Propulsion Lab (JPL) ephemeris system HORIZONS and setting the time of the ephemeris to the day of the first MESSENGER Mercury fly-by.

The parameter $\overline{Y}$ is the photoelectron yield per incoming photon and averaged over the range of photon energies that cause photoemission for MMO. Here we have assumed the MMO to be covered by Multi Layer Insulation (MLI), a thermal insulation covering consisting of several layers of plastic coated with an

outer layer of Aluminium. The average photoelectron yield of Aluminium and its work function were found in (Feuerbacher et al. 1972), the yield parameter used is dependent on the reflectance of the Aluminium, and as such the reflectance is not included as a separate parameter.

The SSM panels that cover the MMO are composed of silver films which have a slightly higher, but similar, work function to that of Aluminium. The parameter A_{ph} denotes the sunlit area of the MMO with and without the magnetometer booms. Since the booms account for a significant part of the sunlit surface area of the spacecraft the properties of Aluminium was used for simulations both with and without the booms.

3.8 Data analysis tools

The PINC framework already includes several Python scripts for analyzing 2D profiles of plasma density, potential, and the electric field. Tools also exist for analyzing the probability distribution of simulation particles. These tools use the numerical computation libraries Numpy, Scipy and Matplotlib to convert values and create readable plots. These tools were built by Jan Deca, Sigvald Marholm, Vigdis Holta, Steffen Brask og Gullik Killie. I modified several of these tools to be specific for my MMO simulations, nevertheless, all credit goes to the original developers. Tools written entirely by myself are mentioned, and some of the more comprehensive ones are expanded upon in Appendix A and Appendix C.

CHAPTER 4

Results

The charging behaviour of a spacecraft is dependent on several factors; the material composition of the craft, its dimensions, its location in space, the local time, and the space weather (Lai 2019). In this thesis, I wished to investigate what effects the presence of MMO's booms have on its charging behaviour, additionally I wished to investigate whether the direction of the ambient plasma drift or the photoelectron temperature would significantly change how the spacecraft charges.

The booms on the MMO extend the characteristic length of the spacecraft to be longer than that of the Debye length of the plasma found in table 3.4, as such I expect there to be a difference in the thickness of the plasma sheaths formed.

Varying the direction of drift may impact the convection of photoelectron away from the surface of the sunlit surface of the spacecraft, which could potentially lead to a difference in the floating potential. Varying the average energy of the emitted photoelectrons will change the charging behaviour of the MMO, as seen in the Rosetta charging simulation paper (Sjögren et al. 2012). To observe to what extent the floating potential and the thickness of the plasma sheath changes is the goal of varying the photoelectron temperature.

	No photo-electrons	drift along $+X$ axis	drift along $-Z$ axis	Inclusion of booms	Mercury magnetic field	Photoelectron temp. 3 eV
Case 1	X	X				
Case 2		X				
Case 3			X			
Case 4			X		X	
Case 5			X			X
Case 6	X	X		X		
Case 7		X		X		
Case 8			X	X		
Case 9			X	X	X	
Case 10			X	X		X

Table 4.1: Summary of the numerical experiments of the MMO in orbit around Mercury, carried out with PINC

Table 4.1 gives an overview of the numerical simulations of the MMO spacecraft carried out for this thesis, case 1 and case 5 give a baseline of the charging behaviour where no photoemission is included and serves as a point of comparison between the photoelectron current and the ambient electron plasma current the MMO is subjected to.

Cases 1 to 5, and cases 5 to 10 are equivalent but for the inclusion of the booms, with case 9 being the closest to the true conditions the MMO will experience when its orbit passes over the north cusp (ecliptic north) of Mercury. As such, case 8 is used as the baseline for the simulation in which the photoelectron temperature is varied from the value computed by PINC.

Save for cases 1 and 6, the simulations without photoemission, the computational domain was decomposed into 64 sub-domains for parallel processing. Cases 1 and 6 were decomposed into 128 sub-domains, the reason being that in the cases containing photoemission, the domain was not divided along the x-axis thus splitting the spacecraft in the YZ plane. This would account for another surface interior to the spacecraft that would have to be discarded as a source for photoemissive nodes.

For the sake of brevity, we will refer to the case number as shown in table 4.1 when presenting analyses for these five experiments. Results will be presented in order such that the plasma conditions closest to those the MMO will actually experience when orbiting Mercury is presented as the last simulation where photoemission parameters are computed directly. Photoemission with a different electron temperature is also presented, since an assumption of constant photoelectron yield was made, the impact of a higher photoelectron temperature will be compared to the other experiments in which the average photoelectron temperature was computed using Planck's law. We have also chosen to group simulations with and without booms together to make comparison across the two configurations simpler.

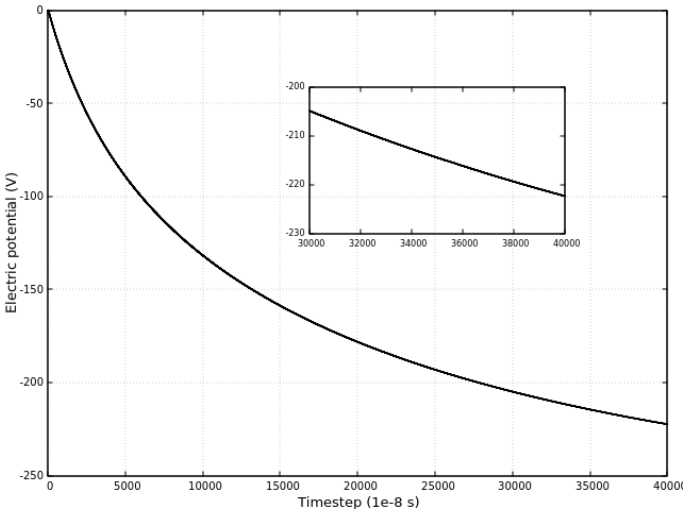
Except for a computational charging analysis of the combined BepiColombo spacecraft under thrust (using the MTM, the Mercury Transfer Module) (*ESA Science & Technology - BepiColombo plasma simulation* 2020), we were unable to find similar numerical experiments of the charging of the MMO spacecraft. We will therefore compare our results to theory through current balance analysis as presented in 2. We will also compare our results qualitatively to numerical charging experiments for different plasma parameters and different objects. Such as in the paper by Deca et al. (Deca et al. 2013) that was used for our verification simulations.

From current balance, with photoemission included, we expect the variation of drift direction (comparing cases 2 and 3 to cases 7 and 8) to have minimal impact on the floating potential PINC will converge to. Similarly we expect the sheath thickness and sheath plasma densities to be comparable for different drift directions. Across all five simulation types run, we expect the floating potential and sheath thickness to be larger in magnitude when the booms on the MMO are fully extended. Since the booms are modeled as photoemissive, the greater surface area will lead to an overall higher electron current flowing from the spacecraft leading to a higher floating potential.

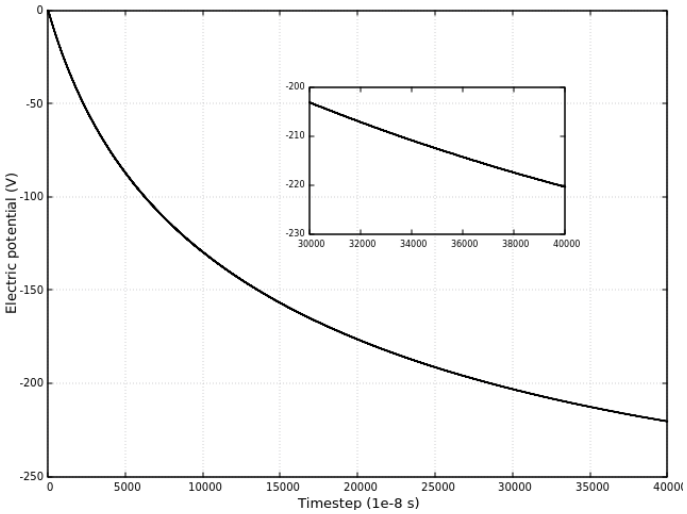
The inclusion of the external magnetic field of Mercury could potentially impact the fraction of photoelectrons with a greater than average temperature

being able to leave the surface of the spacecraft, if the particles' velocity vector does not align with the magnetic field the force exerted by the magnetic field could trap electrons more efficiently, leading to a lower floating potential.

4.1 Charging without photoemission



(a) Booms



(b) Without booms

Figure 4.1: Timeseries plot of potential of the MMO with and without booms. The potential has been converted from PINC dimensionless units to Volts. The inset plots shows the potential of the two configurations for last 10,000 timesteps.

The timeseries in fig 4.1 plots the potential of the MMO for each timestep simulated. From each inset plot, both for the MMO with booms and without, it is apparent from the slope that the system has not converged to a floating potential. Since the ambient plasma density around the MMO were approximately 100 times less dense than the value used in the verification simulation and no other current than ambient particle charging was present, a larger timestep was used in addition to a longer total simulation time.

In order to compare our results with theory, we can estimate the floating potential the simulation would have converged to by fitting a function to our data and propagating the function forward in time until a steady state has been reached. Using Hill's function of the form

$$\phi(t) = \phi_f \frac{t}{h+t},$$

where ϕ_f is the MMO floating potential without a photoemission current present, we estimate a floating potential of $\phi_f \approx -283V$. A full analysis of the curve fitting and the Python script used to fit the curves is given in Appendix C.

From chapter Chapter 2, we know that we can estimate the floating potential from OML theory and a current balance equation. However, one of the assumptions of OML theory is that the plasma must be non-drifting and Maxwellian (Whipple 1981). For non drifting plasmas, the thermal speed of the charged particles is significantly larger than the relative velocity of a spacecraft to the plasma (Jacobsen et al. 2010). For our simulations, we assume a drift speed of $100kms^{-1}$. From table 3.4 the magnetospheric plasma has an electron temperature of $T_e = 1.16 \times 10^6$ which amounts to a thermal speed of $4193kms^{-1}$. Since the electron thermal speed is so high compared to our plasma drift, we can safely make the approximation that electrons in the plasma are non-drifting.

For cases 1 and 6, the only currents affecting the spacecraft potential are the currents due to impinging ions and electron. The current balance equation is therefore

$$I_i(\phi_f) - I_e(\phi_f) = 0, \tag{4.1}$$

where I_i is the current due to impinging ions (proton), and I_e is the current due to impinging electrons. The main body of the MMO can be approximated as a cylinder of height 0.9 m and with a diameter of 1.8 m. We can then substitute in equation (2.42b), and (2.44) to get

$$I_i(0)\left(\frac{2}{\sqrt{\pi}}\sqrt{1-\frac{e\phi_f}{kT_i}}\right) - I_e(0)\exp\left(\frac{e\phi_f}{kT_e}\right) = 0. \tag{4.2}$$

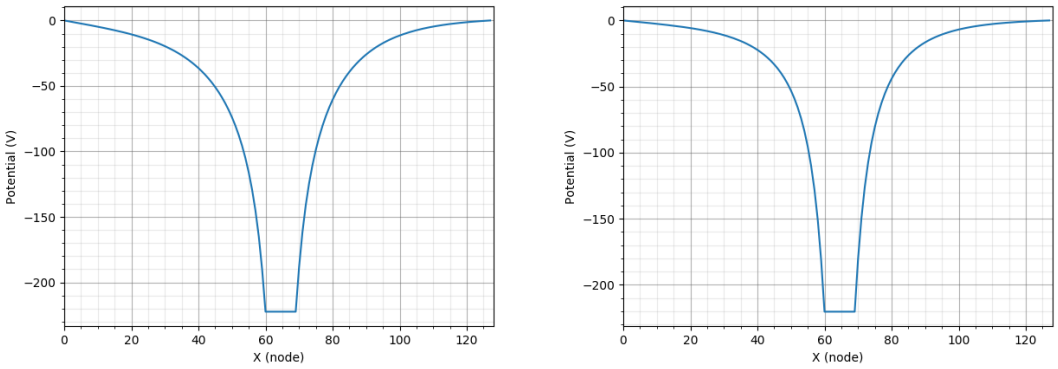
The current when the spacecraft is at zero potential for a cylinder is given by (Lai 2019)

$$I_s(0) = 2\pi a n_s q_s v L, \tag{4.3}$$

where a is the cylinder radius, and L is the cylinder length and v is the speed of the charged particle an infinite distance from the cylinder. Substituting equation (4.3) into equation (4.2), we have

$$\frac{2}{\sqrt{\pi}}\sqrt{1-\frac{e\phi_f}{kT_i}} = -\exp\left(\frac{e\phi_f}{kT_e}\right). \tag{4.4}$$

Here we have assumed that the floating potential of the MMO will be negative, and that $\eta_s = \exp\left(-\frac{q_s\phi_f}{kT_s}\right) > 2$. We solve the transcendental equation (4.4) by plugging in the known values for T_i and T_e to get a floating potential $\phi_f = -298.3V$.



(a) Booms (b) Without booms

Figure 4.2: 4.2a and 4.2b show a potential profile along the X axis for the MMO with and without booms respectively. The line is plotted at $(x, y) = (13.95m, 13.95m)$, or node points $(x, y) = (62, 62)$, and passes through the main octagonal body of the spacecraft. The X axis units are in number of nodes from the origin.

Even without PINC converging to a floating potential, we can still make comparisons of the plasma sheath formed around the spacecraft by plotting the potential profile of the domain. Figure 4.2, shows the potential relative to the ambient plasma plotted on a center line passing through the spacecraft octagonal body. comparing figure 4.2a with figure 4.2b, we see a difference in sheath thickness. For a negatively charged probe, we can estimate the thickness of the sheath using Bohm's sheath criterion (Chen 2018):

$$u_0 \approx \left(\frac{KT_e}{m_i}\right)^{1/2}, \tag{4.5}$$

where u_0 is the velocity required by ions at the sheath boundary. For ions to be accelerated to this velocity we can estimate the required sheath edge potential as

$$\phi_s = -\frac{1}{2}\frac{KT_e}{e}. \tag{4.6}$$

Where ϕ_s is the potential relative to the ambient plasma. Setting the electron temperature, T_e, to 1.16×10^6 we find the sheath boundary potential to be approximately $-50V$ relative to the main plasma body. The sheath thickness is then the distance from the potential along the x axis to where the potential is the same as the sheath boundary potential. From the plots we then estimate

a sheath thickness of 2.5 meters for the MMO configuration without booms, and 3.2 meters for the configuration with booms.

