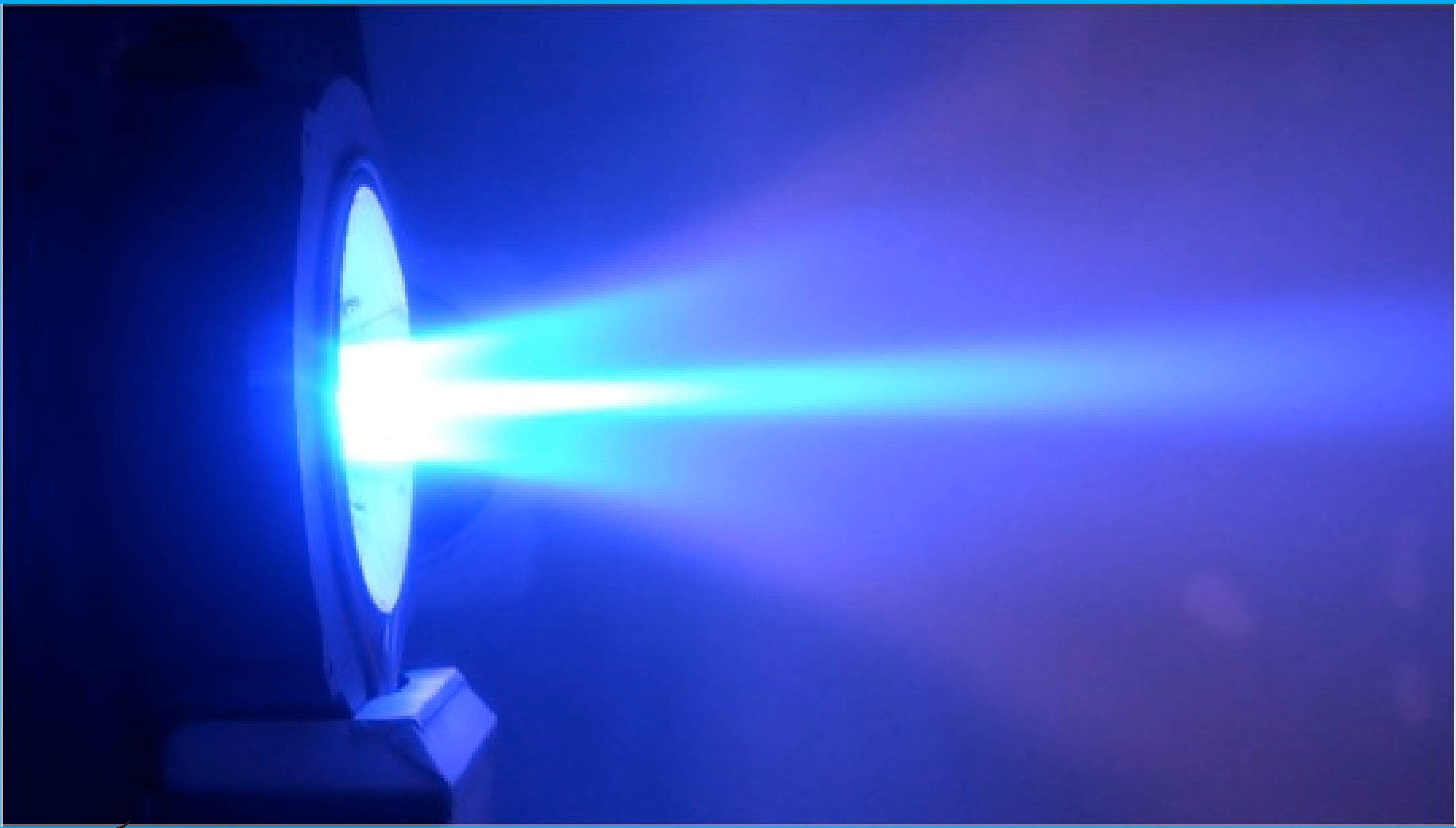
Assessment Avenues of Plasma Modelling Codes for MPD Thrusters

Tanase-Marius Manafu



Assessment Avenues of Plasma Modelling Codes for MPD Thrusters

by

Tanase-Marius Manafu

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Friday May 6, 2022 at 13:30.

Student number:	5163382	
Project duration:	July, 2021 – April, 2022	
Thesis committee:	Dr. Angelo Cervone,	TU Delft (supervisor & chair)
	Dr. Stefano Speretta,	TU Delft (examiner)
	Ir. Marc C. Naeije,	TU Delft (external examiner)
	MEng. Marcus Collier-Wright,	NSS (co-supervisor)

Front cover image: With permission from Institute of Space Systems, University of Stuttgart

Preface

This thesis project was conducted in challenging times, but that allowed me to be more resilient and adapt easier. While I had to manage the expectations of both the TU Delft side and the NSS startup I did my thesis with, I had a few people who helped me see the light at the end of the tunnel throughout the project.

I would like to thank my TU Delft supervisor, Dr. Angelo Cervone, who had supported me from the beginning in my insistent desire to conduct my thesis in the field of electric space propulsion. He has also supported me mentally when things got rough and provided invaluable feedback at critical moments, while giving me ample freedom in my day to day activities.

Another vital person whom I have to the thank is my NSS co-supervisor, Mr. Marcus Collier-Wright. He helped me get a sense of how projects are in real life and how I should consider many factors when it comes to the success of an idea. In addition, he supported me in focusing on one idea at a time and to not diverge too much.

Of high importance when it came to the last part needed for the thesis, plasma simulations, was the help from SPARC Industries, especially Dr. Dejan Petkow. Furthermore, with regards to the plasma simulations, another important mention is Jorge Alberto Garcia-Perez, whose help was essential in conducting the simulations.

Two other people I learned about the business environment technicalities from, are Mr. Manuel La Rosa Betancourt and Mr. Freddy Torres. I would also like to thank two of my colleagues at NSS, with whom I greatly enjoyed collaborating and exchange ideas, Jaime Martin Lozano and Kapish Aggarwal.

Tanase-Marius Manafu
Delft, March 2022

Summary

There is an increasing interest in higher thrust space propulsion technologies that do not face the downsides of the traditional chemical propulsion. As more and more missions aim to go to the moon, Mars and other further away planets and as travelling time faces space radiation, electric propulsion technologies with higher thrust can decrease the time required and reduce fuel consumption. In addition, satellites can be much more lightweight. The technology of focus in this paper, the applied-field magnetoplasmadynamic (MPD) thruster (AF-MPDT) was developed in the 1960s. This technology plays on these benefits, potentially providing more value than current electric propulsion thruster. However, little research has been conducted for this thruster. There is still the need for an accurate model describing the thruster's functioning, its thrust and the physical phenomenons inside. In addition, a vital part of the development, the plasma simulations have not seen great improvement either, with most codes using the magnetohydrodynamics approach in a 2D manner and without any focus on other aspects besides the thrust. There is growing interest in other, more efficient approaches, in terms of simulating a wider range of phenomenons in the plasma.

This project was conducted together with the Neutron Star Systems start-up, who are developing the SUPREME AF-MPDT. This thruster incorporates new technologies and operation approaches making the older simulation codes less suitable. In addition, the use of simulations has the potential to greatly reduce time-to-market and prototype costs by the use of Digital Twins, which is the virtual representation of the thruster. Thus, the need for the very expensive AF-MPDT laboratory testing is reduced, new plasma models are tested faster and the operation parameters are optimised. Therefore, the goal of this project was to identify the most suitable plasma simulation software for SUPREME that can be used in the Digital Twin of an AF-MPD thruster, supported by a set of requirements and a number of important parameters.

To report started with the conceptualisation stage, where the mission need and statement were stated. The systems engineering approach was further developed with the top-level requirements, which were the basis for all the lower level requirements. This was followed by the constraints of the project and the acceptance criteria. The next chapter was aimed at providing the theoretical support for the systems engineering work and delved into some of the most important aspects for SUPREME. The first aspect approached was the electron fall voltage and the corresponding power deposition that decreases the thruster efficiency while increasing erosion of the area. Both electrodes were analysed with the anode region containing most of the losses. An AF-MPD thruster provides a freedom of choice in propellants compared with other technologies such as ion and Hall thrusters. Multiple propellants were briefly analysed, such as Argon and complex molecules. Erosion represents a major concern and the phenomenon of sputtering was also analysed, as it changes the geometry of the thruster and it can potentially have a strong impact on a long term functioning thruster. To better understand the software needs, there was also focus on the operational parameters and their connection, especially on the relationship between thrust and other parameters. Operating modes such as steady-state, pulsed were compared. A new cathode prototype was shown to have potential in using it for SUPREME, while the topic of plume identification in space for defence purposes was touched. In addition to the discussion about operating parameters, there was attention given to the inputs and outputs of parameters such as the current electric field, magnetic Reynolds number etc.