The sheath thickness, d, can also be computed directly by re-arranging the Child-Langmuir law (Chen 2018):

$$d^2 = \frac{4}{9}\left(\frac{2e}{m_i}\right)^{1/2} \frac{|\phi_w|^{3/2}}{4\pi j_{i,s}}. \tag{4.7}$$

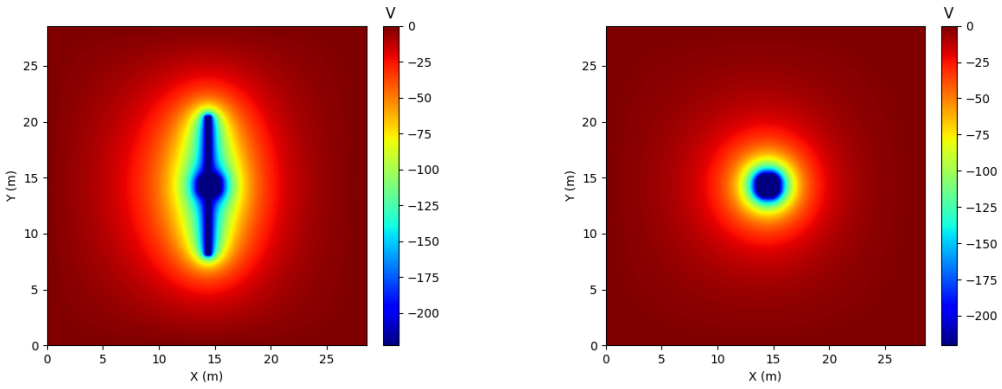
where ϕ_w is the "wall", or spacecraft potential, and $j_{i,s}$ is the ion saturation current density. The ion saturation current density is given by

$$j_{i,s} = q_e n_e c_s. \tag{4.8}$$

Where c_s is the ion speed of sound, and is computed as (Paulsson et al. 2019)

$$c_s = \sqrt{\left(\frac{K_b T_e}{m_i}\right)}, \tag{4.9}$$

Plugging in the known spacecraft potential, the electron density, and electron temperature from table 3.4 the plasma thickness comes out to be 3.63 meters, which is comparable to our estimates.

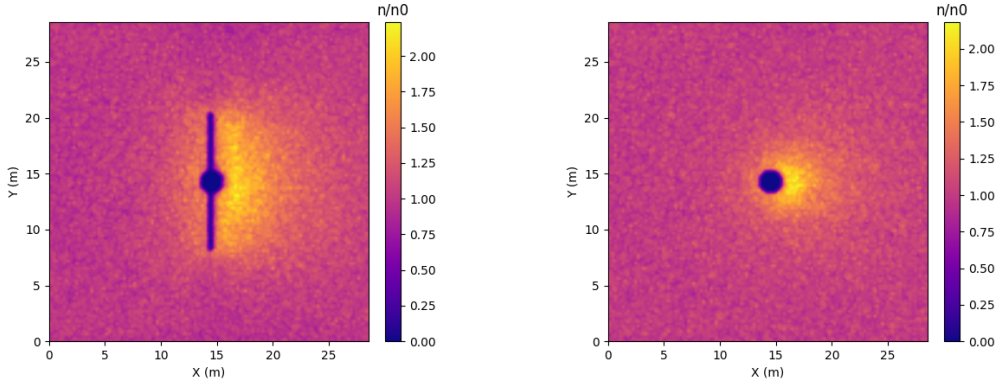


(a) Booms

(b) Without booms

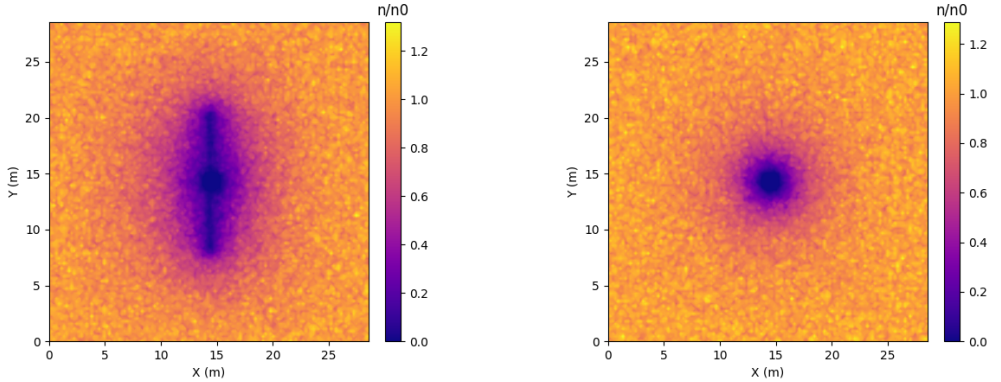
Figure 4.3: 4.3a and 4.3b are 2D slices through $Z = 14.4m$ showing the potential profile of the entire computational domain for simulation cases 1 and 6.

Figure 4.3 shows 2D cut in the plane $Z = 14.4$, dividing the spacecraft in half, and is plotted at the last simulated timestep (timestep 40,000). The contours are plotted using the matplotlib Python package, with contour lines subdivided into 500 levels to better distinguish the plasma sheath. Values are converted from normalized PINC internal values to Volts. The color boundary between orange and red approximately marks the plasma sheath edge. We can see this boundary is equidistant from the MMO body in the boomless configuration, whereas the sheath boundary is relatively closer to the boom tips than to the main octagonal body of the spacecraft.



(a) Booms (b) Without booms

Figure 4.4: Ion density profile plotted at $Z = 14.4m$ with and without deployed booms. The color gradient is normalized against the ion plasma density from table 3.4.



(a) Booms (b) Without booms

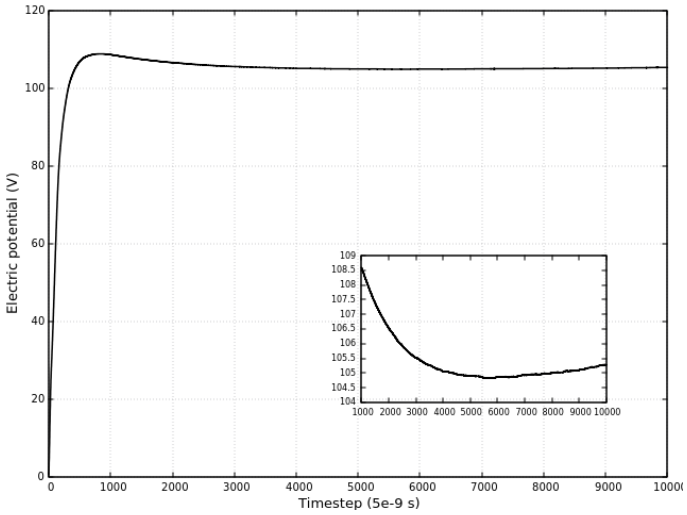
Figure 4.5: Electron density profile plotted at $Z = 14.4m$ with and without deployed booms. The color gradient is normalized against the electron plasma density from table 3.4.

Figures 4.4 and 4.5 show the density of ions (protons) and electrons respectively plotted at timestep 400, 1% through the total simulation time. Similarly to figure 4.3, the 2D slice is placed at $Z = 14.4m$, dividing the spacecraft in halves. The density values have been normalized against the ambient plasma density, and as before, 500 levels were used for the contour lines drawn. For cases 1 and 6, we set the plasma drift along the positive X

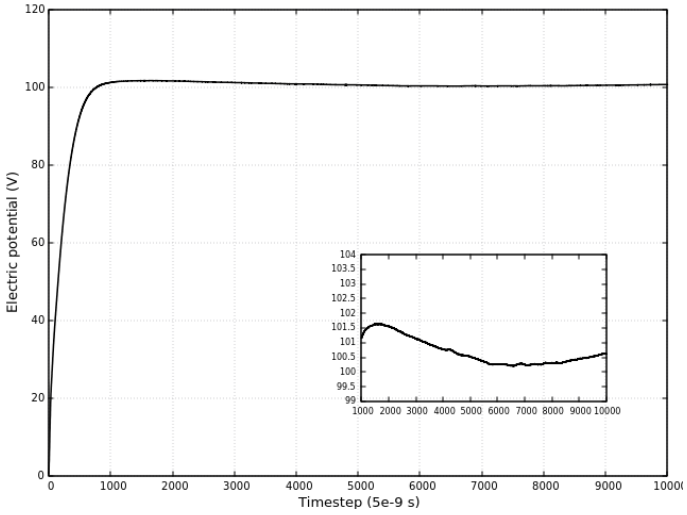
axis, 4.4 clearly shows an ion wake forming "behind" the spacecraft for both boom configurations. We also note the slightly higher density of ions directly in "front" of the spacecraft, denoting the plasma sheath that has formed even this early in the simulation. Figure 4.5 gives the clearest picture of the shape of the plasma sheath formed around the spacecraft, in the MMO configuration with booms, the sheath forms an ellipsoid around the spacecraft where, relative to the ambient plasma, almost no electrons reside. The sheath in the boomless MMO configuration is circular in shape, which gives a good indication that computing the theoretical floating potential of the boomless MMO configuration can be done by assuming the MMO to be a cylinder with the same radius as the circle circumscribing the MMO octagonal body.

4.2 Charging with photoemission

4.2.1 Drift parallel to X axis



(a) Booms



(b) Without booms

Figure 4.6: Time series plot of the potential of the two MMO configurations for drift along X axis. The insets plots the same timeseries starting at 1000 timesteps, where the potential of the spacecraft has begun to oscillate about the floating potential.

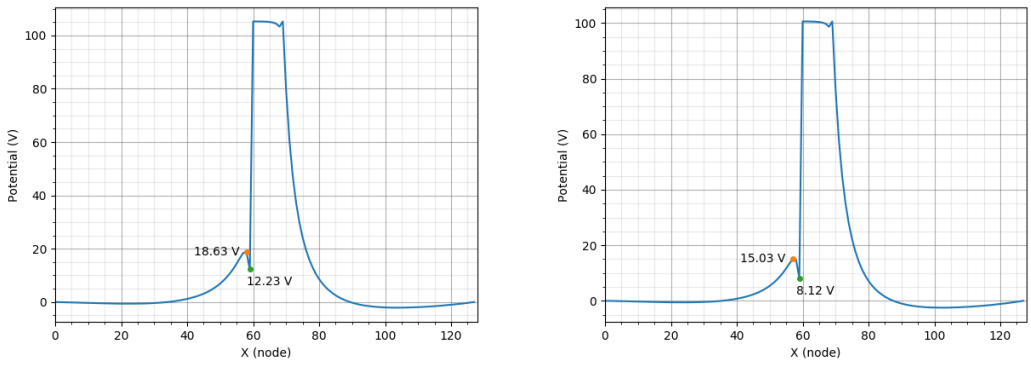
Figure 4.6 shows the convergence to the floating potential of both MMO configurations. Unlike the numerical simulations without photoemission, the potential of both configurations of the MMO converge rapidly to a its floating potential. The slope of the increase in potential early on in the simulation

points to the high ratio between the photoemission current and the charging from the impinging electrons from the ambient plasma. The absolute value of the current density j_{ph} can be calculated from the equation

$$j_{ph} = \frac{N_p \overline{Y}_{ph} q_e}{A_{ph} dt}. \tag{4.10}$$

Where N_p is the number of impinging photons per timestep with energy higher than the material work function, and $\overline{Y}_{ph}$ is the averaged photoelectron yield. Plugging in N_p output by the simulation, and $\overline{Y}_{ph}$ from Table 3.4, gives the photoemission current density. We have $j_{ph} = 0.237 Am^{-2}$ for both configurations of the booms.

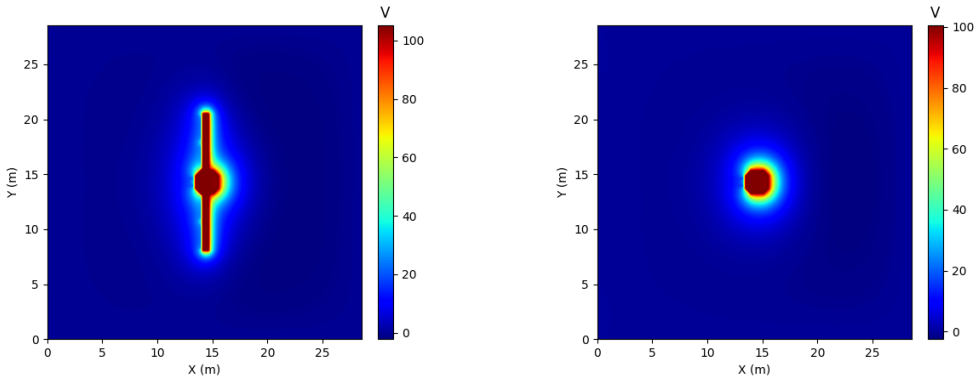
The floating potential for MMO with booms, as seen in figure 4.6a overshoots the floating potential by several volts, before converging to 105.4 V. In the boomless configuration, figure 4.6b, the overshoot is much smaller, and the potential shows oscillatory behaviour around a floating potential of 100.7 V. The overshoot in the floating potential could result from the thin booms extending far from the main body of the MMO developing a potential barrier slower, and smaller in height, than the central body leading to a gradual rather than sudden decrease in the effective photoemission current.



(a) Booms

(b) Without booms

Figure 4.7: Potential profile along the X axis for the two MMO configurations with drift along the X axis and photoemission included. The line is plotted at $(x, y) = (13.95m, 13.95m)$, or node points $(x, y) = (62, 62)$, and passes through the main octagonal body of the spacecraft. The X axis units are in number of nodes from the origin. The two values in each plot show the height of the potential barrier formed relative to the ambient plasma.

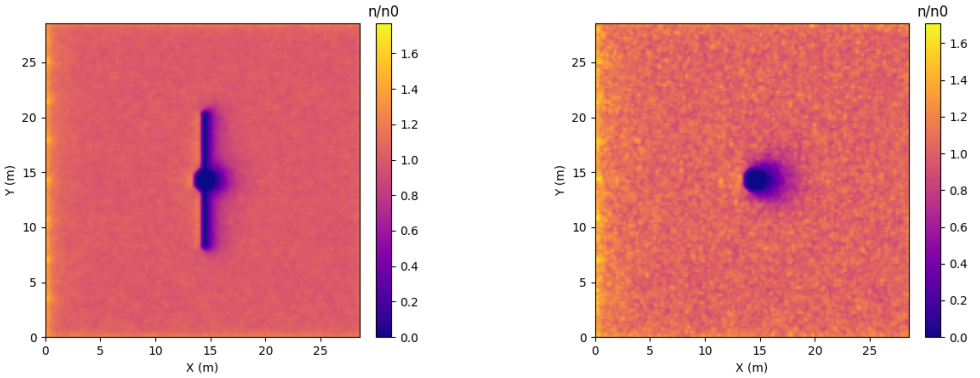


(a) Booms (b) Without booms

Figure 4.8: 2D slices through $Z = 14.4m$ showing the time averaged potential profile of the entire computational domain with drift along X axis, and photoemission included. The potential is time averaged after a floating potential has been reached after 1,000 timesteps.