In the second half of the theory chapter the aspects approached were the nature of the simulaiton code. Thus, the a comparison between particle-in-cell with Monte Carlo (PIC-MCC) collisions and magnetohydrodynamics approaches was analysed. The two most important differences came from the nonlocal effects and the plasma transport coefficients. The PIC-MCC approach was then further split into two sections: fully kinetic and hybrid simulations, which were also compared. In support of the accuracy of fully kinetic simulations, the option of computing the mean free path and the collision cross-section at each interaction were presented. Ionisation is another important aspect tackled, as in the case of

fully kinetic simulations particles can be ionised in a more natural manner. Then, an analysis of the electric and magnetic fields was conducted based on previous experiments and simulations. From research it was noticed that on average the electron temperature is 2-3 eV, but it peaks in the bulk of the plasma and are linearly correlated with the magnetic field. For the ions however, it was shown that this correlation was weak and that on average their temperature is 0-12 eV. Based on plasma simulations conducted by researchers the relationship between the velocity components and the main operating parameters was explained, while the link between the electrons radial component of velocity and current was emphasised. The chapter ended with the results of two full PIC models.

The fourth chapter started with the definition of multiple systems engineering notions such as the test types, methods, levels in the context of the verification and validation approaches. The focus was on the four verification methods of the project: test, analysis, inspection and demonstration. The validation methods, while not the main focus, were also introduced. The test types presented were: functional, integration and environmental tests, while the test levels introduced were: development, qualification, acceptance and operation tests.

Then the lower level requirements were all presented here. Each requirement was marked with the corresponding verification method(s), the type of requirement (functional, performance, interface, constraint and design) and they had multiple identifiers: time, acceleration mechanisms, thrust accuracy, losses accuracy, erosion accuracy, anode fall voltage, user interface, cost, high applied magnetic field, magnetic Reynolds number and diagnostic. After they were stated, each requirement was motivated and their verification method(s) was/were discussed.

The second half of the chapter introduced additional systems engineering notions. The Context Diagram was used to show the connections between each part of a plasma simulation software project. This was followed by a detailed Input-Output matrix that would be expected for the chosen simulation software. As an example, a short Sequence Diagram was presented. In the final part, additional notions were presented, such as possible tests for the validation methods. In addition, for the longer term strategy of this project, the topics of fault mitigation and costs were also explained and how the interactions with the chosen software developing stakeholder should proceed.

In the fifth chapter, three plasma simulation softwares were described and then compared. In addition, the cost options were also detailed as they play an important role, especially for a small start-up such as NSS. As such, a trade-off comparison was conducted between the three options based on three criteria: time for development, cost of achieving a fully operational software and hardware required. This trade-off resulted in multiple suggestions that depend on the resources NSS has or would have at their disposal.

In the second half of this chapter one of the software options was used for a simple plasma sheath simulation for an example case and a test case. This involved many changes in the parameters of the simulation to ensure that the results have a physical significance and that the time required is reduced. The two cases were thoroughly described and three assumptions for good collisionless plasma sheath results were stated. The results of the simulations were compared against the theoretical case represented as plots for the plasma density and potential near a conductive sheath. For a complete image of the simulations and to ensure the steady-state was reached, the following parameters were analysed: the electric potential, the electric field, the density of the charged species and the current flow at the conductive sheath. The chapter ended with an analysis of the results and by explaining the relevance of these simulations for choosing a plasma simulation stakeholder and for the requirements designed.

In the final chapter, the results of the report are summarised. In addition, the research questions are answered. Using these sections, the future work was then detailed.

Contents

Nomenclature

List of Abbreviations

Abbreviation	Complete meaning
PPU	Power Processing Unit
ADCS	Attitude Determination and Control System
S/C	Spacecraft
GIT	Gridded Ion Thruster
HET	Hall Effect Thruster
MPD	Magnetoplasmadynamic
MPDT	MPD Thruster
SF-MPD	Self-Field MPD
AF-MPD	Applied-Field MPD
HTS	High-Temperature Superconductors
NSS	Neutron Star Systems
SUPREME	SUPerconductor based Readiness Enhanced Magnetoplasmadynamic Electric propulsion
EP	Electric Propulsion
TRL	Technology Readiness Level
MC	Monte Carlo
MCC	MC Collisions
DSMC	Direct simulation Monte Carlo
PIC	Particle-in-Cell
SNR	Signal to Noise Ratio
GUI	Graphical User Interface
BC	Boundary Conditions
ESM	Extended Scaling Model
MHD	Magnetohydrodynamics
AFV	Anode Fall Voltage
CFV	Cathode Fall Voltage
EMF	Electromotive Force
SEE	Secondary Electron Emission
V&V	Verification and Validation
SE	Systems Engineering

List of Symbols

Abbreviation	Complete meaning	SI Units
g_0	Gravitational acceleration	[m/s^2]
η	Efficiency	-
$\dot{m}$	Mass flow rate	[kg/s]
I_{sp}	Specific impulse	[s]
a_0	Sonic velocity	[m/s]
$\vec{E}$	Electric field vector	[V/m]
$\vec{B}$	Magnetic field vector	[T]
r_L	Larmor radius	[m]
λ_D	Debye length	[m]
ω_p	Plasma frequency	[rad/s]
k or k_B	Boltzmann constant	[J/K]
ϵ	Absolute permittivity	[F/m] = $[\mathrm{C^2 \cdot s^2}/(\mathrm{kg \cdot m^3})]$
ϵ_0	Vacuum permittivity	[F/m] = $[\mathrm{C^2 \cdot s^2}/(\mathrm{kg \cdot m^3})]$
ϵ_r	Relative permittivity	[F/m] = $[\mathrm{C^2 \cdot s^2}/(\mathrm{kg \cdot m^3})]$
μ	Permeability	[H/m] = $[\mathrm{kg \cdot m}/(\mathrm{s^2 \cdot A^2})]$
μ_0	Vacuum permeability	[H/m] = $[\mathrm{kg \cdot m}/(\mathrm{s^2 \cdot A^2})]$
μ_r	Relative permeability	[H/m] = $[\mathrm{kg \cdot m}/(\mathrm{s^2 \cdot A^2})]$
$\vec{j}$	Current density vector	[A/m^2]
ν_e	Collision frequency per electron	[s^{-1}]
σ	Electrical conductivity	$[\mathrm{C^2 \cdot s}/(\mathrm{kg \cdot m^3})]$
v_A	Alfvén velocity	[m/s]
β_{Hall}	Hall parameter	-
β_{plasma}	Plasma parameter	-
R_m	Magnetic Reynolds Number	-

List of Figures

List of Tables

1 Introduction

According to the thesis proposal, the work detailed in this report was focused on the systems engineering aspects of the plasma simulation code for an applied-field Magnetoplasmadynamics (AF-MPD) thruster, which is one of the most powerful propulsion technologies and was developed in the 1960s. However, progress has been limited on multiple subareas of research as higher Technology Readiness Level (TRL) was achieved with other electric propulsion technologies. Those thruster technologies were also understood better in terms of the physical processes inside and in terms of plasma simulation.

While the AF-MPD technology achieved some theoretical and experimental foundations, the plasma simulation tools remain limited and not accurate in simulating the thrust, the erosion and processes inside. The Neutron Star Systems (NSS) company is working towards developing an advanced AF-MPD thruster that gather advanced technologies from other fields as well, such as the use of superconductors. This further enlarges the rift between the flight capability of their SUPREME thruster and the other technologies such as Hall and ion thrusters. As such, considerable resources must be focused on understanding the physics behind this new thruster and the simulation capabilities. As explained in the Literature Study the experimental tests for AF-MPDTs are more expensive and harder to be conducted than other technologies, which puts incredible strains on a funds limited start-up such as NSS. To account for this there is more emphasis put on the simulations within NSS and similar competitors.

Thus, the motivation for this thesis project was given by the need of a plasma simulation software designed according to the needs of NSS. Developing such a novel code would be a monumental achievement requiring extensive knowledge and a medium to long time-range. This task was supported in this project by interacting with multiple relevant stakeholders to design the needs, acceptance criteria, top-level and lower level requirements as a first step towards choosing a software option. In addition, an analysis into the most important physical processes occurring in an AF-MPD was conducted. This provided essential support in the design of requirements and an essential understanding in the communication with stakeholders. This would allow the chosen team developing the plasma simulation code to focus and structure the code accordingly. The systems engineering work conducted for this project represents the initial stage/starting point which would be supplemented by choosing a stakeholder to develop the software. The report provided an analysis into the most suitable choices available. This would be followed by NSS engaging with the corresponding stakeholder and agreeing on further lower level requirements and other aspects such as cost, time needed, hardware needed.

1.1. Research Questions

The thesis work conducted was structured around a main research question:

What are the requirements of a plasma simulation software to describe as accurately as possible the acceleration mechanisms within a state of the art AF-MPDT such as the SUPREME thruster?

To build a clear path towards answering the main question, the following research sub-questions were also designed:

- What is the current state of AF-MPD research?
- Is a 3D model preferred over a 2D one to achieve thrust, erosion, losses results with errors of up to 5% from the values in experiments?
- What are the plasma simulation codes applicable for the AF-MPD developed by the NSS?
- How would a simulation software tackle the backtracking phenomenon of charged particles?
- What are the boundary conditions?
- How would the requirements be verified?
- Can open-source codes be useful for the AF-MPDT simulations?

If more time was available or this project would be continued with emphasis on the physics of the AF-MPDT, the following research questions would be attempted:

- How to solve the strong electron current from the cathode in the context of plasma simulation?
- How does the processes in the magnetic nozzle influence the plasma simulation code?
- How is the swirl acceleration influenced by the magnetic nozzle?
- How is the thrust influenced by the back-tracking of electrons outside the thruster?
- How is the confinement influenced by the magnetic nozzle shape?
- What role does the ambipolar field have?

The main research question comes from the need for NSS to have the first steps in choosing and designing the plasma simulation software. As previously explained, this would have many benefits in terms of reducing costs and time-to-market, as well as testing new physics models.

The research sub-questions start with a natural continuation from the Literature Study. Given the theoretical nature of the SUPREME thruster, the most up to date aspects were researched. The difference between different dimensionality codes represents an important point in the race for better simulations, especially as most solvers were developed around a 2D geometry. There are a number of plasma simulation codes, however, not all of them are suitable for the SUPREME case. To further increase the utility of the SUPREME thruster, an AF-MPDT should have a wide variety of applications. As such, there is increasing interest on the backtracking phenomenon of charged particles and their impact on the rest of the spacecraft. The question about the domain boundaries stems from the high computational power needed. Thus, if only a certain part of the thruster is simulated, the simulated domain is reduced and the software can run faster. In order to ensure the quality of the requirements which are part of the main research question, there have to be clear processes explained on how the requirements are going to be verified. The third part of the report was structured around simulations from an open-source code of simple cases. This would shed light on whether open-source codes can be potentially useful for SUPREME.

1.2. Outline

The thesis started with the design of the mission need and statement, top-level requirements, the constraints and acceptance criteria, as seen in Section 2. This chapter introduces the reader in the fundamental aspects of the systems engineering work and represent the basis for deciding on the software specifications. Section 3 is a continuation of the Literature Study, while tackling the most important theoretical aspects of an AF-MPDT, for designing the requirements. Going into more details, Section 4 explains the different tests types, methods. This chapter then states all the lower level requirements, followed by their reasoning and verification methods. The rest of the chapter introduces additional aspects, such as the input/output matrix and topics that shall be considered after a software choice is made. Section 5 introduces a number of software choices and explains in detail their capabilities and the differences between them. This part was built mostly from extensive meetings with the corresponding stakeholders. Different suggestions for the appropriate choices are done, according to the available resources for NSS. The second part of the chapter explains an example case and a test case and the software from one of the investigated options is used to analyse its capabilities on basic plasma sheath cases. Lastly, Section 6 summaries the main findings from this report, answers the research questions and describes the meaning of their results. In addition, possible future work is stated.

2

Conceptualisation

In this section the top-level requirements, constraints and the most important physical processes occurring in an AF-MPD were designed. The three branches of applied systems engineering used in this project are processes, methods and models. The first was the responsibility of the thesis and it meant the design of the structure for the simulation software. By software it is meant both the graphical user interface (GUI) and the kernel that executes the simulation and the computations involved.

The methods are represented by the plasma simulation software that shall be provided by a company and the models for the simulation resulted from both the work conducted for the thesis and the academic knowledge. These tasks can be expressed in the following mission need and mission statement:

The different parameters of the NSS thruster and the related components will be designed such that the desired output is maximised. These components need to be simulated as an integrated system to ensure the customer needs are respected.

The plasma simulation software will provide the outputs from changes in the thruster system design and will integrate within the Digital Twin.

First, the stakeholders were identified for a kinetic plasma simulation code for AF-MPD thrusters. Thus, the stakeholders were divided into active (will interact with the software after it is developed), passive (will influence the software, but will not be involved after its full completion) and customers, which are similar to passive stakeholders.

Thus, NSS is the type of active stakeholder that is a customer and it also has the highest priority. It interacts with the software after its development and it would pay for a license. Other active stakeholders include all the companies that develop electric space propulsion and space agencies, such as the American, Chinese and Russian agencies, which are also developing an AF-MPD. The most important type of passive stakeholders are academic institutions. This is due to the cutting edge nature of the field, which requires the plasma simulations developing companies to take into account, correctly, the physical processes in thrusters. Other passive stakeholders include space propulsion laboratories which can provide experimental data to plasma simulation companies to test the accuracy and validity of their code.

The development of the plasma simulation code tailored for AF-MPD thrusters, especially the SUPREME thruster should be viewed in two or more development cycles/stages. The main goal of the first part of the first cycle shall represent the proof of concept of the four acceleration mechanisms, while the main goal of the second part shall represent the electrode processes (especially losses) with an accurate simulation of the acceleration mechanisms.

2.1. Top-Level Requirements

The top-level requirements (stakeholder requirements) describe how the plasma simulation code shall be structured at the highest level. A crucial step for the code to be used for AF-MPDs is the ability of taking into account not only the acceleration, but also the losses and the erosion and to a high degree of accuracy. Also, the simulation time frame should be short enough that the code has relevance in the development and improvement of the SUPREME thruster, while the cost of a licence shall be feasible for a start-up with limited funding, that has to invest in other aspects of the product cycle as well. CAP refers to the Capability type of requirement (behavioural - reflect functions) and CHAR refers to characteristics (non-behavioural - reflect system attributes):

1. NS-TOP-CAP-1: The simulation time to achieve outputs shall be less than a day.
2. NS-TOP-CAP-2: The software shall simulate the four main acceleration mechanisms (swirl, Hall, gas-dynamic, self-field).
3. NS-TOP-CAP-3: The accuracy of the total thrust results shall be less than 5% from experimental values.
4. NS-TOP-CAP-4: The accuracy of the losses results shall be less than 5% from experimental values.
5. NS-TOP-CAP-5: The accuracy of the erosion results shall be less than 5% from experimental values.
6. NS-TOP-CAP-6: The software shall take into consideration the influence of the anode fall voltage.
7. NS-TOP-CHAR-7: The interface of the software shall be designed to be utilised by non-professional people.
8. NS-TOP-CHAR-8: Design-to-Cost shall be implemented throughout the life-cycle of the software.
9. NS-TOP-CHAR-9: The software shall be designed for applied magnetic field larger than 0.1 T.
10. NS-TOP-CAP-10: The software shall include the magnetic Reynolds number ranges in the thruster.
11. NS-TOP-CAP-11: The software shall have diagnostic procedures.

Here, experimental values refer to experiments conducted on other AF-MPDTs, for which the NSS has the results.

Throughout the report, each requirement received identifiers. The first part, NS, came from the company that requested the thesis work (NSS) with regards to the plasma simulation code and from the name of the thruster, SUPREME (*NS*). The second identifier came from the type of requirement. Above, the top-level requirements were mentioned, thus, the name added was *TOP*. The third identifier came from the type of requirement and the fourth one came from the number. There are also constraints that set the boundaries between the software package and the external environment:

1. NS-SYST-CONSTR-1: The cost of a complete licence shall be up to [TBD] euros.
2. NS-SYST-CONSTR-2: The software licence shall be developed in a partner country with the country the funding is coming from, or the software shall be open-source.
3. NS-SYST-CONSTR-3: The company developing the software shall not store input and output data from the simulations.
4. NS-SYST-CONSTR-4: The software shall be able to integrate within the Digital Twin framework.

Here, the identifier *CONSTR* came from the name constraint, while *SYST* came from system. This also shows these were system level constraints.

While there are stakeholder requirements concerning the accuracy of the outputs, they show the desired crucial values of the simulation. However, an in depth analysis of the software may result in NSS still going ahead with purchasing the license, although the accuracy is still worse than the values of 5%.