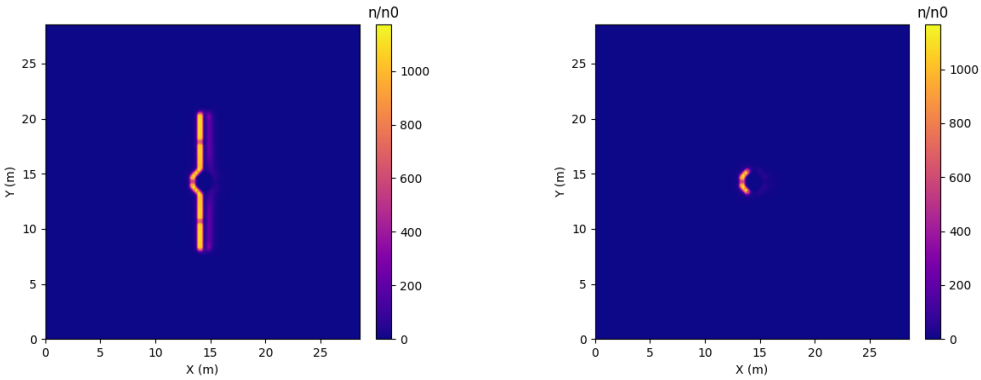
We next plot the potential along the X axis as seen in figures 4.7a and 4.7b. The profile is plotted at timestep 10,000, when the floating potential has stabilized. Note that the profile line does not pass through the centre of the octagonal body of the MMO, rather passing through one of the panels at an angle to the YZ plane, as will become apparent when the 2D potential profile is shown. Directly in front of the MMO, both configurations of the spacecraft show a distinct potential barrier forming on the photoemissive side. In 4.7a the potential difference 6.4 V floating potential of the spacecraft is $93.17V$, and in 4.7b the height of the barrier relative to the plasma is $92.58V$. The configurations of MMO with booms then has a barrier that is $0.59V$ deeper than the configuration without booms. Both plots also show a small distinct drop in potential directly behind the spacecraft, likely caused by photoelectrons orbiting the craft before being reabsorbed.

Figure 4.8 plots contour lines for the potential in 2D slices of the computational domain. Contrasting figures 4.8a and 4.8b with figures 4.3a and 4.3b, we can see a thinner sheath form around both configurations around the MMO when photoemission is included. For both figures 4.8a and 4.8a we can also see that there are regions within the sheath on the sunlit side of the spacecraft where the potential differ varies along the booms. On both booms half way along the length, a small region of slightly higher potential, this suggests a gradient in the local electrical field. This variation in the potential could be the result of the angled geometry of the main body of the MMO, or potentially as a result of the voxelization of the spacecraft. Similarly, the sunlit panel of the octagonal main body closest to the sun shows a lower potential than the sunlit slanted panels. Appendix D contains an analysis and discussion of these potential variations within the photoelectron sheath.



(a) Booms (b) Without booms

Figure 4.9: Ion density profile plotted at $Z = 14.4m$, the color gradient is normalized against the ion plasma density from table 3.4. Plasma drift is along the X axis, and photoemission is included



(a) Booms (b) Without booms

Figure 4.10: Electron density profile plotted at $Z = 14.4m$, the color gradient is normalized against the electron plasma density from table 3.4. Drift is along the X axis, and photoemission is included, the sun is located in the negative X direction.

Plasma particle densities are plotted for both ions (protons) and electrons in figures 4.9 and 4.10 respectively. Due to a high photoemission current, the positively charged spacecraft does not form an ion wake, as can be seen in both 4.9a and 4.9a. Downstream of the spacecraft, there is a relative low ion density relative to the ambient plasma which is more prominent in the case of the boomless configuration of the MMO.

Variations in the ion density in the wake of a spacecraft is also realted to the plasma Mach number, as seen in the paper by Miloch et al. (Paulsson et al. 2019). We can compute the Mach number of the plasma using equation (4.9),

and the Mach number equation

$$M = v_d/c_s, \tag{4.11}$$

where v_d is the velocity of the plasma relative to the spacecraft. Thus far in our simulation we have assumed the relative velocity of the orbital speed of the MMO to be negligible as compared to the drift velocity of the plasma. However, to accurately compute the Mach number, we must compute the relative speed of the plasma to the MMO. Due to its eccentric orbit, the orbital speed of the MMO is computed using the equation

$$v = \sqrt{\mu\left(\frac{2}{r} - \frac{1}{a}\right)}. \tag{4.12}$$

In equation (4.12), r denotes the distance from the MMO to the center of Mass of Mercury, μ is the standard gravitational parameter, and a is the major axis of the elliptical orbit. The distance r is approximately half the radius of Mercury at the location in the orbit selected for our simulations (Benna et al. 2009), and the major axis can be found from the perihelion distance of 400 km, and aphelion distance of 11824 km of the MMO in orbit. For mercury, the gravitational parameter is $\mu = GM_{Mercury} = 2.2032 \times 10^{13}$. Substituting into equation (4.12), the orbital speed of the MMO is 5.7 km/s. The highest possible speed of the plasma relative to the MMO is then 105.7 km/s. With the ion speed of sound c_s we computed in the the previous section for charging without photoemission, the maximum Mach number possible in our simulation is $M \approx 1.08$. Since the Mach number is weakly supersonic, the ion enhanced wake is narrow and does not extend far from the spacecraft in the direction of the drift, which is in good agreement with results found by Paulsson et al. (Paulsson et al. 2019).

Figures 4.10a and 4.10b show high concentration of electrons adjacent to the sunlit surfaces of the MMO, with electron densities as high as 1175 times the density of electrons in the ambient plasma.

At such high densities, the local Debye length is significantly shorter than that of the ambient plasma. We can compute the local Debye length from equation (2.30) and from the photoelectron temperature. The photoelectron temperature is computed in PINC internally by numerical integration of equation (3.28) in terms of energy and photons per second, subtracting the work function of the surface material gives the photoelectron temperature of $T_{ph} = 0.62\ eV$. Substituting the photoelectron density and temperature into equation (2.30) we have a local Debye length of only 0.017 meters. Since our grid resolution is 0.225 meters, we are unable to resolve the local small scale variations in electrical potential due to Debye shielding.

We can estimate the thickness of the photoelectron sheath using the equation (Zhao et al. 1996)

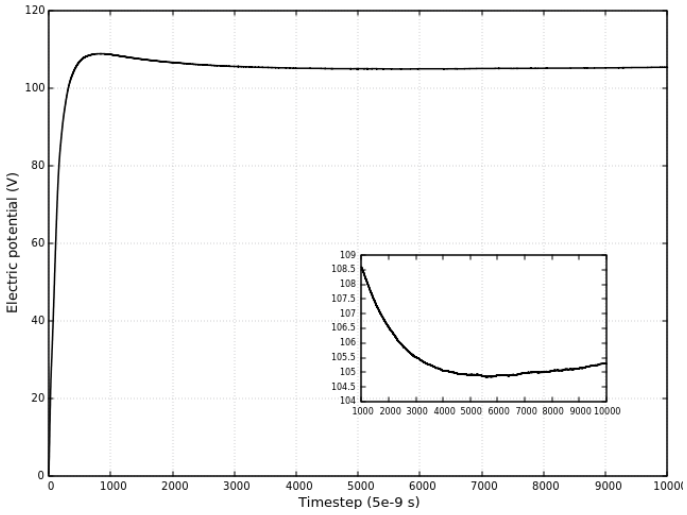
$$n_{ph}(x_{ph}) = n_{e0} \tag{4.13}$$

Where $n_{ph}(x_{ph})$ is the density of photoelectrons at a distance x_{ph} from the spacecraft, and n_{e0} is the electron density in the ambient plasma. The distance x_{ph} is the radial distance from the spacecraft, and is normalized by the spacecraft radius R. Note that the implementation of photoelectron injection in this thesis does not separate photoelectrons as a separate species from the ambient plasma,

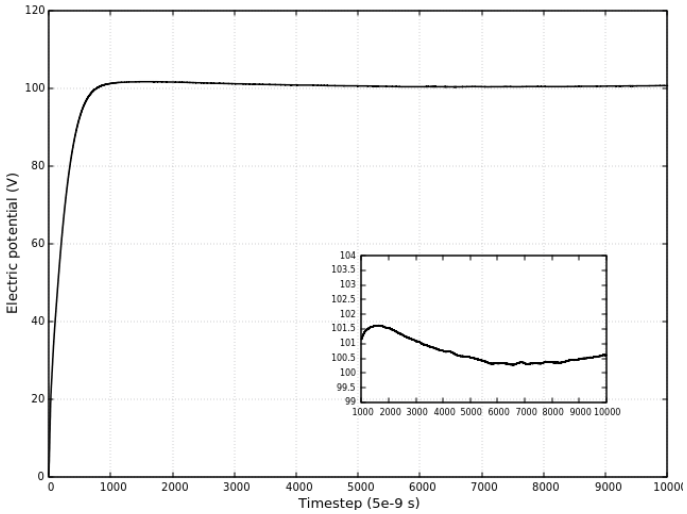
and the distance between density measurements is relatively high (one cell width being 0.225 meters). By plotting the electron density as a line through the spacecraft parallel to the X axis, we estimate the photoelectron sheath to be 6.75 meters thick.

4.2.2 Drift parallel to Z axis

Drift along the negative Z axis, with photoemission, and no external electric field (cases 3 and 8) is a more accurate plasma flow conditions that the MMO will experience in its orbit over the ecliptic north pole as opposed to drift along the X axis (cases2 and 7). Results for cases 3 and 8 will be compared to previous results shown for cases 2 and 7 to discern what effect, if any, drift direction has on the photoelectron barrier, the floating potential, and the plasma sheath.



(a) Booms



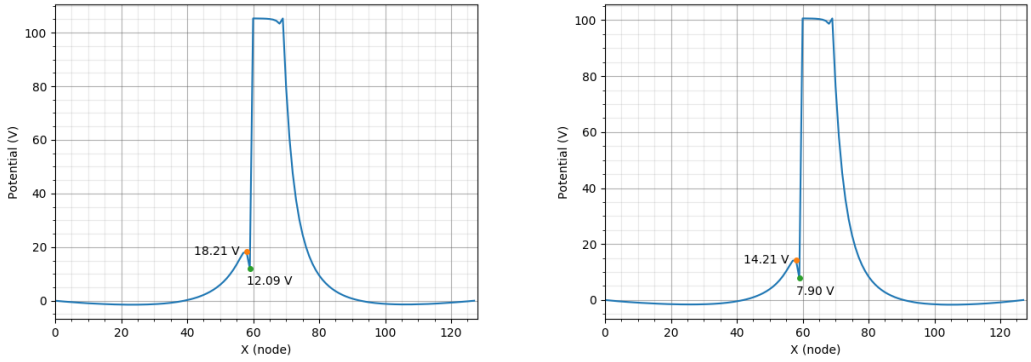
(b) Without booms

Figure 4.11: Time series plot of the potential of the MMO with and without booms, where the drift is along the negative Z axis and photoemission is included. The inset plots the same timeseries after 1000 timesteps, where the potential of the spacecraft has begun to oscillate about the floating potential.

Figures 4.11a and 4.11b show the potential convergence to the floating potential for both MMO configurations. Comparing these figures to 4.6a and 4.6a, we can see that the final floating potential, and total charging time to the floating potential, seem to be independent of the drift direction. In the configuration of MMO with booms 4.6a, the potential converges to 105.42 V, and in the boomless configuration 4.11b converges to 100.71 V. The difference in floating potential for the two directions of drift are so small as to be negligible. Since the photoemission current, and the emissive surfaces remain the same,

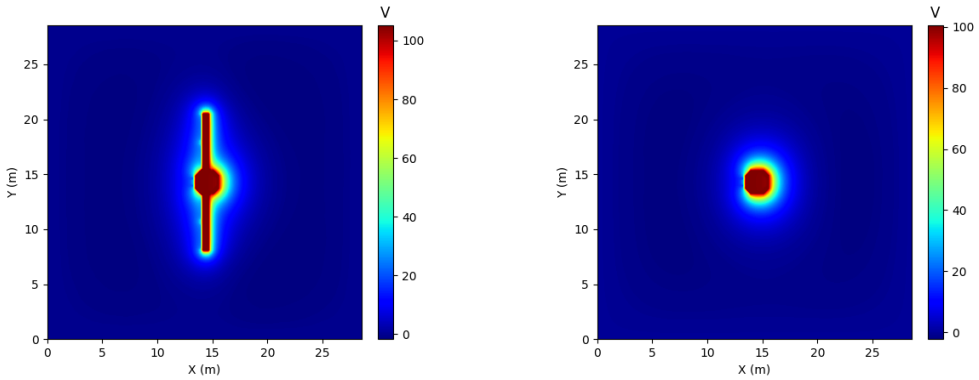
only the ambient electron charging could be the cause of a difference in floating potential.

The MMO without booms can be approximated as a cylinder, equation (2.42b) then gives the approximate current due to impining electrons. The current collected when the spacecraft is neutrally charged, $I_{th,e}$ is dependent on the cylinder geometry, from equation (4.3), it is apparent then that the current from the ambient electrons is dependent on the total surface area of the spacecraft and the magnitude of the drift velocity, but not on its direction.



(a) Booms (b) Without booms

Figure 4.12: Potential profile along the X axis for the two MMO configurations with drift along the negative Z axis and photoemission included. The line is plotted at $(x, y) = (13.95m, 13.95m)$, or node points $(x, y) = (62, 62)$, and passes through the main octagonal body of the spacecraft. The X axis units are in number of nodes from the origin. The two values in each plot show the height of the potential barrier formed relative to the ambient plasma.

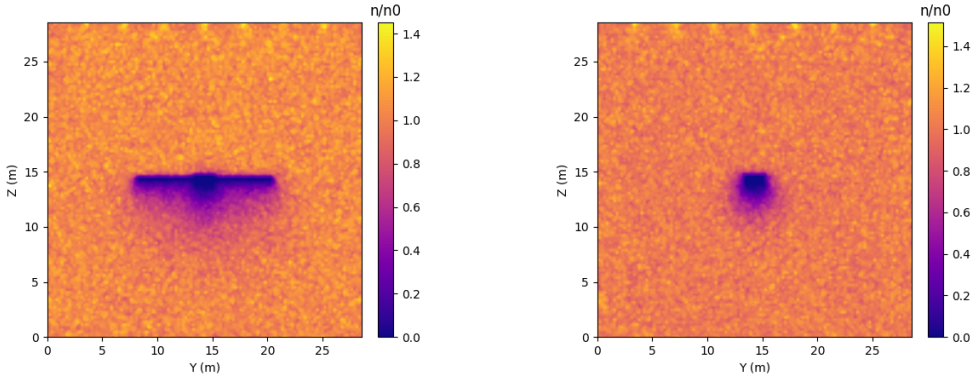


(a) Booms (b) Without booms

Figure 4.13: 2D slices through $Z = 14.4m$ showing the time averaged potential profile of the entire computational domain with drift along the negative Z axis, and photoemission included. The potential is time averaged after a floating potential has been reached after 1,000 timesteps.

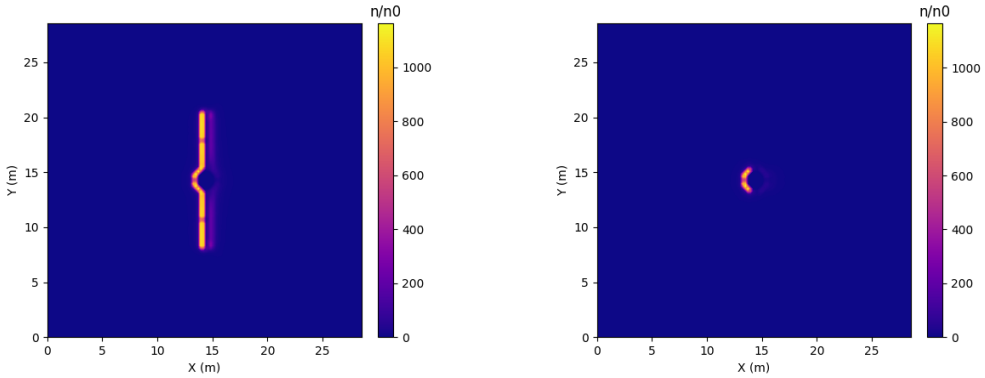
Figures 4.12b and 4.12a show the potential profile along the X axis for cases 3 and 8. The minimum potential in the potential barrier is slightly lower than for cases 2 and 6, the potential barrier height from figure 4.12a is $93.33V$, and for the MMO without booms figure 4.12b gives a barrier height of $92.81V$.