As mentioned earlier, for a start-up, time-to-market is an essential notion and it is more important than cost-to-market, especially when the space field is in accelerated interest from the public. So, requirement NS-TOP-CAP-1 ensures that the time from the start of the simulation with the selected operation parameters until the output of results is smaller than a day. This way, there should be enough time for the NSS team to analyse the results, iterate improvements and start a new simulation, to achieve faster product development times.

Requirement NS-TOP-CAP-2 refers to the four acceleration mechanisms widely accepted by researchers and they should all be included in the simulation to understand how the changes in geometry and operation parameters change the thrust models.

Looking at requirements NS-TOP-CAP-4 and NS-TOP-CAP-5, these come as an addition to NS-TOP-CAP-3. A paid license for an advanced plasma code shall not take into consideration only the four acceleration mechanisms like most (open-source) codes for MPD thrusters, but shall also tackle the very important concepts of losses and erosion for an accurate lifetime analysis.

Losses and erosion can have many sources, thus, a new top level requirement (NS-TOP-CAP-6) was designed. This would ensure that the code would take into consideration the important losses caused by anode power deposition ([20]).

Top requirement NS-TOP-CHAR-7 refers to the way the software is designed, with the Graphical User Interface (GUI). This should allow employees or business users without prior coding knowledge to obtain outputs in a timely manner.

Requirement NS-TOP-CHAR-8 ensures that the cost is an active factor in the design process, providing a criterion in the competing software selection, more certainty in the project and a supportable design for the code. In addition, it would provide cost traceability that would help in the design of the interface of the plasma simulation subsystem with the other subsystems.

While requirement NS-TOP-CHAR-9 might seem rudimentary, the reader shall remember what was discussed in the Literature Study [23] and Section 2: most of the AF-MPD thrusters from literature were using mostly low applied B-field and the operation mode was usually high regulated discharge current. SUPREME brings a paradigm shift in the operation of AF-MPDTs.

An important parameter in the thruster in the magnetic Reynolds number, which should be computed at different places in the thruster and compared with the topologies of the electric field and the discharge current. This is emphasised by requirement NS-TOP-CAP-10.

Requirement NS-TOP-CAP-11 focuses on a very important aspect of making sure the code is properly analysed, as well as its outputs. This would ensure checking the code for reliability in results and for understanding different patterns.

In addition to the constrained regarding the thrust accuracy, there are other constraints that are essential in the case of a very young start-up in a sensitive field (space technology). As the main investments are likely to come mainly from USA and VC investors from USA, NSS has to have higher ethics when it comes to business and this means that they can engage with products and services that originate from allied-countries.

Thus, given the reduced budget start-ups have at the beginning, it is essential that the costs from one part of the system not to take most of the available budget (NS-SYST-CONSTR-1). Constraint NS-SYST-CONSTR-2 was designed to ensure the ethical sourcing of products and services. Furthermore, constraint NS-SYST-CONSTR-3 was designed to ensure privacy of the development stages of the SUPREME thruster and reduce the chance of private company data to leak. The fourth constrained ensures that the plasma simulation code fits within the broader project of NSS.

To determine whether a plasma simulation software would be defined as the best solution for NSS and whether a license would be purchased, the following acceptance criteria need to be fulfilled:

1. The software will not output acceleration mechanisms results that differ with more than 10% from experiments.

2. The large particles such as neutrals and ions will be simulated using a Particle-in-Cell approach.
3. The software will consider at least a source of losses.
4. The software will take as input any magnetic topology given.

Acceptance criteria act as a threshold beneath which the product/service is denoted as unacceptable. Again, by experimental values it was meant experiments conducted on other AF-MPDTs, for which the NSS has the results.

Taking the values obtained for PIC-MCC (explained later) by Tang et. al. (2012) [38], depending on the applied magnetic field and discharge current, the simulation results may yield values that are 10% - 20% away from the experiments. As such, 10% is taken as a cut-off (acceptance criteria 1). In addition, Sankaran et. al. (2019) [33] used an MHD two-temperature code to yield acceleration values of 2%, although no other processes (such as electrode sheat processes) have been taken into consideration. Thus, the desired value of 5% was chosen as a top requirement (NS-TOP-CAP-3) for the thrust/acceleration mechanisms.

The second acceptance criteria is designed such that the software uses a more accurate simulation method of Particle-in-Cell (PIC) over other methods such as magnetohydrodynamics (MHD) simulation. Acceptance criteria 3 represents the start point of the plasma simulation code for MPD thrusters. One source (i.e. anode fall voltage) would present the impact losses can have and is a gateway for deeper analysis on losses further ahead in the development of NSS. The fourth acceptance criteria ensures that the code can simulate the processes in an AF-MPDT for any magnetic configuration such that tests can be easily conducted to increase thrust and decrease erosion, back-tracking and magnetic field strength behind the thruster.

3
Theoretical Focus For The Software

This section covers some of the most important physical processes in an AF-MPD, with a focus on steady-state thrusters, that have been introduced in the Literature Study report. This came as a deeper theoretical framework was needed for a proper design of requirements and overall software structure. However, a brief introduction into the theory is presented.

3.1. The Working Principles Of An AF-MPDT

One of the most powerful (highest thrust) electric propulsion technologies is the magnetoplasmadynamic (MPD) thruster [26], [19]. They also have a much higher plasma density than ion and Hall thrusters [26]. It was developed in the 1960's in the USA, with the largest contribution coming from Princeton University. Currently, it is being worked on in a number of countries such as Germany, especially Stuttgart University [4] where it has TRL of 4 [18], while Russia has managed to achieve a TRL of 5.

However, due to the heavy electronics required, it has not been space worthy. Recent improvements in other areas of technologies, such as superconductors can provide a fundamental change in their development.

There are two types of MPD thrusters (MPDTs): self-field and applied-field. The former version works at very high discharge currents which induces a magnetic field. This type of thruster requires power in the order of 100 kW. The later thruster uses an applied magnetic field, either by magnets or superconductors, lowering the need for power in the order of 10 kW In addition, it results in higher thrust possible [26].

All MPDTs have two electrodes: a central rod that acts as a cathode, an annular anode at the backplate (the entrance of the propellant) [4].

This is what NSS is trying to achieve: an AF-MPDT called the SUPerconductor based Readiness Enhanced Magnetoplasmadynamic Electric propulsion (SUPREME). In order to decrease the size and mass of required electronics, high-temperature superconductors (HTS) are used to create high magnetic field from 0.1 T to desired values of up to 1-2 T [6]. In addition, the operation parameters change to high discharge voltage and lower discharge currents and the thruster would work in steady-state mode [37].

The thruster prototype SUPREME is built on is the SX3 AF-MPDT developed by University of Stuttgart. It has a 59% efficiency, uses only 60 kW and has an applied magnetic field of 0.4 T. SX3 used Argon as a propellant and the outflowed particles reached a velocity of 50 km/s [6], [15]. Section 3.3 delves into more details about the aspect of liquid and gaseous propellants.

The magnetic field can also guide (mostly just electrons) the electrodes, especially the anode, where most of the losses occur [2]. Using a stronger magnetic field, depending on its shape, the erosion of the thruster can also be reduced. The current of the thruster cannot be increased continuously as after a threshold (varies with the thruster) the onset phenomenon occurs. This is characterised by increased erosion and losses, and decreased efficiency [4]. So, by applying a magnetic field, the discharge current can be reduced, not only removing the chance of the onset occurring, but also reducing the erosion due to high current (even though the erosion at normal functioning is lower than at onset).

An important source of losses is the electrode fall voltage, especially the anode fall voltage (AFV). With higher discharge current and applied field, the particle density near the anode decreases, increasing the AFV and thus, the anode losses. In addition, a larger anode radius also leads to an increase in the AFV. A way of tackling this is by injecting propellant through the anode slit (from the backplate ($\dot{m}_A$ - see Figure 3.1) and by carefully chosen applied magnetic field shape.

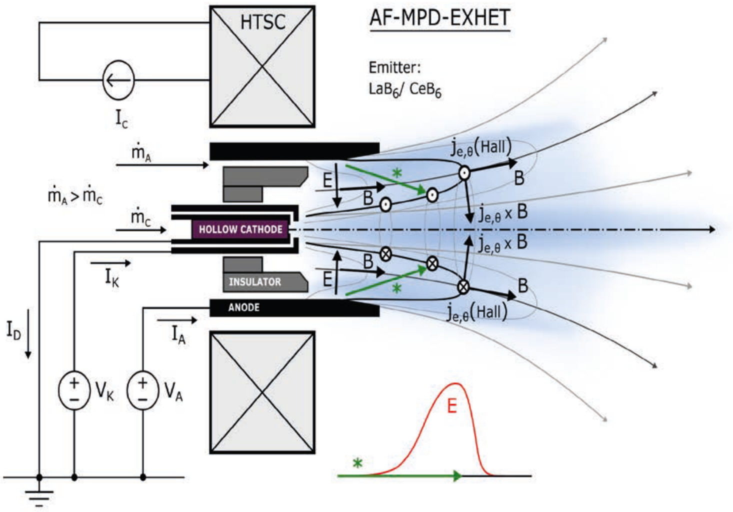


Figure 3.1: A general AF-MPD schematics with vectors shown [6]. E is the electric field; $\dot{m}$ with indices of A and C correspond to the mass flow from anode, respectively, cathode; $j_{e,\theta}$ is the same as j_θ and represents the induced Hall current; B is the applied magnetic field; I and V represent the current and voltage at different sections.

The use of an applied magnetic field also creates two additional forces that increase the thrust of an AF-MPDT: the Hall force (F_H) and the swirl force (F_{sw}). The former is created by the interaction between the induced Hall current and the applied magnetic field ($j_\theta B_z + j_\theta B_r$), while the later is created by the interaction between the discharge current and the applied magnetic field ($-j_r B_z + j_z B_r$).

The other two forces present in both MPDTs are the self-field and the gas-dynamic components. The the self-field force (F_{sf}) is the interaction between the discharge current and the self-induced magnetic field created by the current ($j_r B_\theta + j_z B_\theta$). The gas-dynamic component comes from the thermodynamic expansion of gases [2], [35]. To have a clear understanding of the geometry and vectors involved in AF-MPD thrusters, please see Figure 3.2.

Experimental tests, especially for AF-MPDTs require large, specialised vacuum chambers, facilities, making tests very expensive. For a start-up with limited resources and for which time-to-market is important, the use of simulation codes is necessary. Thus, different codes can simulate different parts of the AF-MPD thruster and work together in a Digital Twin framework to present the image of the thruster after a period of functioning. The plasma simulation inside SUPREME was the focus of this report. Using simulations can reduce and decrease the experimental costs and time, it can lead to faster optimisation and new physics models can be tested faster.

Over the years there were a number of developments on the plasma simulation side of AF-MPDTs. For example, the USA Air Force has developed a magnetohydrodynamics (MHD) simulation code: the Multi-block Arbitrary Coordinate Hydromagnetic (MACH). The most recent recent update is MACH2 in 2.5 dimensions, with the focus being on MACH3 (3 dimensions) [19], [1], [10]. MHD codes simu-

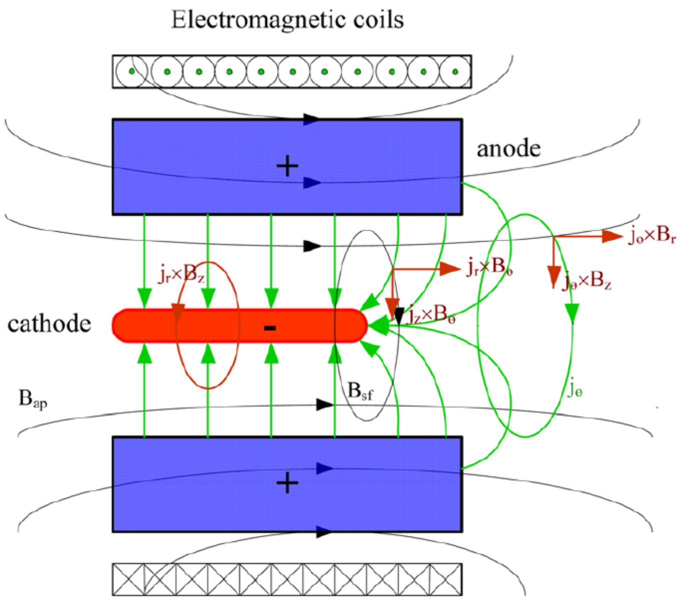


Figure 3.2: Schematics of an AF-MPD thruster with the self-field, swirl and Hall acceleration thrust vectors drawn [38]. Here $j_r \wedge B_z$ (first circle from the left of the image) is one of the components of the swirl force (F_{sw}), but this component is negative when computing the entire swirl force, $j_r \wedge B_\theta$ and $j_z \wedge B_\theta$ (second cirlce) are the components of the self-field force (F_{sf}) and $j_\theta \wedge B_z$ with $j_\theta \wedge B_r$ (third circle from the left) are the components of the Hall force (F_H).

late particles as fluids, usually two fluids (two temperature): heavy particles (ions) and light particles (electrons). Another example is the two-temperature code developed by Sankaran et. al. (2019) [33] which yielded results that were within 2% from the thrust experimental data. However, the code has not accounted for phenomenons inside the thrust, such as the electrode fall voltage. The high accuracy should also be taken with a pinch of salt. The code is applied at the beginning of the test and for a short period of time, as losses and erosion do not have a significant role in a very short time frame.

Another approach is the use of kinetic codes instead of MHD. ESA is also interested in this research avenue and is collaborating with one of the options under investigation from this report. In addition to the modular nature of new codes which would be suited for multiple electric propulsion technologies, the newer codes tend to move towards kinetic codes. This new approach is more suitable as electric thrusters deal with low Knudsen numbers [7]. The kinetic approach splits into two approaches: fully kinetic (all particles are treated individually) or hybrid (electrons are treated as a fluid - MHD and the heavier particles such as neutrals and ions are treated kinetically). This kinetic approach is called the Particle-in-Cell method, to which almost always applies the Monte Carlo (MC) method to simulate particle collisions. By using PIC the time-dependent full Maxwell-Vlasov equations are solved numerically at each time step and in each cell, so particles interact with the fields [13]. While as a general rule of thumb, PIC simulations tend to be more accurate than MHD ones, they do require more computational resources.

3.2. Electrode Fall Voltage

In the Literature Study it was explained that Myers (1995) and Kurtz (1971) [16], together with Albertoni et. al. (2015) [2] have shown experimentally that with higher discharge current or higher applied field, the pinching effect increases, causing the anode fall voltage to increase linearly. This was also supported by Lev et. al. (2011) [20].

In addition, Lev. et. al. (2011) explained that the anode power deposition is the main power dissipation process, which affects especially the AF-MPD thrusters up to several tens of kW, where the current is not high enough for strong self-magnetic fields. This in turn makes the thruster to have low efficiency because of the anode power deposition [20]. On the other hand, Sankaran et. al. (2003) [33] explained that as the current density is inversely proportional with the radius, in the case of cathode (where radius is lowest), the power deposition and dissipation phenomenons are large in the cathode area (especially

1 cm around the cathode). However, this might also be due to the geometry of the thruster used at Princeton. In addition, this near cathode region is the only place where the kinetic power is larger than the Ohmic heating [33]. According to Myers and Soulas (1992) [30] the anode power deposition in a proportion of 60%-95% is caused by *"electron current conduction into the anode"*, deposition from the cathode radiation accounts for 5%-35%, while up to 5% of the anode power deposition is influenced by the hot plasma through *"convective heat transfer"*. The impact of temperature is explained later. In addition to increasing the anode mass flow rate to reduce the AFV, thus, the anode power deposition, Myers and Soulas (1992) [30] concluded that the shape of the electrodes and the applied magnetic field topology should be changed so that *"parallel current conduction into the anode"* is supported.

Thus, this is an important process to be considered in the plasma simulation code. However, while the SUPREME thruster is intended to be ran at a few kW of power for the first products, the operation mode of voltage regulated would play a major role in reducing the anode fall voltage (AFV). In addition, the code shall consider the impact of the mass flow rate, current and applied B-field with regards to the AFV, while treating the anode sheath as unmagnetised as the Debye length for MPDs ($\sim 10^{-6}$) is much smaller than the electron gyro-radius ($\sim 10^{-4}$) (although the AFV behaviour supports the claim of the magnetised fall region [30]).

The simulation shall also take into consideration the thermionic current (j_{th}; with increasing electrodes temperature, both the anode and the cathode release electrons if the thruster operates in steady-state mode [23]) and can be treated according to Lev et. al. (2011) in the context of net current density from the anode (j_∞) [20]:

$$\begin{aligned} j_\infty &= j_e - j_{th}, \\ j_\infty &= \frac{J}{A_a}, \end{aligned} \tag{3.1}$$

where j_e is the electron current density through the anode seen in Equation 3.2, J is the total discharge current applied and A_a is the total anode area.

$$j_e = \frac{1}{4} e n_e \left(\frac{8 k_B T_e}{\pi m_e} \right)^{1/2} \exp\left(\frac{eV_a}{k_B T_e}\right), \tag{3.2}$$

where n_e, T_e, m_e are the electron density, temperature and mass and V_a is the anode fall voltage.