Figure 4.13 the 2D potential profile for simulation cases 3 and 8; no significant difference can be seen in the shape and thickness of the sheath formed around the MMO. Both figures 4.13a and 4.13b again show the distinct variation in potential within the sheath along the booms, and on the sunlit octagonal panel perpendicular to the Sun/MMO vector.



(a) Booms (b) Without booms

Figure 4.14: Ion density profile plotted at $X = 14.4m$, the color gradient is normalized against the ion plasma density from table 3.4. Drift is along the negative Z axis. and photoemission is included.

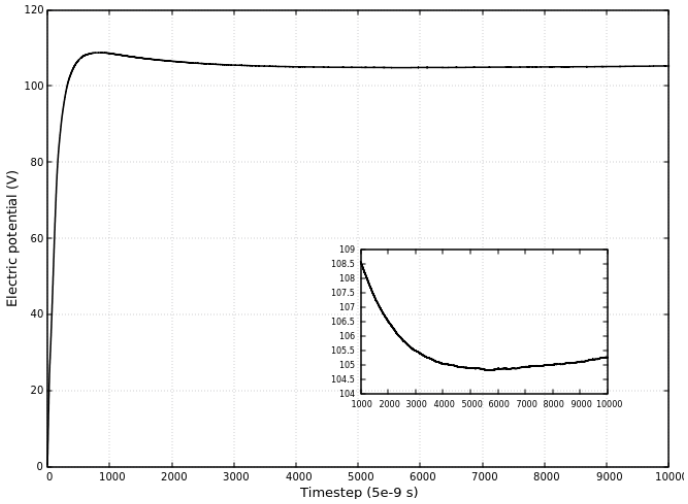


(a) Booms (b) Without booms

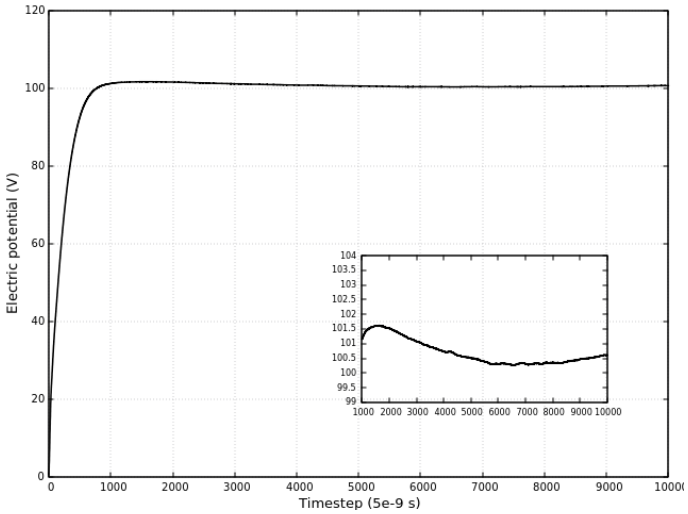
Figure 4.15: Electron density profile plotted at $Z = 14.4m$, the color gradient is normalized against the electron plasma density from table 3.4. Drift is directed into the page, and photoemission is included. The sun is located in the negative X direction.

The ion density and electron density profile plots for simulation cases 3 and 8 are given by figure 4.14 and figure 4.15 respectively, since drift is along the negative Z axis the ion densities are plotted in the YZ plane. Similarly to the ion densities for cases 2 and 7 both MMO configurations show a thin ion enhanced wake downstream of the spacecraft. The profile plots of electron density are illustrative in showing the formation of the photoelectron cloud, yet gives no information on the average temperature or velocity distribution of the particles within the photoelectron sheath. Section Appendix B contains an analysis comparing the particle distributions found within the photoelectron sheath, and for the whole computational domain.

4.2.3 Charging in an external magnetic field



(a) Booms



(b) Without booms

Figure 4.16: Timeseries plot of the potential of the two configurations of the MMO, drift is along the negative Z axis, photoemission and an external magnetic field are included. The potential has been converted from PINC dimensionless units to Volts. The inset plots shows the potential of the two configurations for last 9,000 timesteps.

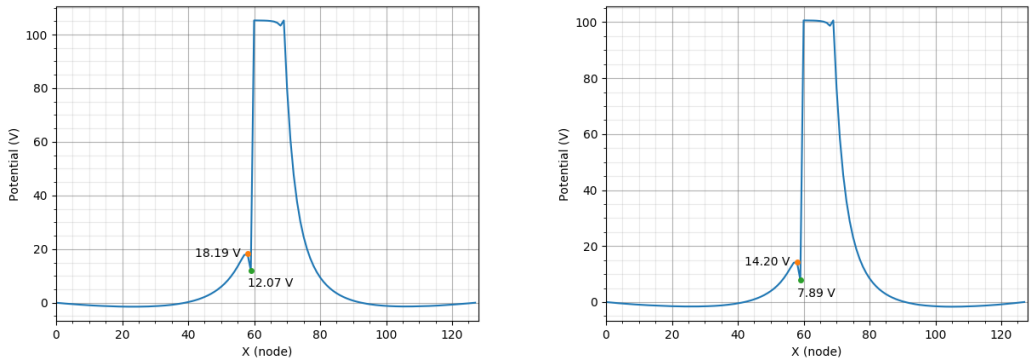
The convergence of the potential of the two MMO configurations to a floating potential for simulate case 4 and 9 are given in figure 4.16. Similarly to case 3 and case 9, the plasma drift is along the minus Z axis of the domain. It is apparent from both figures 4.16a and 4.16b, that the potential of the two MMO configurations converge to almost exactly the same floating potential as for the

two simulations without an external field: $105.4V$ for the MMO with booms, and $100.7V$ in the boomless configuration. The same overshoot behaviour of the potential before convergence as seen in figure 4.16a, is also seen in figure 4.11a. Figure 4.16b also displays the same behaviour of convergence to the floating potential as in figure 4.11b without overshooting the floating potential.

Particles that enter the computational domain experience a force given by equation (2.1), in the absence of an external electric field, this equation becomes

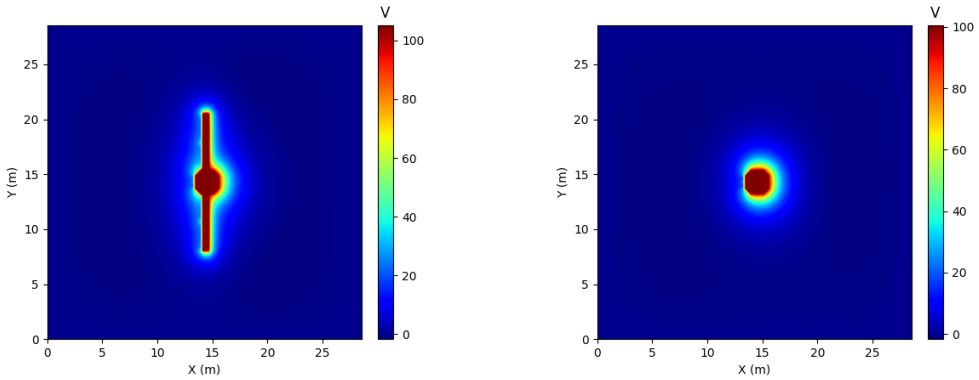
$$\mathbf{F} = q\,(\mathbf{v} \times \mathbf{B}) = -q|v||B|\sin\theta\,\mathbf{j}. \tag{4.14}$$

Here, we have used θ to denote the angle between the velocity vector and the magnetic field vector in the XZ plane, and $\mathbf{j}$ as the unit vector along the Y axis. Plugging in the drift velocity, the magnetic field strength, and $\theta = 5.03°$ a force of $1.404 \times 10^{-22}N$ will act on the proton and electron in the negative $\mathbf{j}$ and positive $\mathbf{j}$ direction respectively. This force will cause the ions and electron to spiral in opposite direction along guiding centers aligned with the magnetic field lines. We have already seen that drift direction does not affect the floating potential obtained from previous results, thus the floating potential remains the same. So called E cross B drift found close to the MMO where the electrical field is non-zero, is discussed briefly in appendix D.



(a) Booms (b) Without booms

Figure 4.17: Potential profile along the X axis for the two MMO configurations with drift along the negative Z axis, photoemission and an external magnetic field are included. The line is plotted at $(x, y) = (13.95m, 13.95m)$, or node points $(x, y) = (62, 62)$, and passes through the main octagonal body of the spacecraft. The X axis units are in number of nodes from the origin. The two values in each plot show the height of the potential barrier formed relative to the ambient plasma.

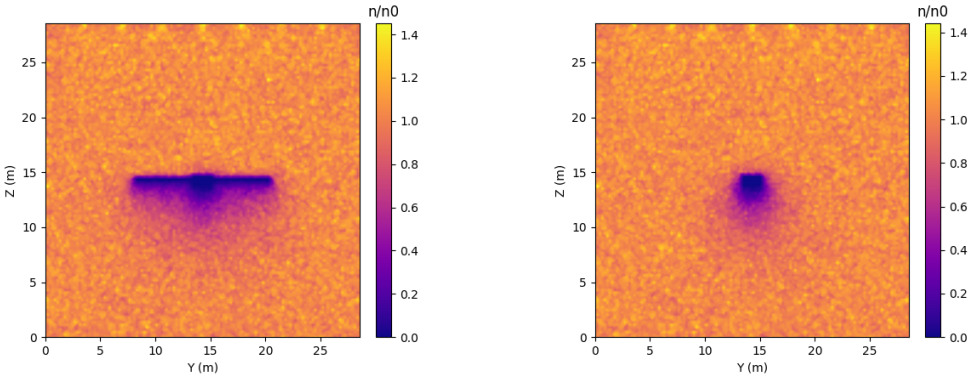


(a) Booms (b) Without booms

Figure 4.18: 2D slices through $Z = 14.4m$ showing the time averaged potential profile of the entire computational domain with drift along the negative Z axis, photoemission and an external magnetic field are included. The potential is time averaged after a floating potential has been reached after 1,000 timesteps.

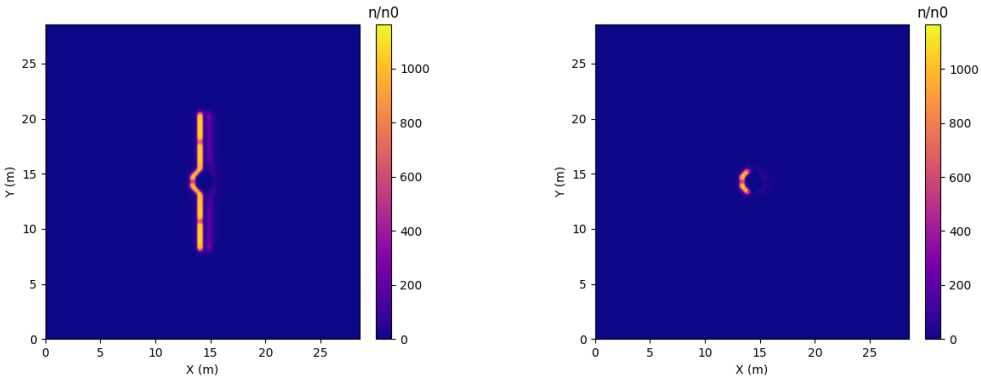
Figures 4.17a and 4.17b show the potential profile for simulation cases 4 and 9 respectively. The potential barrier height with boom configuration of the MMO is 93.33 V, and the barrier height for the configuration without booms is 92.81 V. These potential barrier heights are the exact same as in simulation cases 3 and 8, with very small differences in floating potential and the minimum potential of the barrier: the difference in potential between the minimum potential in the barrier between case 8 and case 9 is only 0.02 V.

Figure 4.18 shows the 2D potential profile for cases 4 and 9. The overall structure of the potential sheath in both configurations of the MMO are similar to those found in cases 3 and 8. From figure 4.18a, the thickness in the sheath on the sunward side is however slightly larger than the sheath thickness seen in figure 4.13a.



(a) Booms (b) Without booms

Figure 4.19: Ion density profile plotted at $X = 14.4m$, the color gradient is normalized against the ion plasma density from table 3.4. Drift is along the negative Z axis, photoemission and an external magnetic field are included.

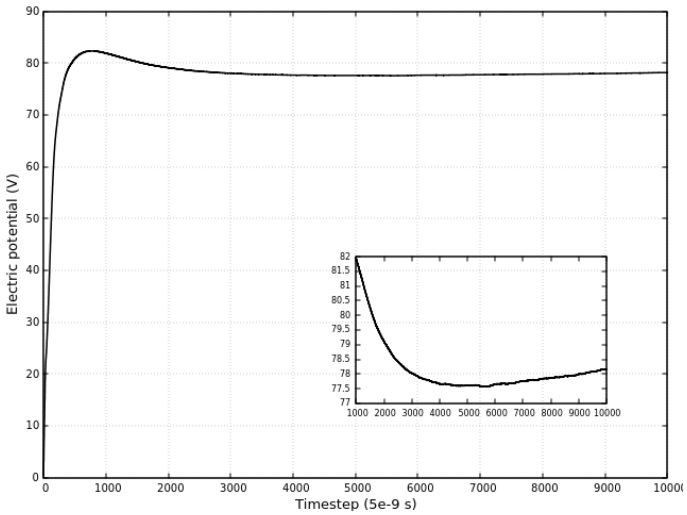


(a) Booms (b) Without booms

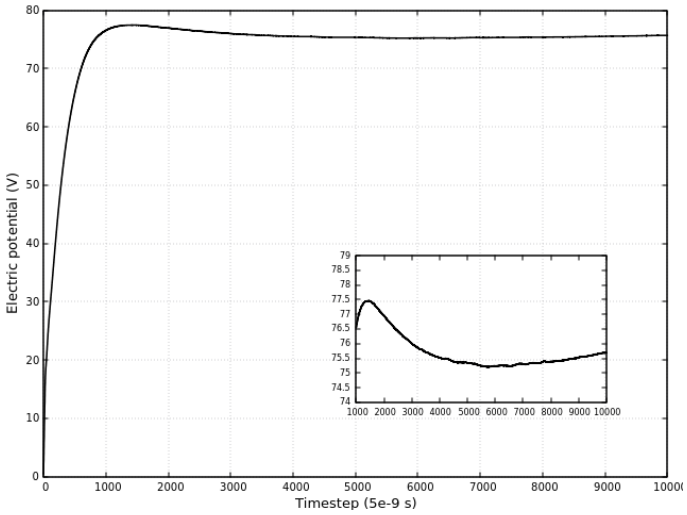
Figure 4.20: Electron density profile plotted at $Z = 14.4m$, the color gradient is normalized against the electron plasma density from table 3.4.Drift is along the negative Z axis, photoemission and an external magnetic field are included. The direction of the sun is along the negative X axis.