As such, the software shall also model the pinching effect due to the applied magnetic field and discharge current and how it affects the AFV and the power deposition. According to Lev et. al. (2011) [20] the density decreases exponentially with r^2, while Li et. al. (2019) showed that the density varies depending on the location in the thruster. A mapping of the density of species may yield important advancements in the theory of MPD thrusters. Lev et. al. (2011) [20] also shown through theoretical concepts and experimentally that the anode temperature is mostly dependent linearly on discharge current, while the applied magnetic field and the mass flow rate also have a small influence in the fitting of the data points of the anode temperature to the model. These three operational parameters (J, B_a and $\dot{m}$) are also the three parameters used to model the anode fall voltage.

Another approach would be to use the anode power balance equation. However, this is not a feasible approach since it requires the cathode temperature and there is not enough theoretical development to link it to these three operation parameters. Thus, it might be better not to be implemented in the code as it might slow the development [20]. Lev et. al. (2011) [20] states that at high current values the anode overheats, which causes an increase in the thermionic current, resulting in an increase in the electron current density, thus, increase in the anode fall voltage (see Equation 3.2). In addition, it was theorised that higher mass flow rates result in an increase in the centrifugal forces in the thruster, increasing the particles density throughout the thruster.

While it was shown that current has a linear relationship with the AFV, there are three effects happening that define this relationship. Firstly, the higher total thruster current (J) would cause an increase in the electron current density, which causes an increase in the AFV according to Equation 3.2. Secondly, the increase in current would result in higher anode temperature, thus, higher thermionic current and in turn higher electron current density (and AFV) to account for the emitted electron flux. Lastly, higher current would result in higher plasma pinching (applied B-field would have a similar pinching effect), decreasing the density at the anode and increasing it at the thruster centre. This in turn would decrease the random electron flux of the anode. To maintain Equation 3.1 (current density balance) at a lower density, a higher sheath voltage is required. The pinching effect by the applied B-field explains the approximately linear relationship between the field and the AFV, while the sheath electrons are unmagnetised as mentioned before. On the other hand, the higher mass flow rate would cause a higher random electron current (higher density), thus, the AFV decreases to balance the current density balance equation [20]. The mass balance has the following relationship with AFV: $V_a \sim \ln{(1/\dot{m})}$.

Lev et. al. (2011) [20] concludes by stating that in quasi-steady MPDTs the pulse length is too short for heating the anode, thus, thermionic emission can be neglected in quasi-steady devices. It can also be understood that when designing an AF-MPDT, an anode material with a higher work function is desired for a reduced thermionic current.

While Li et. al. (2019) [21] recently showed through simulations that the current density to be almost constant for the cathode during the operation (quasi-steady AF-MPDT) at about 150 A/cm^2, Sankaran et. al. (2003) [33] simulated values ranging between 200 A/cm^2 to 550 A/cm^2 for a quasi-steady AF-MPDT for a cathode longer than the discharge chamber. Boyle [9] measured for a shorter cathode, values of even 1000 A/cm^2 at the cathode tip. Sankaran et. al. (2003) [33] suggested the differences can come from the difference in cathode length and the shape of the cathode tip: hemispherical tip corresponds to lower current density as opposed to conical tip. Reducing the electrode current density should result in lower erosion levels and a decrease in the Ohmic dissipation [33].

3.3. Propellant Use For An AF-MPD

There is a range of propellants that can be used by the SUPREME thruster and the software shall take them into consideration with focus given depending on the priority: Argon shall be simulated first, as this is a cheap gas, while being a noble gas that is easy to handle and its plumes do not interfere with the rest of the spacecraft. Other choices are Hydrogen with high specific impulse (I_{sp}) and large Alfvén velocity (v_A). Lithium (6.94 g/mol [40]) on the other hand has a very low first ionisation potential (5.4 eV), compared with Argon (15.7 eV, 39.95 g/mol [40]) and Hydrogen (13.6 eV, 2.02 g/mol [40]) and a much larger second ionisation potential than these propellants. Having a many more singly ionised particles reduces the frozen flow losses, while lower electrode losses expected for lower first ionisation energy. In addition, Lithium with addition of Barium results in the cathode having lower work function. Thus, it can run at lower temperatures, reducing the erosion [20]. However, their plumes have deposition issues on the electrodes and spacecraft (S/C) surfaces, which is not suitable for a thruster used by space stations, scientific missions.

Furthermore, the type of propellant would play a role from the start as they would influence the start-up time of the thruster. In addition, lighter ion atoms have a higher mobility and would result in faster start-up [17]. Also, higher atomic mass would result in lower thermal velocity, thus, lower gas-dynamic thrust. Thus, given the many factors involved in the choice of propellant, additional elements can be considered: Ammonia (10.07 eV, 17.03 g/mol [40]), Hydrazine (8.36 eV, 32.05 g/mol [40]), Ammonium Dinitramide (124.02 g/mol [27]) and mixtures of different propellants. These are unusual for electric propulsion, but have been widely used in chemical propulsion. Due to the permissive nature of MPD thrusters they can be considered as viable options depending on the specifications of a mission.

In addition, in the recent years H_2O (water) is being used increasingly more for space propulsion, especially for small satellites. This propellant is however, not used in the SUPREME thruster as water leads to the production of atomic oxygen, which can erode the electrodes and decrease their lifetime. This is a similar argument to any other possible propellant that causes the production of atomic oxygen. In more extreme cases (longer use for example), the thrust accuracy and direction can also be affected.

As concluded in the Literature thesis [23], lightweight propellants are preferred as this leads to less complex storage and feed system, reducing the total mass of the thruster [18].

As such, for the initial prototypes, Argon was chosen as the propellant for SUPREME. While all of the three plasma simulation options analysed in this report have the capacity of implementing the use of Argon, only one option has the ability of simulating a limited number of more complex propellants. The analysis conducted in this subsection emphasised the propellant of choice for the first prototype and the future requirement for the next iterations of the code.

3.4. Sputtering

Sputtering phenomena is also common on other propulsion technologies such as the ion thruster and it refers to the erosion of material by ions, which later deposits onto other materials in the vicinity, as a thin film layer [28]. According to Nakles et. al. (2003) [31] the sputtering in ion thrusters is caused by low energy ions and the tests conducted showed that at 70 eV there are more sputtered layers than at 100 eV. However, Munoz (2005) [28] stated that higher energy ions can cause a larger wave of disturbance in the target ions. The tests conducted with high energies (500 eV - 1500 eV) where Argon was used as propellant have shown that on a tantalum target, the peak sputtering occurs at 30° and it is the same for all energies. On a titanium target, the peak sputtering yield was at an incidence angle of 60° and at 1000 eV, while on a tungsten target the peak yield was at 45° and at the value of 500 eV [28]. Given the average peak sputter yield of about 45° in the experiments, it should come natural that the desired shape of the magnetic nozzle in the case of SUPREME would be either parallel with the walls or converging, while the diverging part would be outside the discharge chamber in order to reduce the sputtering phenomenon. This should ensure low incident angles, thus, low sputter yield.

However, there are multiple ways of tackling the erosion caused by this and it is natural to assume that the methods would work for both ion and AF-MPD thrusters. Firstly, the use of lower sputter yield materials in the most affected regions or the regions prone to sputtering. Secondly, the specialised treatment of the deposition regions. In the case of AF-MPDs, they are mostly on the cathode. Thirdly, the appropriate operating parameters to reduce sputtering. In addition, it was observed that the geometry of the thruster has a strong impact on the sputtering in an ion thrusters, thus, it is logical to assume it will impact this phenomenon in AF-MPDTs as well [24].

Since sputtering is caused mostly by secondly ionised ions, this is further reason for the propellant used to have a low first ionisation energy level and much higher multiple ionisation levels [31]. Also, thruster/target materials with a larger atom size than the size of the propellant atoms can lead to a significant reduction in sputtering [28].

There are two notions that have to be defined: total yield represents the total number of target atoms sputtered per ion , while differential yield represents the total number of target ions sputtered per ion at a certain angle/direction [28].

3.5. Operational Characteristics

The increase of mass flow rate has been previously explained as a step to reduce the AFV, however, decreasing the mass flow would increase the I_{sp}, which can also lead to intensive cathode erosion. This is due to the fact that with higher current, there is higher specific impulse and without a larger mass flow to counteract this, extensive cathode erosion applies [12].

Analysing the pulsed, quasi-steady and steady-state operation, LaPointe et. al. (2001) [19] explained that pulsed MPD thrusters have worse performance (lower thrust) than steady-state thrusters. However, they do not suffer from some of the instabilities present in steady-state and can be used as a faster alternative for testing an MPD thruster. In addition, it was explained that quasi-steady thrusters have are likely to have a slightly worse performance than steady-state ones. Nevertheless, with increasing discharge powers, these differences become smaller [19]. In addition, the steady-state operation can have a much lower cathode erosion than the other operating modes [36].

3.5.1. Experimental Measurements For Code Testing

In order to test the accuracy and the detailed output of the plasma simulation software, the SUPREME thruster shall undergo experimental testing. Myers [29] and Tang et. al. (2012) [38] showed experimentally, and the later also computationally, that these main characteristics increase the thrust: linearly with an increase in discharge current and applied magnetic field, parabolically with an increase the anode radius (AFV and thruster efficiency also increase with anode radius) and mass flow rate, with a decrease in the cathode radius and with increasing electrodes length. As such, careful experimental tests have to be conducted that are in line with the plasma simulation and attention should be given to a voltage regulated thruster.

An important aspect of the simulation, the AFV can be tested in theory using an emissive Langmuir probe that measures the floating potential of the anode. However, this has not been conducted yet and such a process might yield a significant amount of new information that can be used in later iterations of the plasma simulation software. As there are no extensive tests for this, it was not considered in the analysis of the simulation options. However, after a choice is made and a plasma simulation software for SUPREME is developed, the results of the experiments with the emissive Langmuir probe would be integrated in the improvement of the code. In addition, the anode temperature can be measured through an optical access in the vacuum chamber using an optical pyrometer [20].

To test sputtering, an analytical balance can be used to weigh the pieces before and after operation. In addition, the depth of the sputtering can be measured by a profileometer, after covering the areas with a mask, usually made of iron and tantalum. To test for deposition rates, masked quartz slides were put in different regions and their composition and thickness were quantified by the use of the aforementioned profileometer and a spectrographic analysis. Besides geometry and erosion zone mask, the effect of high discharge voltage should be analysed in the sputtering phenomenon [24].

A new cathode technology that can be used in high power systems such as the MPDTs is the lanthanum hexaboride (LaB_6) hallow cathode. In the startup phase it would need less than 440 W and about 150 V for ignition. After reaching keeper discharge, the heater can be turned off.

An important aspect that has to be considered for LaB_6 hallow cathodes is the very limited testing conducted (at Princeton University and Nagoya University) at magnetic field strengths of more than 100 mT. This range is much smaller than planned magnetic fields for AF-MPDTs that might reach even 1-2 T. In addition, it has been tested at regultated voltages of up to 450 V with 40 A discharge current, while in continuous operation it has been tested mostly in the range of 100-225 V. [8].

3.5.2. Plume ID

The optical pyrometer can also be seen as a tool used to identify the different plumes from electric space propulsion technologies by measuring the plume brightness temperature after being gray-body corrected following Equation 3.3:

$$T_g = \alpha[\ln(1 + \epsilon\,(e^{\alpha} - 1)], \tag{3.3}$$

where ϵ is the gray-body emissivity (that of the filament of the probe) and $\alpha = {}^{hc}/_{\lambda k_B}$. Here, c is the speed of light in vacuum, h is the Planck's constant, λ is the wavelength of the emitted radiation and k_B is the Boltzmann's constant.

Another way of identifying a specific thruster came from the Sankaran et. al. (2003) [33] research that showed that the Princeton benchmark thruster presented a luminous barrel known as "cathode jet", from which electron density can be inferred based on the light emission from the discharge.

Aspects that can be used together for plume ID and thus have the possibility of identifying the thruster components, operational modes and who developed the MPDT technology are: the acceleration corresponding to the light emitted at a certain wavelength, exit diameter, divergence of the plume, rate of density fall off, intensity regions at the thruster exit, rate of radiation emission.

3.6. Parameter Influence In The Code

The plasma simulation software has to not only represent correctly the processes occurring in an AF-MPD thruster, but to also use a certain set of inputs and provide a certain set of outputs. These outputs can be ran in other simulations down the line, in order to help improve the technological development of AF-MPDTs, but also its integration within satellites, space stations. One such needed output is the electron Hall parameter, which gives a relationship between the self-induced Hall current (perpendicular to the electric field) and the discharge radial current (parallel to the electric field): $j_\theta = \beta_{e,Hall} \cdot j_r$.

Furthermore, thermionic current should be given as an output, which plays a role in the radial current density and as was explained in Section 3.2, it has an important effect in steady-state MPDTs. Having the induced current field as output as well, would thus, allow understanding the radial current map in the thruster. If possible to output, the average velocity of each species in the three thrust regions (discharge, confinement, divergence) would allow to compare the velocities profile of an AF-MPD with that of a chemical propulsion thruster: sub-sonic, sonic and super-sonic.

While Tobari and Ando [20] have measured experimentally that the radial electric field is linear with distance ($E_r = E_0 \cdot r$), Li et. al. (2019) [21] showed experimentally and through code modelling that the electric field is stronger closer to the electrodes and it is shielded in the bulk plasma region. Thus, another output shall be the AFV which plays a role in the radial electric field, while the total discharge voltage is a parameter controlled by NSS and shall be an input. In addition, the density of different species as an output would provide information to better understand erosion and the design of the thruster.

Considering the sputtering phenomenon which is caused especially by multiple ionised ions, the plasma simulation software shall also take as input the energy sputtering yields and the sputtering threshold [31].

In addition, according to Sankaran et. al. (2003) [33] and Tang et. al. (2012) [39], the backplate (insulator/dielectric area) can be treated as the first boundary, the inner walls (electrode walls/conductor walls) can be treated as a second boundary, while the third one can be the free boundary. Furthermore, depending on the size of the simulation domain (AF-MPDT plumes travel a large distance) the third boundary can be set at a certain distance from the thruster.

Although self-field thrust component is much smaller than the other mechanisms, there has to be an update of the formula given by Maecker ($F_{sf} = \frac{\mu_0}{4\pi}\left(\frac{3}{4} + \ln\frac{r_a}{r_c}\right) I^2$ [23]). According to Choueiri [38], there are two regimes: the fully ionised regime which has the same formula (mostly the case for SUPREME as the electrons have high energy) and the partially ionised regime:

$$T_{sf} = \frac{\dot{m} v_a I}{I_{fi}}, \tag{3.4}$$

where v_a is the Alfvén Critical Ionization Velocity (CIV) (explained in the Literature Study [23]), $\dot{m}$ is the mass flow rate, I is the discharge current and I_{fi} is the fully ionised plasma current:

$$I_{fi} = \sqrt{\frac{4\pi \dot{m} v_a}{\mu_0}\left[\ln\left(\frac{r_a}{r_c}\right) + \frac{3}{4}\right]^{-1}}, \tag{3.5}$$

where r_a and r_c are the anode and cathode radii.

In addition, Tang et. al. (2012) [38] showed through simulations that for a high discharge current operation, changing the applied B-field does not influence the Hall thrust, while increasing the current slightly increases this thrust component. This has to be checked with the simulation code as the Hall thrust is expected to be the major thrust component for high applied magnetic field operation. Also, other research groups have shown the Hall component to be the largest thrust, as it added two new components that depend on the high applied magnetic field. The ratio of forces given in the study of Tang for self-field, Hall, gas-dynamic and swirl mechanisms was: 1:10:10:100 and this shall be tested against the new plasma simulation code from a chosen stakeholder.

3.7. Additional Code Aspects

For a better understanding of the code, the two most important aspects of using a particle-in-cell Monte Carlo collisions (PIC-MCC) method over an MHD code need to be restated: the nonlocal dynamic effects of all the particles can be taken into consideration and transport coefficients of the plasma flow can be obtained easier. Given that the interactions between the electromagnetic fields and the particles resulted from discharge current and voltage are similar to some extent for ion, Hall and MPD thrusters, it can be a natural assumption that experience and software developed for the first two can be used for the latter.

The need for statistical methods is also supported by the rarefied flows in the thruster and the high (larger than one) Knudsen numbers obtained (the ratio between mean free path and characteristic length, which in this case can be the length of the thruster - axial direction). The PIC-MCC approaches split into two: hybrid (computes the density of electrons from the Boltzmann relation), which is easier to develop, eliminates incorrectly small scales for the electrons, keeps plasma quasineutral and is better suited to deal with the 2D magnetic and geometric (this would be less important for 2D in space, 3D velocities or fully 3D) and fully kinetic, which would not approximate $\nabla \wedge \vec{B} = \mu_0 \vec{J}$ to be zero and overall more accurate results (such as thrust, erosion etc.), especially for the physical processes. A fully kinetic model however, would also require more computational power as the simulation time would be given by the electron time-scale. This would result in a million of time-steps, which is the time needed for an average neutral or ion to pass the simulation range. Furthermore, the use of a hybrid PIC, while it would decrease the computational time, it can be applied if it obeys three conditions [34]:

- electrons are treated as massless particles: no inertia,
- the motion of the electrons must not be affected by strong fields (such as the applied magnetic field in AF-MPDTs),
- the velocity of the electrons can be sampled from a Maxwellian distribution, as there is thermodynamic equilibrium for the electrons within the plasma.