Figures 4.19 and 4.20 show the ion and electron density profile for both cases 4 and 9 respectively. The enhanced wakes formed downstream in both figure 4.20a and 4.20b show no discernible difference than the wakes formed in simulation cases 3 and 8, with similar shape and characteristic length. Likewise, there is no difference density of electrons on the photoemissive surfaces of the MMO, with the maximum density from 4.20a and 4.15a being 1164.73 times the density of the electrons in the ambient plasma.

4.2.4 Charging at different photoelectron temperatures



(a) Booms



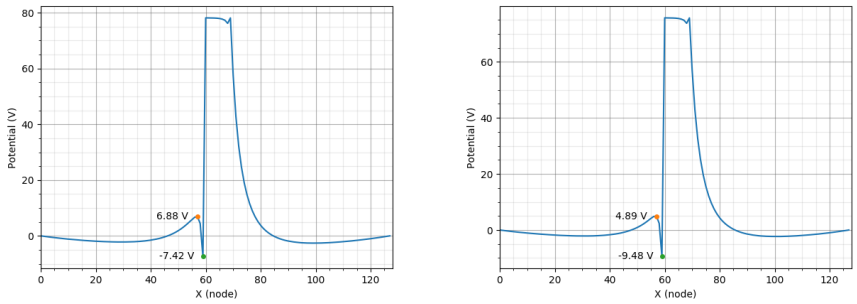
(b) Without booms

Figure 4.21: Timeseries plot of the potential of the two configurations of the MMO, drift is along the negative Z axis and the photoelectron temperature has been set to 3 eV. The potential has been converted from PINC dimensionless units to Volts. The inset plots the same timeseries after 1000 timesteps, where the potential of the spacecraft has begun to oscillate about the floating potential.

Internally in PINC, we have made the assumption of constant yield of photoelectrons by averaging the yield over the energy of incoming photoelectrons. Since our computation of the temperature is based on this averaging, another simulation was carried out where the photoelectron temperature was set to a

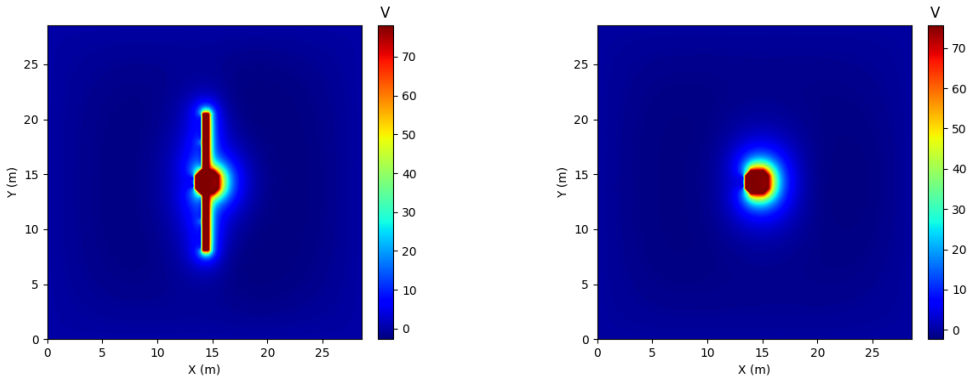
constant, $T_{ph} = 3\ eV$. The photoemission current was kept constant and no external magnetic field was included. Since drift was along the negative Z axis, these simulation cases (cases 5 and 10) are compared against simulation cases 3 and 8.

Figures 4.21a and 4.21b show the converge of the potential convergence of two boom configurations for case 5 and 10. A significantly lower floating potential is reached for both boom configurations as compared to cases 3 and 8. The floating potential of the MMO with deployed booms was 78.15 V, and 75.75V for the MMO without deployed booms. The difference in floating potential between the two configurations is smaller than for the cases with a lower photoelectron temperature. Unlike case 3, figure 4.21b shows an overshoot of the spacecraft potential before converging to the final floating potential, which could be due to a greater fraction of the photoelectrons having high enough kinetic energy to escape the potential barrier formed, thereby reducing the effective photoemission current.



(a) Booms (b) Without booms

Figure 4.22: Potential profile along the X axis for the two MMO configurations with drift along the negative Z axis, photoemission is included with a photoelectron temperature of 3 eV. The line is plotted at $(x, y) = (13.95m, 13.95m)$, or node points $(x, y) = (62, 62)$, and passes through the main octagonal body of the spacecraft. The X axis units are in number of nodes from the origin. The two values in each plot show the height of the potential barrier formed relative to the ambient plasma.

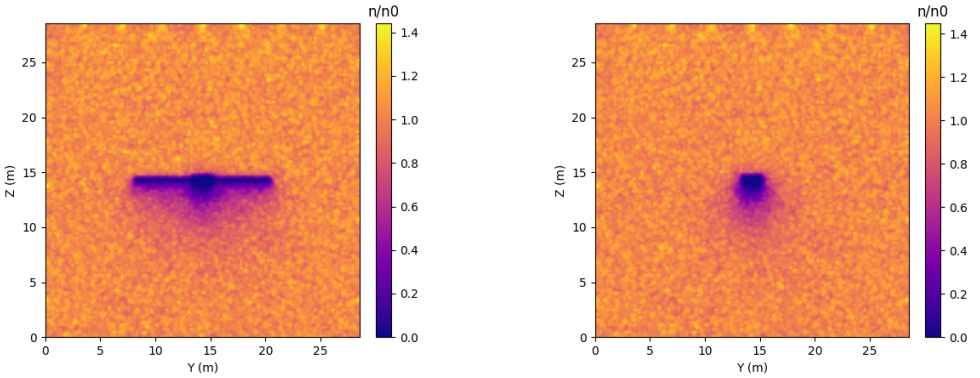


(a) Booms (b) Without booms

Figure 4.23: 2D cut through $Z = 14.4m$ showing the time averaged potential profile of the entire computational domain with drift along the negative Z axis, photoemission is included with a photoelectron temperature of 3 eV. The potential is time averaged after a floating potential has been reached after 1,000 timesteps.

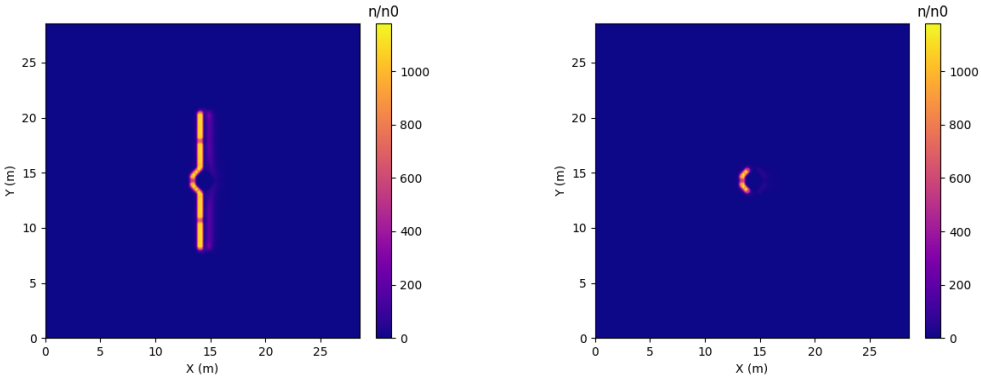
Figure 4.22 shows the potential profile of the MMO along the X axis for both boom configurations. In both cases, the minimum potential of the potential barrier is significantly lower than the minimum potential for other cases where the photoemission current is included: The barrier height is 85.58 V when booms are deployed and 85.23 V without booms. The absolute value of the potential barrier height for the simulation cases $T_{ph} = 3\ eV$ is therefore substantially larger than for cases with $T_{ph} = 0.62\ eV$.

The 2D potential profile for case 10 and case 5 is shown by figure 4.23a and figure 4.23b respectively. The potential at the saddle point located adjacent to the sunlit surface perpendicular to the vector pointing towards the sun is noticeably more negative for both boom configurations as compared to cases 3 and 8. Additionally, the potential of the two symmetrical spikes along the booms in figure 4.23a is lower than the spikes seen in figure 4.13a for case 8.



(a) Booms (b) Without booms

Figure 4.24: Ion density profile plotted at $Z = 14.4m$, the color gradient is normalized against the ion plasma density from table 3.4. Drift is along the negative Z axis, and photoemission is included with a photoelectron temperature of 3 eV.



(a) Booms (b) Without booms

Figure 4.25: Electron density profile plotted at $Z = 14.4m$, the color gradient is normalized against the electron plasma density from table 3.4. Drift is along the negative Z axis, and photoemission is included with a photoelectron temperature of 3 eV.

Figure 4.24 show a 2D profile of ion density for case 10 and case 5. Comparing figure 4.24a with 4.24a, we can see that the length of the wake formed behind the MMO in the direction of drift is shorter than in figure 4.14a for case 3. If we extract the ion density for each cell in the Z axis at $X = 64$ and $Y = 64$, we can estimate the length of the wake formed applying the following equation

$$n_i(z_w) = n_{i0}. \tag{4.15}$$

Where z_w denotes the length of the wake measured from the bottom panel of the MMO to the edge of the wake, and n_{i0} denotes the density of the ambient

ions in the plasma upstream from the MMO. Using this method, the length of the wake for case 10 is 4.73 m, and $5.53m$ for case 3.

The 2D electron density profile for case 10 and case 5 are given in figure 4.25a. Although not immediately obvious from the color contours, the maximum electron density adjacent to sunlit surfaces is slightly higher in case 5 and 10 as compared to the maximum electron density in case 3 and 8. The maximum density of electrons for case 10 is 1180.45 times the ambient electron density, and 1164.74 times the ambient electron density for case 8. The following equation gives the density of photoelectrons able to escape the potential barrier (Zhao et al. 1996)

$$n_{ph}(x) = \frac{J_{ph}(x)}{e\overline{v}_{ph}}. \tag{4.16}$$

Where J_{ph} has been used to denote the absolute value of the effective photoelectron current density, and $\overline{v}_{ph}$ denotes the mean velocity of emitted photoelectrons. The effective photoelectron current density is however proportional to the height of the potential barrier (Zhao et al. 1996)

$$J_{ph} \propto \exp\left\{-\frac{e\phi_b}{kT_{ph}}\right\}. \tag{4.17}$$

Where ϕ_b denotes the potential barrier height. Since the barrier height is much larger in case 8, the effective current density flux compensates for the higher average velocity of photoelectrons in case 10, leading to a higher maximum density of photoelectrons in case 10.

4.3 Summary of results

We can now summarize the results for all the simulation cases completed. For all the simulation cases, the floating potential of the MMO was computed, and an estimate for the plasma sheath thickness was found. In the simulation cases where the photoemission current was included, we also computed the potential barrier height due to the photoelectron cloud formed adjacent to the sunlit surfaces of the spacecraft. Table 4.2 shows these values for all simulation cases run for this thesis.

Simulation case	Floating potential (V)	Barrier height (V)	sheath thickness(m)
Case 1	-283^*	N/A	2.5
Case 2	100.70	92.58	6.75
Case 3	100.71	92.81	6.75
Case 4	100.70	92.81	6.3
Case 5	75.75	85.23	4.95
Case 6	-281^*	N/A	3.2
Case 7	105.40	93.17	6.525
Case 8	105.42	93.33	6.75
Case 9	105.40	93.33	6.3
Case 10	78.15	85.58	6.75

Table 4.2: Summary of computed floating potentials, barrier height for photoemissive simulation cases, and the plasma sheath thickness. The asterisk over the floating potential values for cases 1 and 6 indicate the values as estimations from the curve fitting given in Appendix C

CHAPTER 5

Summary and Discussion

5.1 Photoemission implementation

The main objective of this thesis was to analyze the charging of the MMO spacecraft in direct sunlight while in orbit around Mercury using three dimensional Particle-In-Cell simulations. Achieving this goal necessitated further developing the object charging module of the PIC framework PINC with functionality for including a photoemission current.

In order to inject photoelectrons in a physically realistic way, the new code had to meet the following requirements: The code had to be able to identify what surfaces of the simulated spacecraft were exposed to sunlight, the average temperature and the amount of photoelectrons to inject, and crucially, how to inject the photoelectrons from the identified photoemissive surfaces.

Two methods were implemented for determining the average photoelectron temperature and what amount to inject per timestep. The simplest method implemented took the photoelectron temperature and current density as input variables, this method then converted the current density to the correct number of photoelectrons and gave these photoelectrons a Maxwellian velocity distribution using the average temperature given by the user. The second method, and the one used to simulate the MMO in Chapter 4, computed the amount of photoelectrons and average temperature directly by numerically integrating Planck's law for Blackbody radiation using the photoelectron yield and work function of the spacecraft surface, as well as the spacecrafts distance from the sun as input parameters.

Three methods for injecting photoelectrons were implemented and compared to other PIC codes for verification purposes: The method selected sampled the tangential and normal velocity of the photoelectrons with respect to the sunlit surface, and filling the computational cells adjacent to the sunlit spacecraft uniformly. This method was found to be the injection method that was best able to reproduce the floating potential other PIC codes converged to. We were also able to show that the algorithm selected for photoelectron injection has a significant impact on the charging behaviour and the floating potential the simulated spacecraft will converge to.

5.2 Comparison of results with theory and other charging simulations

For any spacecraft with plasma instruments as a payload, the floating potential is a critical value that must be considered. Spacecraft charged to a non-zero potential will attract particles with charge of opposite sign to the potential, and reflect particles with charge of the same sign as the potential. This causes a sheath to form around the spacecraft where the density of the attracted particles is greater than the ambient plasma. To accurately measure the ambient plasma temperature and density, the plasma instruments must therefore be able to account for this difference in density.

In section Chapter 4, we have demonstrated that the MMO spacecraft will charge to a positive floating potential in low density plasma due to a high ratio between the photoelectron current and current due to impinging electrons from the ambient plasma, this is in accordance with data from (Garrett 1981), for spacecraft charging in solar wind plasmas. We have also demonstrated that the floating potential of a spacecraft will increase when booms are deployed and not electrically isolated. Furthermore, 4.2 show that the floating potential of a spacecraft is unaffected even in the presence of weak magnetic fields, or slow relative plasma drift relative to the spacecraft: Assuming a non-drifting and un-magnetized plasma, OML theory and the current balance equation show that the floating potential is dependent on the density, temperature of ambient electron and photoelectrons, as well as the shape of the spacecraft. In our simulations these parameters are kept constant when the relative drift velocity is changed, or when an external magnetic field is included, and the floating potential remains constant. This can also be seen when photoemission is not included, for simulation cases 3 and 4, where the floating potential is described by equation (4.4). Similarly for cases 9 and 10 in table 4.1, there is a large decrease in the floating potential of the MMO when the photoelectron temperature is increased. When the photoelectron temperature increases, more electrons This is again in accordance with OML theory, and can be seen in PIC simulations of other spacecraft as well (Sjögren et al. 2012).

For all the simulations cases in Chapter 4, we have demonstrated the presence of a photoelectron cloud. When a potential profile is plotted as a line passing through the spacecraft and parallel to the sun-spacecraft axis, a significant decrease in the potential relative to the ambient plasma is seen. This evidence of a potential barrier forming due to the photoelectron cloud present in from of spacecraft surfaces adjacent to the sunlit spacecraft surfaces. Such barriers are also demonstrated in other PIC spacecraft charging simulations (Deca et al. 2013; Meyer-Vernet 2007; Sjögren et al. 2012). We also see a dependence of the potential barrier height on the photoelectron temperature, which is demonstrated mathematically in the paper by Zhao et al. (Zhao et al. 1996). However, the panel perpendicular to the sun-spacecraft vector shows signification variations in the potential barrier. None of the simulations we have compared our simulations to show this phenomena. Appendix D contains discussion on these relative increases in the plasma potential. We believe these location variations to be unphysical errors resulting from the coarse voxelization of the spacecraft or due to a subtle flaw in the implementation of the photoelectron injection scheme.

The formation of a photoelectron sheath are seen in our results from electron density profile plots. Close to the spacecraft, the electron density reaches maximums of approximately a thousand times that of the ambient plasma, with a similar decrease in the density of ions close to the spacecraft. Using equation (4.13), we were able to estimate the thickness of the photoelectron sheath. However, there is no evidence from our results that show a relationship between the photoelectron temperature and the photoelectron sheath thickness, which is not correct (Zhao et al. 1996). Furthermore, within the photoelectron sheath, our results show spikes in potential along the booms. Literature shows these spikes can occur for booms that are electrically isolated from the main body of the spacecraft (Lai 2019; Miyake, Cully, et al. 2013; Paulsson et al. 2019). The MMO was simulated as perfectly conducting however, and we were unable to show if these spikes are physical or due to a numerical error.

CHAPTER 6

Future work

6.1 Improving the object module

In order to simulate the charging of objects in ambient plasma, PINC uses the capacitance matrix method as described in (Miyake and Usui 2009). The implementation of this method in the PINC framework when the capacitance matrix is applied and charges are redistributed to the object surface assumes the object is composed of a homogeneous, and perfectly conducting, material: Charges due to impinging particles, and photoelectrons, are redistributed evenly in such a way as to achieve an equipotential. Since dielectric surfaces are poor electrical conductors, charged particles tend to remain where they strike on the surface.

In physical systems, this assumption of a perfectly conducting body does not necessarily hold, spacecraft that are composed of both dielectric and conducting surfaces can charge to different levels in sunlight (Lai 2019). For spacecraft with different surface material composition, this difference in surface potential leads to differential charging. For spacecraft exposed to sunlight, like the MMO, the shadowed dielectric surfaces emit secondary electrons when high temperature electrons impinge of the surface. These secondary electrons are re-absorbed by conducting surfaces grounded to the less negatively charged sunlit side, thereby charging the spacecraft more (in the negative sense) as compared to a perfectly conducting spacecraft.

Dielectric surfaces could be implemented by splitting the arrays containing surface nodes into separate arrays for conducting surfaces and dielectric surface nodes. Alternatively, since the object module in PINC is able to simulate several objects at the same time, the spacecraft could be divided into separate objects with an additional label marking the object as either dielectric or conducting.

6.2 Implementation of additional charging currents

The photoelectron current is only one of several currents that have a significant impact on the floating potential, and plasma sheath formed around a spacecraft in a drifting plasma. Secondary electrons, discussed in Chapter 2, are especially important for spacecraft situated close to the sun where the plasma tends to be hotter. They affect both the floating potential, and depth of the potential barriers (Deca et al. 2013). The current due to backscattered electrons is another such important current that can in some cases exceed the current due incoming fluxes (Garrett 1981). These currents, in addition to the photoemission current,

could be included as a separate "extra current" module where the user of PINC could specify which current they want to be active as part of the input file.

6.3 Improving the photoemission current implementation

Several assumptions were made in how the photoemission current was implemented for this thesis. These assumptions can be addressed to improve the accuracy of the model. First, we assumed in chapter Chapter 3 that the incident angle of the sun to be perpendicular to sunlit surfaces. Reflectance, and thus photoelectric yield, are dependent on the incident angle of photons. As a first approximation the reduction in radiance can be modeled, using Lambert's cosine law, as proportional to the cosine of the incident angle.

Furthermore, the photoelectron cell injection algorithm is effective but inefficient: By necessity during implementation and debugging, several computations in the function are calculated each time the function is called. Extracting these computations into a structure variable could significantly reduce overall computation times for longer simulations.

Domain decomposition with the photoemission functionality can also be improved, it was mentioned briefly in section Chapter 3 that splitting the computational domain perpendicularly to the sun-spacecraft axis is not currently possible. If the split of the domain passes through the object to be simulated, another "sunlit" surface is exposed to the function that finds the nodes exposed to sunlight. If this function is rewritten to discard these cut surfaces, then domain decomposition becomes much more streamlined and the code can be more effectively run on a higher number of CPU's.

finally, the photoelectron emission code could be improved for the end user experience by separating the photoelectrons as a separate specie of the same charge and mass as the ambient electrons. PINC supports multi-species plasma, but as for now, the photoelectron injection code does not distinguish between ambient electron and photoelectron.

Appendices

APPENDIX A

Pseudocode of photoemission functions

Identification of sunlit surface nodes

Listing A.1: Pseudo code for finding sunlit object surface nodes

```
void oFindAllSunlitNodes(...){

    ... define and initialize variables ...

    for(int k = 0; k<size[z]; k++){

        for(int j = 0; j<size[y]; j++){

            int flag = 0;
            for(int i = 0; i<size[x]; i++){

                index = i*sizeProd[1] + j*sizeProd[2] + k*sizeProd[3];
                for(int n = 0; n < nSurfaceNodes; n++){

                    surfNode = surface[n];
                    if(surfNode == index){
                        ... append index to exposed node array ...
                        flag = 1;
                        break;
                    }

                if(flag) break;

                }
            }
        }
    }
}
```

Identification of cells adjacent to sunlit spacecraft surfaces

Listing A.2: C pseudocode for finding cells adjacent to sunlit object surfaces

```
void oCellFillingNodes(...){

    ... variable definitions and initialization ...

    for(long int i = surfaceOffset[a]; i<surfaceOffset[a+1]; i++){

        int nObjNode = 0;
        int yzPlane = 0;

        //cell nodes
        long int p = surface[i];
        long int pu = p + sizeProd[3];
        long int plu = p + sizeProd[2] + sizeProd[3];
        long int pl = p + sizeProd[2];
        long int po = p - sizeProd[1];
        long int pou = p - sizeProd[1] + sizeProd[3];
        long int poul = p - sizeProd[1] + sizeProd[3] + sizeProd[2];
        long int pol = p - sizeProd[1] + sizeProd[2];

        for(int j = interiorOffset[a]; j < interiorOffset[a+1]; j++){
            long int intNode = interior[j];
            if(po == intNode || pou == intNode ||
               poul == intNode || pol == intNode){
                nObjNode++;
            }
        }

        for(int j = surfaceOffset[a]; j < surfaceOffset[a+1]; j++){
            long int surfNode = surface[j];
            if(po == surfNode || pou == surfNode ||
               poul == surfNode || pol == surfNode){
                nObjNode++;
            }

            if(p == surfNode || pu == surfNode ||
               plu == surfNode || pl == surfNode){
                yzPlane++;
            }
        }

        if(yzPlane == 4 && nObjNode < 8){
            ... store index p as an emitting node for cell filling ...
        }
    }
}
```

Numerical integration of Planck's blackbody radiation law

Listing A.3: Numerical integration of Planck's law in terms of photon flux

```
void oPlanckPhotonIntegral(...){

  ... Variable and constant definitions and initialization ...

  for(int a=0; a<nObj; a++){
        //sigma[a] = ((double)sigma[a]) / (Planck * speedOfLight * 100.);
        double c1 = Planck * speedOfLight / Boltzmann;
        double x = c1 * 100. * (double)sigma[a]/temperature;
        double x2 = x * x;

        // decide how many terms of sum are needed
        double iterations = 2.0 + 20.0/x;
        iterations = (iterations < 512.) ? iterations : 512;
        int iter = (int)(iterations);
        // add up terms of sum
        double sum = 0.;
        for (int n=1; n<iter; n++) {
          double dn = 1.0 / n;
          sum += exp(-n*x) * (x2 + 2.0*(x + dn)*dn) * dn;
        }
        //result, in units of photons/s/m2/sr, convert to photons/timestep
        double kTohc = (Boltzmann * (double)temperature) / (Planck * speedOfLight);
        double solidAngle = (double)area[a] / pow(distFromSun,2.);
        radiance[a] = 2.0 * pow(kTohc,3.) * speedOfLight;
        radiance[a] = radiance[a] * sum;
        radiance[a] *= solidAngle * sunSurfaceArea;
        radiance[a] *= (1.0 - reflectance) * phYield;
        radiance[a] *= time;
  }

  ... Store radiance to object structure here ...

}
```

Listing A.3 shows a numeric implementation of equation (3.27) and equation (3.28). The original algorithm (*Calculating Balckbody Radiance* 2020) has been converted from C++ code to C for compatibility with the PINC framework.

Particle injection into cells adjacent to sunlit spacecraft surfaces

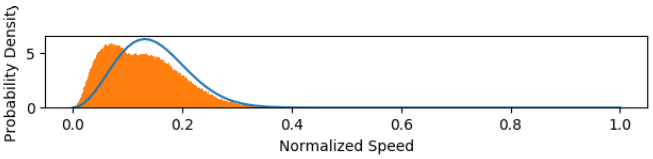
Listing A.4: C style pseudocode of cell filling algorithm

```
void pPhotoelectrons(...){

  ... initialize variables ...
  ... divide photoelectron flux by total number of cells to be filled ...

  for(int a=0; a < nObjects; a++){
    for(long int j=0; j<(long)flux[a]; j++){
      for(int i = 0; i < nEmittingNodes; i++){

        long int node = emittingNodes[emittingOffset[a] + i];
        gNodeToGrid3D(... node, pos ... ); //computes node location in grid
        memcpy(newPos, pos, pop->nDims * sizeof(*newPos));
        while(vel[0] >= 0.0){
          vel[0] = univariate_gaussian(... avgVel[a] ...); //normal velocity component
        }
        bivariate_gaussian( ...avgVel[a], &vel[1], &vel[2] ...); //tangential velocity components
        newPos[0] -= random(...); //subtract a random value in range [0,1]
        newPos[1] += random(...);
        newPos[2] += random(...);
        pNewParticle(...); //create the new particle
        ... reset "vel" array to zeroes ...
      }
    }
  }
}
```

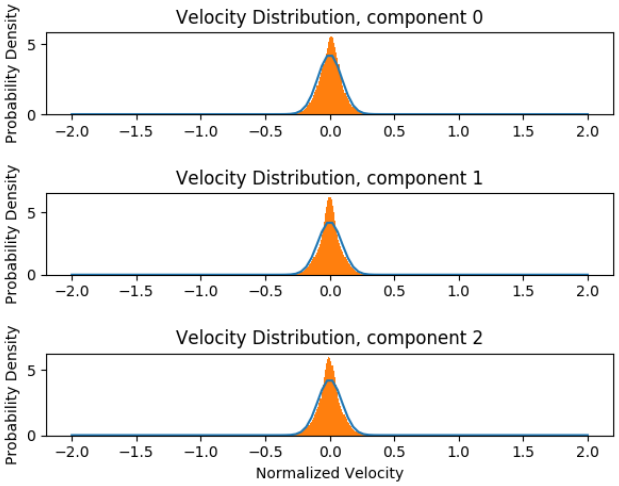
APPENDIX B

Electrons temperature distribution for MMO with

In this appendix, we plot the normalized velocity component distribution and normalized speed distribution for Case 8. The plots show the distribution at timestep 1,000 when the charging of the MMO has begun to stabilize and converge to the floating potential. The filled in orange sections show the binned particle distribution, while the blue lines show the normalized theoretical Gaussian distribution curves. Figure B.1 shows the distribution for the whole domain. Due to the large amount of particles in the whole domain, every hundredth electron was included for a total of 1.47×10^6 particles. From B.1a, it is apparent that the distribution contains a mixture of two electron distributions; the ambient photoelectrons with a higher average temperature, and the photoelectrons. Figure B.2 shows the normalized distributions only for particles close to the sunlit spacecraft surface. Note in figure B.2a the lower average temperature of of electrons close to the spacecraft surfaces.

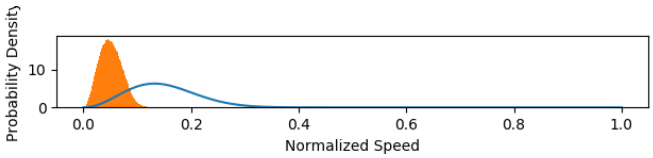


(a) Normalized distribution of speed of electrons

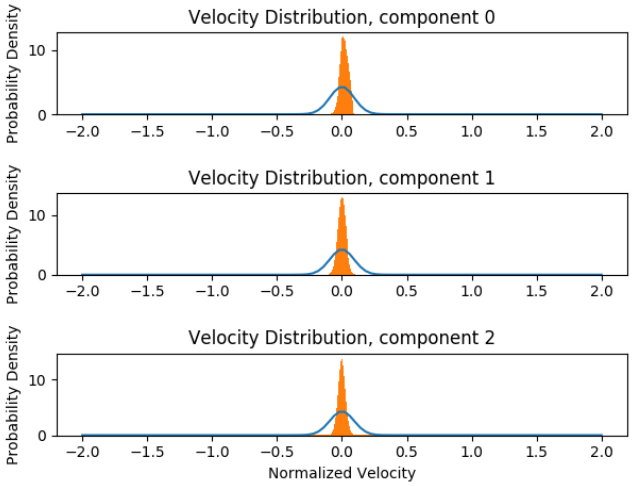


(b) Normalized distribution of velocity components of electrons

Figure B.1: Normalized distributions for both speed and velocity components in the whole computational domain



(a) Normalized distribution of velocity components for electrons close to the sunlit spacecraft surfaces



(b) Normalized distribution of speed for electrons close to the sunlit spacecraft surfaces

Figure B.2: Normalized distributions for both speed and velocity components for particles close to the spacecraft sunlit surfaces

APPENDIX C

Curve fitting of potential for MMO without photoemission

From Chapter 4 it became apparent from the timeseries plot of the potential of the MMO spacecraft in the cases where photoemission was not included that a convergence to a floating potential had not been achieved. In this appendix we present a Python script for fitting curves to these timeseries plots of potential in order to estimate the floating potential that PINC would eventually converge to.

Three functions were optimized to fit against the absolute value of the potential and compared for accuracy.

$$\text{Hill function: } y(t) = a\frac{x}{h+t} \tag{C.1a}$$

$$\text{Exponential function: } y(t) = a\exp\{-bt\} - c \tag{C.1b}$$

$$\text{Generalized logistic function } y(t) = A + \frac{K-A}{(C+Qe^{-Bt})^{\frac{1}{\nu}}}.. \tag{C.1c}$$

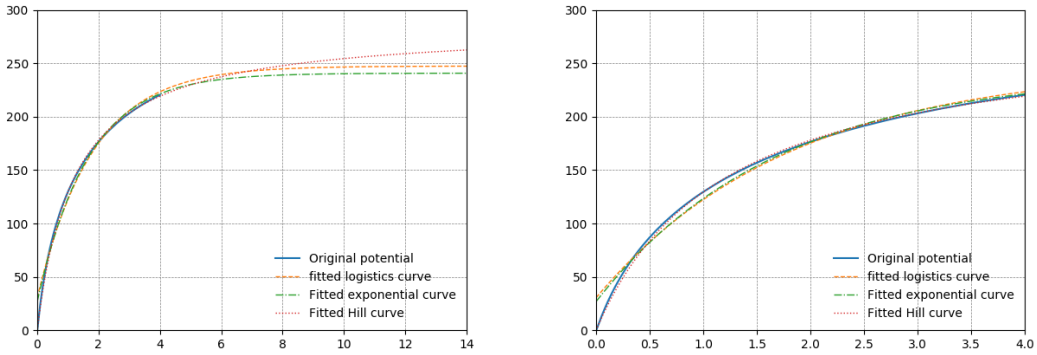
The three functions described all converge to a finite value, and the rate of growth can be easily controlled. The function parameters, except the independent variable t denoting time, are constants and can be adjusted for optimum fit. The Python package Scipy contains a curve fitting function. This curve fitting function uses a non-linear least squares algorithm to fit some function to a data set. This was combined with a trial and error checking of parameters since the optimization function did not converge for the exponential and generalized logistic function when all the controllable parameters were included.

The python script below also outputs the root mean square error, such that the fit of our trial functions could be checked numerically during trial and error testing. The root mean square error is described as:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}\left(y_{pred,i} - y_i\right)^2}{n}}. \tag{C.2}$$

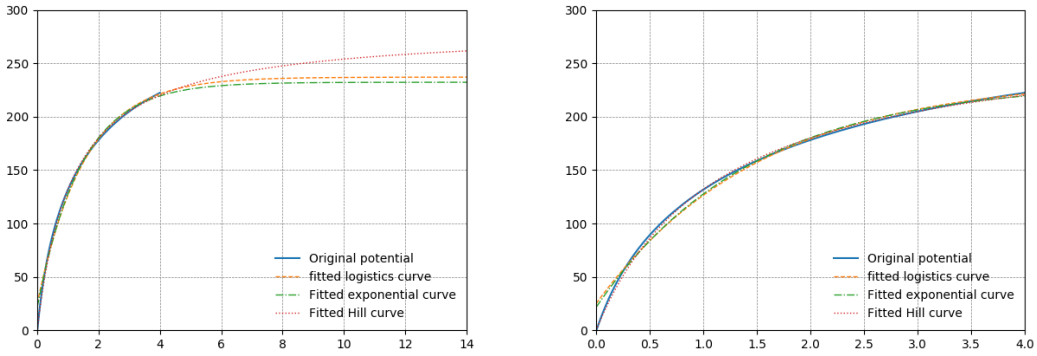
The absolute value of the potential found with PINC simulation was used to simplify fitting the trial functions to the data, the independent time variable t was also normalized by 10,000 timesteps to reduce errors from small floating point operations when computing the e^{-bt} term found in two of the trial functions. The Hill function was found to be the function that best fit the

curves of the absolute value of potential. The lowest RMSE values for the Hill fitting found that the potential converged to $-283\ V$ for the MMO with deployed booms, and $-281\ V$ for the MMO without deployed booms.



(a) 14,000 timesteps (b) 4,000 timesteps

Figure C.1: Curve fitting using a generalized logistic function, Hall function, and exponential function of the potential of the MMO spacecraft without booms. The Y axis is the absolute potential in Volts, and the X axis is units of 1,000 timesteps



(a) 14,000 timesteps (b) 4,000 timesteps

Figure C.2: Curve fitting using a generalized logistic function, Hall function, and exponential function of the potential of the MMO spacecraft with booms. The Y axis is the absolute potential in Volts, and the X axis is units of 1,000 timesteps

```
import numpy as np
import matplotlib.pyplot as plt
import scipy
import scipy.optimize as opt


def genLog(x, A, K, nu):
    y = A + ((K - A) / ((1 + 0.0001 * np.exp(-0.55*x))**(1/nu)))
    return y


def expFun(x,a,c):
    return a*np.exp(-0.6*x)-c


def hill(x,h):
    return 285 * x/(x+h)


def main():
    x_fit = np.linspace(0,4,num=40000)
    x = np.linspace(0,14,num=140000)
    pot = np.loadtxt('./potentials/noPH_NB_pot.txt', delimiter='\n')
    pot = np.absolute(pot)
    print(pot[0])

    popt, pcov = opt.curve_fit(genLog, x_fit, pot)
    print(popt)
    y_logistic = genLog(x, *popt)
    y_log_res = pot - genLog(x_fit, *popt)
    y_log_rmse = (scipy.sum(y_log_res**2)/(y_log_res.size-2))**0.5
    print(f"RMSE of general logarithmic fitting function = {y_log_rmse}")


    popt, pcov = opt.curve_fit(expFun, x_fit, pot)
    print(popt)
    y_exp = expFun(x, *popt)
    y_exp_res = pot - expFun(x_fit, *popt)
    y_exp_rmse = (scipy.sum(y_log_res**2)/(y_log_res.size-2))**0.5
    print(f"RMSE of exponential fitting function = {y_exp_rmse}")


    popt, pcov = opt.curve_fit(hill, x_fit, pot)
    print(popt)
    y_hill = hill(x, *popt)
    y_hill_res = pot - hill(x_fit, *popt)
    y_hill_rmse = (scipy.sum(y_hill_res**2)/(y_hill_res.size-2))**0.5
    print(f"RMSE of Hill fitting function = {y_hill_rmse}")
```

```
    fig, ax = plt.subplots()
    ax.set_xlim(0,14)
    ax.set_ylim(0,300)
    ax.grid(b=True, linestyle='--', linewidth=0.5, color='gray')
    ax.plot(x_fit,pot, label="Original potential")
    ax.plot(x,y_logistic, label="fitted logistics curve", lw=1, linestyle='--')
    ax.plot(x,y_exp, label="Fitted exponential curve", lw=1, linestyle='-.')
    ax.plot(x,y_hill, label="Fitted Hill curve", lw=1, linestyle=':')

    ax.legend(frameon=False, loc='lower right')
    plt.show()


if __name__ == "__main__":
    main()
```

Listing 1: Python script for optimizing curves fit to the timeseries plot of electric potential. The optimized curves are plotted forward in time past the total number of timesteps simulated using PINC in order to estimate the floating potential of the MMO when no photoemission is included.

APPENDIX D

Analysis on variations in photoelectric potential barrier

In this appendix, we investigate the uneven potential barrier on the sunlit panel perpendicular to the Sun-MMO vector where two distinct regions of relatively lower potential are found. We also investigate the prominent spikes in the plasma potential half way down the deployed booms where a region of relative low electron density can be found. Finally we comment on the presence of an small electric field gradient within the interior of the simulated spacecraft.

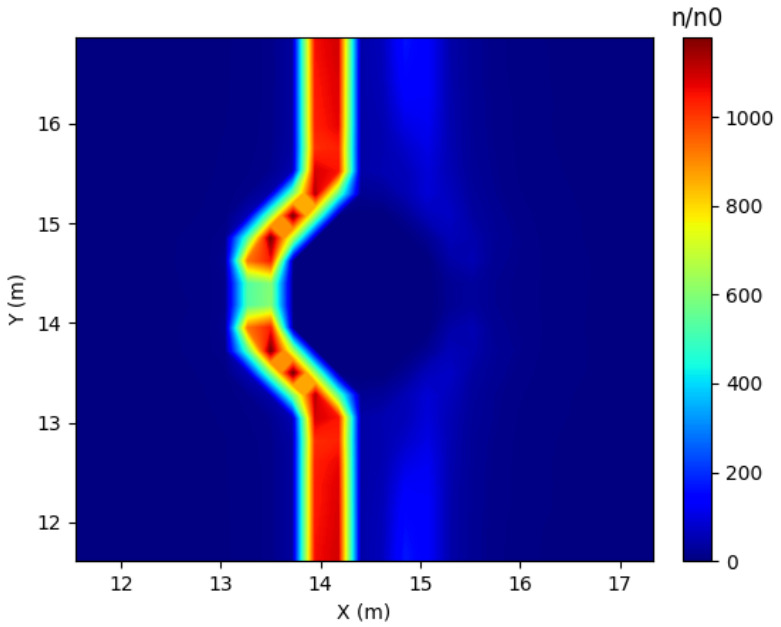


Figure D.1: Electron density profile n_e plotted in the XY axis. The X and Y axes have been restricted to the vicinity of the MMO octagonal main body, and the color gradients have been changed to better illustrate the spikes in electron density on the slanted panels

Figure D.1 shows a zoomed in 2D electron density profile in the vicinity of main body of the MMO spacecraft. Four symmetrical regions of very high electron density can be found on the sunlit side of the spacecraft. From section Chapter 3 we know from figure 3.13 that when we perform a voxelization of the spacecraft we are unable to produce diagonal straight edges in any plane. Instead, a zig-zag pattern approximates this diagonal line. This seems to be the cause of the issue illustrated in D.1; these four spikes in density correspond to empty "corner" cell along the diagonal spacecraft panel. The high density electrons cause a relatively high electric field in the local region after charge densities are interpolated to computational nodes. The panel perpendicular to the Sun-MMO vector shows a corresponding "hole" in the electron density. In the presence of a large radially outward electric field, photoelectrons injected into this region are quickly reabsorbed by the spacecraft surface.

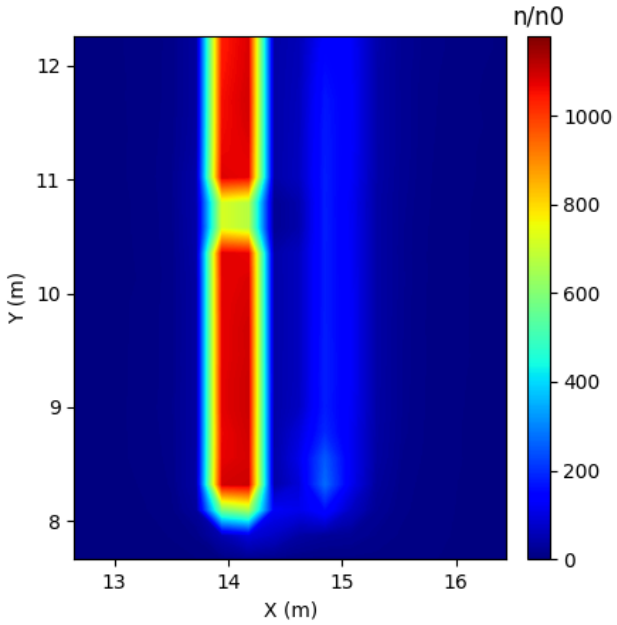


Figure D.2: Electron density profile n_e plotted in the XY axis. The X and Y axes have been restricted to the vicinity of the lower boom (in the Y direction), this density profile is symmetric about the XZ plane diving the MMO in half

Figure D.2 shows a zoomed in view of the electron density profile along the boom closest to the XZ. A small region of relatively low electron density (roughly 60% of the average electron density in the electron cloud). At least two possible reasons exist for this region of low density. First, the region could mark the point along the boom where the influence of the ambient plasma matches the influence of the boom and main body. This would cause the electron trajectories to diverge away from the region, causing a lower local density. Second, the location of these "holes" almost perfectly match the deliniation of the points along the Y axis where the computational domain has been decomposed. When

particles travel from one sub-domain to another, they are temporarly stored in a buffer in so called ghost cells. It is possible that some information loss happens over this boundary, especially if not enough memory has been allocated for these buffer array. This would also cause a decrease in the local electron density.

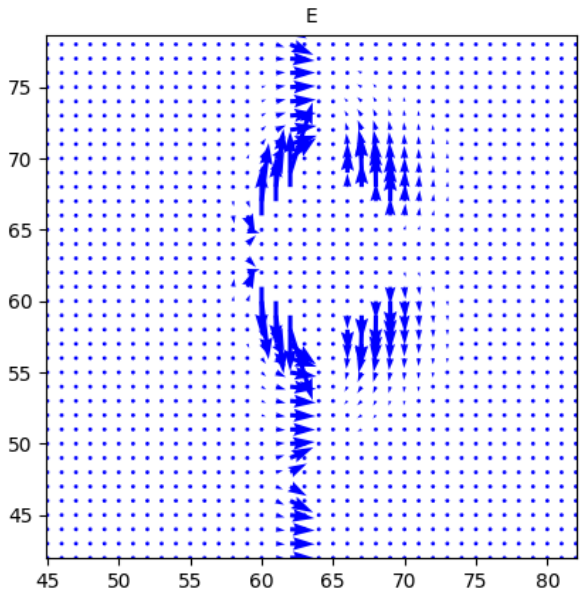


Figure D.3: Electrical field **E** drawn in the XY plane, with vector lines stretched to better illustrate the relative size of the field. The plot has been zoomed in to capture the field lines close to the spacecraft body, and the field lines along the boom where the electron density is low.

Figure D.3 illustrates the electric field lines close to the surface of the main body of the MMO. We see relatively long field lines extending from the region of high relative electron density, electron move in the opposite direction to the electric field lines, and so this plot suggests that electron will quickly reabsorbed by the spacecraft surface due to the force exterted by the local electrical field. Towards the bottoms of the plot, we also see diverging electrical field lines in the "hole" region of lower electron density along the lower boom. The opposing electrical field lines in this region forces electrons to scatter away from the region, which would explain the decrease in electron density, and the increase in relative potential in the region.

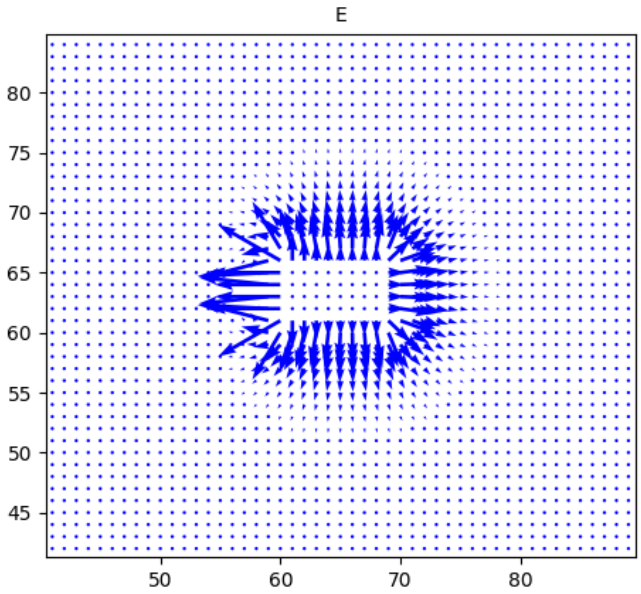


Figure D.4: Electrical field **E** drawn in the XZ plane, with vector lines stretched to better illustrate the relative size of the field.

Figure D.4 show the electrical field cut through the central body of the MMO in the XZ plane. A very large (relatively speaking) electrical field points radially out of the spacecraft. As mention previously, electrons will be pushed in the opposite direction to these field lines, directly into the spacecraft. The region therefore becomes a sink, electron quickly reabsorbing before new photoelectrons are emitted, causing a density "hole" to form.

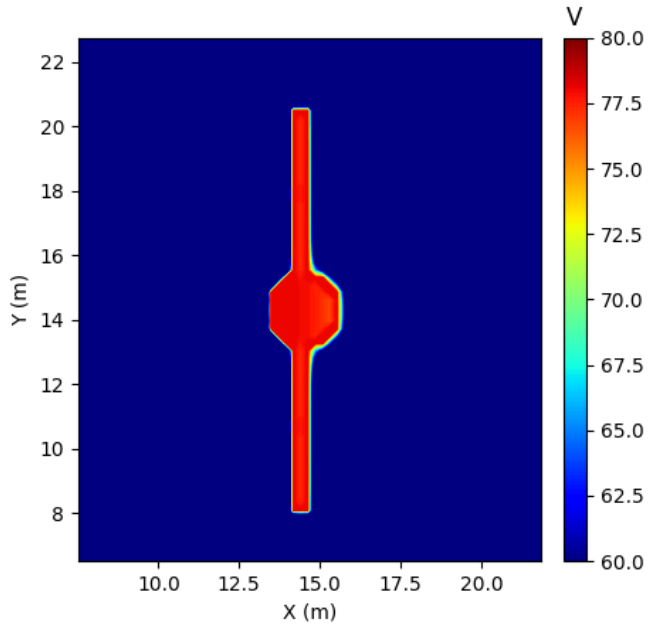


Figure D.5: Potential ϕ plotted at timestep 10,000. The maximum and minimum values of potential have been restricted to better display the potential gradient in the spacecraft interior

Figure D.5 shows a potential profile of the MMO cut in the XY plane, where Voltages have been restricted to values close to the spacecraft floating potential ϕ_f. From the plot we see that a small potential gradient has formed in the interior of the spacecraft. The capacitance matrix method attempts to redistribute charges to the spacecraft surface in such a way that there is an equipotential throughout the object. The gradient can not be explained by particles interior to the object that have yet to be removed from the computational domain, this is because the object-plasma module removes particles from the domain before the potential data is stored to file. Rather, this is most likely caused by numerical inaccuracies in the capacitance matrix: PINC takes the multigrid solver tolerance as an input matrix, and the capacitance matrix is formed by solving Poisson's equation for a single charge located at each surface node. It is likely if that adding a lower tolerance to the multigrid solver would smooth out the presence of this internal gradient at the cost of a higher total compute time.

Bibliography

Anderson, B. J. et al. (2010). "The Magnetic Field of Mercury." In: *Space Sci Rev* vol. 152, pp. 307–339. DOI: 10.1007/s11214-009-9544-3.

Benkhoff, J. et al. (2009). "BepiColombo-Comprehensive exploration of Mercury: Mission overview and science goals." In: DOI: 10.1016/j.pss.2009.09.020. URL: www.elsevier.com/locate/pss.

Benna, M. et al. (2009). "Modeling of the magnetosphere of Mercury at the time of the first MESSENGER flyby." In: DOI: 10.1016/j.icarus.2009.11.036.

Birdsall, C. K. and Langdon, A. B. (2004). *Plasma physics via computer simulation*, pp. 1–465. ISBN: 9781482263060.

Brask, S. M. (2018). *Instabilities in Collisional Plasmas Studied by Means of 3-D Kinetic Particle-In-Cell Simulations (master thesis)*. December.

Calculating Balckbody Radiance (2020). URL: https://www.spectralcalc.com/blackbody/CalculatingBlackbodyRadianceV2.pdf (visited on 04/21/2020).

Cartwright, K. L., Verboncoeur, J. P., and Birdsall, C. K. (2000). "Loading and Injection of Maxwellian Distributions in Particle Simulations." In: *Journal of Computational Physics* vol. 162, pp. 483–513. DOI: 10.1006/jcph.2000.6549.

Chen, F. F. (2018). *Introduction to Plasma Physics and Controlled Fusion*, E1–E1. ISBN: 9783319223087. DOI: 10.1007/978-3-319-22309-4_11.

Deca, J. et al. (2013). "Spacecraft charging analysis with the implicit particle-in-cell code iPic3D." In: *Physics of Plasmas* vol. 20, no. 10, pp. 1–9. ISSN: 1070664X. DOI: 10.1063/1.4826951.

Einstein, A. (1905). "Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt." In: *Annalen der Physik* vol. 322, no. 6, pp. 132–148. ISSN: 00033804. DOI: 10.1002/andp.19053220607. URL: http://doi.wiley.com/10.1002/andp.19053220607.

ESA (2020a). *ESA - Solar spectrum*. URL: https://www.esa.int/ESA%7B%5C_%7DMultimedia/Images/2017/12/Solar%7B%5C_%7Dspectrum (visited on 06/15/2020).

— (2020b). *ESA Science & Technology - Fact Sheet*. URL: https://sci.esa.int/web/bepicolombo/-/47346-fact-sheet (visited on 06/15/2020).

ESA Science & Technology - BepiColombo plasma simulation (2020). URL: https://sci.esa.int/web/bepicolombo/-/60190-bepicolombo-plasma-simulation (visited on 03/11/2020).

Feuerbacher, B. and Fitton, B. (1972). "Experimental investigation of photoemission from satellite surface materials." In: *Journal of Applied Physics* vol. 43, no. 4, pp. 1563–1572. ISSN: 00218979. DOI: 10.1063/1.1661362.

Fitzpatrick, R. (2015). *Plasma physics: an introduction*. Boca Raton, Fla.: CRC Press, Taylor & Francis Group. ISBN: 9781466594265.

Garrett, H. B. (1981). "The charging of spacecraft surfaces." In: *Reviews of Geophysics* vol. 19, no. 4, pp. 577–616. ISSN: 19449208. DOI: 10.1029/RG019i004p00577.

Hockney, R. W. and Eastwood, J. W. (1988). *Computer simulation using particles*. eng. Bristol: Institute of Physics Publishing. ISBN: 0852743920.

Holta, V. (2018). *Dynamics of a Plasma Blob Studied With Particle-In-Cell Simulations (master thesis)*.

Hutchinson, I. H. (2002). *Principles of plasma diagnostics*. 2nd ed. Cambridge: Cambridge University Press. ISBN: 0521803896.

Jacobsen, K. S. et al. (2010). "A new Langmuir probe concept for rapid sampling of space plasma electron density." In: *Measurement Science and Technology* vol. 21, pp. 85902–85912. DOI: 10.1088/0957-0233/21/8/085902.

Killie, G. V. (2016). *A Parallel Multigrid Poisson Solver for PINC , a new Particle-in-Cell Model (master thesis)*.

Lai, S. T. (2003). "A Critical Overview on Spacecraft Charging Mitigation Methods." In: *IEEE Transactions on Plasma Science* vol. 31, no. 6, pp. 1118–1124. ISSN: 0093-3813. DOI: 10.1109/TPS.2003.82096.

— (2005). "Charging of mirror surfaces in space." In: *Journal of Geophysical Research: Space Physics* vol. 110, no. A1, pp. 1–11. ISSN: 21699402. DOI: 10.1029/2002JA009447.

— (2019). *Introduction to Spacecraft Charging*. DOI: 10.2307/j.ctvcm4j2n.5.

Lai, S. T. and Tautz, M. T. (2006). "Aspects of spacecraft charging in sunlight." In: *IEEE Transactions on Plasma Science* vol. 34, no. 5 II, pp. 2053–2061. ISSN: 00933813. DOI: 10.1109/TPS.2006.883362.

Lapenta, G. (2011). "Particle simulations of space weather." In: DOI: 10.1016/j.jcp.2011.03.035. URL: http://perswww.kuleuven.ac.be/%7B~%7Du0052182/.

Leach, R. and Alexander, M. (1995). *Failures and Anomalies Attributed to Spacecraft Charging*. Tech. rep.

Luhmann, J. G., Ledvina, S. A., and Russell, C. T. (2004). "Induced magnetospheres." In: *Advances in Space Research* vol. 33, no. 11, pp. 1905–1912. ISSN: 02731177. DOI: 10.1016/j.asr.2003.03.031.

Marholm, S. (2019). "The unstructured particle-in-cell method with applications for objects ionospheric plasmas." PhD thesis. Oslo.

Meyer-Vernet, N. (2007). *Basics of the Solar Wind*. ISBN: 9780521814201. DOI: 10.1017/cbo9780511535765.

Miloch, W. J. (2015). "Simulations of several finite-sized objects in plasma." In: *Procedia Computer Science*. DOI: 10.1016/j.procs.2015.05.313.

Miloch, W. J. et al. (2012). "Spacecraft charging in flowing plasmas; numerical simulations." In: *Journal of Physics: Conference Series* vol. 370, no. 1. ISSN: 17426596. DOI: 10.1088/1742-6596/370/1/012004.

Miyake, Y., Cully, C. M., et al. (2013). "Plasma particle simulations of wake formation behind a spacecraft with thin wire booms." In: *Journal of Geophysical Research: Space Physics* vol. 118, no. 9, pp. 5681–5694. ISSN: 21699402. DOI: 10.1002/jgra.50543.

Miyake, Y. and Usui, H. (2009). "New electromagnetic particle simulation code for the analysis of spacecraft-plasma interactions." In: *Phys. Plasmas* vol. 16, p. 62904. DOI: 10.1063/1.3147922. URL: https://doi.org/10.1063/1.3147922.

Mott-Smith, H. M. and Langmuir, I. (1926). "The theory of collectors in gaseous discharges." In: *Physical Review* vol. 28, no. 4, pp. 727–763. ISSN: 0031899X. DOI: 10.1103/PhysRev.28.727.

Mullen, E. G. et al. (1986). "SCATHA survey of high-level spacecraft charging in sunlight." In: *Journal of Geophysical Research* vol. 91, no. A2, p. 1474. ISSN: 0148-0227. DOI: 10.1029/JA091iA02p01474. URL: http://doi.wiley.com/10.1029/JA091iA02p01474.

NASA (2020). *SCATHA P78-2*. URL: https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=1979-007A (visited on 05/27/2020).

Nobel Media AB (2020). *The Nobel Prize in Physics 1921*. URL: https://www.nobelprize.org/prizes/physics/1921/summary/ (visited on 05/24/2020).

Okuda, H. et al. (1979). *SPLINES AND HIGH ORDER INTERPOLATIONS IN PLASMA SIMULATIONS*. Tech. rep., pp. 227–231.

Paulsson, J. J. et al. (2019). "Effects of booms of sounding rockets in flowing plasmas." In: *Physics of Plasmas* vol. 26, no. 3. ISSN: 10897674. DOI: 10.1063/1.5051414. URL: http://dx.doi.org/10.1063/1.5051414.

Piel, A. (2017). *Plasma Physics: An Introduction to Laboratory, Space, and Fusion Plasmas*. Tech. rep. URL: http://www.springer.com/series/8431.

Planck, M. (2020). *The theory of heat radiation*. URL: https://babel.hathitrust.org/cgi/pt?id=uc1.b3740331%7B%5C&%7Dview=1up%7B%5C&%7Dseq=111 (visited on 03/02/2020).

Qin, H. et al. (Aug. 2013). "Why is Boris algorithm so good?" In: *Physics of Plasmas* vol. 20, no. 8, p. 084503. ISSN: 1070-664X. DOI: 10.1063/1.4818428.

Rothery, D. A. (2015). *Planet Mercury: From a Pale Pink Dot to Dynamic World*, p. 191. ISBN: 9783319121161. DOI: 10.1007/978-3-319-12117-8.

Saito, Y. et al. (2010). "Scientific objectives and instrumentation of Mercury Plasma Particle Experiment (MPPE) onboard MMO." In: *Planetary and Space Science* vol. 58, no. 1-2, pp. 182–200. ISSN: 00320633. DOI: 10.1016/j.pss.2008.06.003. URL: http://dx.doi.org/10.1016/j.pss.2008.06.003.

Sjögren, A., Eriksson, A. I., and Cully, C. M. (2012). "Simulation of potential measurements around a photoemitting spacecraft in a flowing plasma." In: *IEEE Transactions on Plasma Science* vol. 40, no. 4, pp. 1257–1261. ISSN: 00933813. DOI: 10.1109/TPS.2012.2186616.

Slavin, J. A. et al. (2012). "MESSENGER and Mariner 10 flyby observations of magnetotail structure and dynamics at Mercury." In: *J. Geophys. Res* vol. 117, p. 1215. DOI: 10.1029/2011JA016900.

Solanki, S. K. (2003). *Sunspots: An overview*. DOI: 10.1007/s00159-003-0018-4.

Taylor and Francis Group (2019). *Handbook of Chemistry and Physics 100th Edition*. URL: http://hbcponline.com/faces/contents/ContentsSearch.xhtml (visited on 05/25/2020).

Trottenberg, U., Oosterlee, C. W., and Schuller, A. (2000). *Multigrid*. URL: https://books.google.no/books?id=-og1wD-Nx%7B%5C_%7DwC%7B%5C&%7Dprintsec=frontcover%7B%5C&%7Ddq=trottenberg+2000%7B%5C&%7Dhl=en%7B%5C&%7Dsa=X%7B%5C&%7Dved=0ahUKEwjYkqSX35zpAhWS7KYKHftjD9AQ6AEIOjAC%7B%5C#%7Dv=onepage%7B%5C&%7Dq=trottenberg%202000%7B%5C&%7Df=false (visited on 05/05/2020).

Verboncoeur, J. P. (2005). "Particle simulation of plasmas: Review and advances." In: *Plasma Physics and Controlled Fusion*. Vol. 47. 5 A. DOI: 10.1088/0741-3335/47/5A/017.

Wade, M. (2020). *SCATHA*. URL: http://www.astronautix.com/s/scatha.html (visited on 03/10/2020).

Whipple, E. C. (1981). "Potentials of surfaces in space." In: *Reports on Progress in Physics* vol. 44, no. 11, pp. 1197–1250. ISSN: 00344885. DOI: 10.1088/0034-4885/44/11/002.

Widger, W. K. and Woodall, M. P. (1976). "Integration of the Planck Blackbody Radiation Function." In: *Bulletin of the American Meteorological Society* vol. 57, no. 10, pp. 1217–1219. ISSN: 0003-0007. DOI: 10.1175/1520-0477(1976)057<1217:iotpbr>2.0.co;2.

Yamakawa, H. et al. (2008). "BepiColombo Mercury magnetospheric orbiter design." In: *Acta Astronautica* vol. 62, pp. 699–705. DOI: 10.1016/j.actaastro.2008.01.040. URL: www.elsevier.com/locate/actaastro.

Zhao, H. et al. (1996). *Self-consistent determination of the electrostatic potential barrier due to the photoelectron sheath near a spacecraft*. Tech. rep. A7, pp. 15653–15659. DOI: 10.1029/96JA00322.