As can be seen, the second condition cannot be respected in the operation scenario of SUPREME.

There are multiple ways to tackle this large number of steps and the large computational power that would be needed. First, the method mentioned in the Literature Study ([23]) is the use of super-particles: bunching particles together. Secondly, the heavy particles can be treated as having a lower mass to *"accelerate convergence"*. Lastly, increasing the Debye length by increasing ϵ_0, the vacuum permittivity [39].

3.7.1. Mean Free Path And Cross-Sections

The mean free path is a very important parameter in plasma simulations and can be computed based on theoretical calculations and if available, with experimental data as well. However, a more accurate way, but more resource consuming, might be computing the mean free path for each collision between particle 1 and a background particle 2, according to:

$$\lambda_{12} = \frac{1}{Q_{12} \cdot n_2}, \tag{3.6}$$

where Q_{12} is the cross section for the collision and n_2 is the density of the background particle 2 [39]. The cross-section is not a fixed value either. Given the energy of the colliding particles, it changes and should be computed for each case. However, given that the energy of the particles might be in the eV range, it can be approximated as a fixed value.

Tang et. al. (2012) [39] provided a table (first two processes are the most important for the simulation) with interactions and their respective mean free paths which can be seen in Table 3.1, for assumed densities: $n_n = 4 \cdot 10^{15}\,\mathrm{m}^{-3}$, $n_i = n_e = 3 \cdot 10^{15}\,\mathrm{m}^{-3}$ and mean temperatures: $T_e = 0.15$ eV, $T_i = 0.2$ eV. Here, the indices n, i, e refer to neutral, ion and electron.

Sankaran et. al. (2003) [33] showed that the theoretical current contours follow the experimental measurements only at higher current values and closer to the backplate. This could have come from larger

Collision Process	Mean Free Path (m)
Electron - neutral ionisation	30
Ion - neutral exchange	20
Electron - ion Coulomb	0.02
Electron - electron Coulomb	0.009
Ion - ion Coulomb	2
Neutral - neutral scattering	70

Table 3.1: Interaction processes and their mean free path [39].

magnetic Reynolds number values in the experiments or the continuum assumption starting to break down in low collisionality regions. The latter represents an important aspect in the plasma simulation code and experiments should be conducted to improve the accuracy of the code in this area as well. Their research also showed that in the higher current density regions mentioned earlier, there is a higher effective ionisation level. This comes naturally as in higher current regions, it is more likely for ions to be multiple ionised.

In addition, the interactions between particles and the closed boundaries (inner surfaces) have to be considered. The first interaction, erosion, is caused by ions hitting the material of the thruster and was discussed as part of the sputtering section (see Section 3.4). The second interaction is secondary electron emission (SEE) which occurs when an electron or an ion hits a surface and causes at least one electron to be ejected into the plasma (created in the code). The third type is electron absorption in which an electron hits the surface and deposits its charge (it is deleted from the code/simulation). The fourth type is the ion neutralisation: ions hit the surface, get an electron (deposit charge) and are scattered as neutrals. Then, the fifth type are scatterings, which are also split into three types: open (the inflow/propellant injection and outflow/particles accelerated out of the thruster), adiabatic elastic scattering (particles change their normal velocity component by 180° and isothermal scattering (in which the velocity components change) [34].

3.7.2. Ionisation Processes

Plasma in MPDs tends to be in a state of ionisational nonequilibrium. As such finite-rate ionisation models are not suitable to take into consideration higher levels of ionisation. To consider the cases of singly and multiple ionised ions, the chosen plasma simulation software shall use a multilevel equilibrium ionisation model [33].

Using the fluid theory cannot take into consideration ionisation, which according to experiments occurs almost completely ($\sim 95\%$) in the propellant a few millimetres from the inlet. The solution used was a high enough temperature that would provide the conditions for almost full ionisation near the backplate (at the mass flow entrance) [33]. According to Tang et. al. (2012) [39] the only ionising collisions are electron - ground state neutral, as mentioned earlier. This results in two electrons and one ion at the end of the time-step.

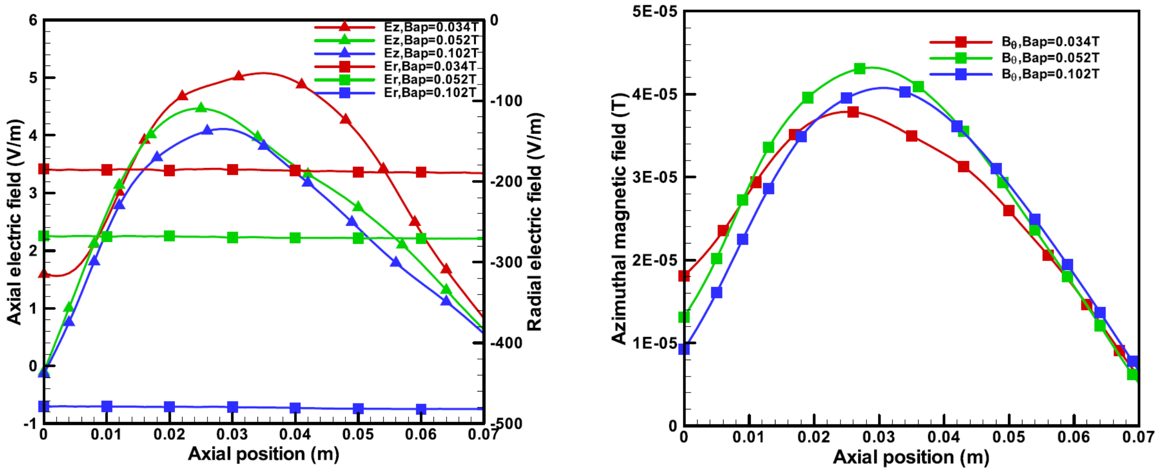
The procedure has to follow MC rules. As such, first the programme checks if an electron has a kinetic energy larger than the first ionisation energy of the neutral. If it has, then the probability of collision (taking into consideration the cross-section) would be compared with a random number between 0 and 1 to check if the ionisation event occurs. If it does, then the primary electron is scattered stochastically and the energy of the two electrons (primary and secondary) follows a probability distribution.

3.7.3. Fields And Thrust Mechanisms

Tang et. al. (2012) [39] also showed that while the radial electric field is almost perfectly constant along the thruster's length, the axial electric field was much smaller (an order of magnitude). The reader should not confuse this with what was determined by Li et. al. (2019) [21] with regards to variations along the radius, not along the symmetry axis. The axial electric field magnitude started increasing at the discharge chamber entrance, reaching a peak in the middle of the thruster, after which it drops back down to similar values at the chamber exit (see Figure 3.3a). The software showed this for multiple magnetic field strengths and the conclusion reached was that the particle motion does not have an important influence on the radial electric field.

Looking at the azimuthal (self-induced) magnetic field, which in their case was three orders of magnitude smaller than the applied field, the same shape as for the axial electric field was observed, from chamber entrance up to chamber exit (Figure 3.3b). In addition, the conclusion reached was that the azimuthal B-field peak was influenced by plasma processes and was not affected by the magnitude of the applied B-field [39]. However, attention must be given on whether their simulation model took into consideration all the processes involved. Thorough experimental tests are needed to exclude the interaction between self-induced and applied magnetic fields.

In addition, Tang et. al. (2012) showed that using a fully kinetic PIC-MCC simulation (2D axisymmetric volume, plasma components are 2D in space, velocity components are 3D), the thrust results achieved were closer to the experimental measurements than the MACH2 (based on the MHD approach) code used by NASA (see Figure 3.4). Both experimental data and simulations showed that in this case the swirl acceleration was the main thrust component, however, the applied magnetic field was very low (up to 0.12 T) compared with the values desired for the SUPREME thruster and the SUPREME thruster would operate at regulated voltage. As such, for SUPREME the Hall acceleration is expected to be the dominant thrust mechanism. However, experiments have to test this, as Tang et. al. (2012) [38] stated that the applied B-field did not influence the Hall acceleration created mainly through the electron swirl.



(a) Axial and radial components of the electric field as they vary with axial position [39].

(b) Azimuthal (self-induced) magnetic field as it changes with axial position [39].

Figure 3.3: Electric field and self-induced magnetic field at different applied B-field strengths, using Argon propellant and a discharge current of 1 kA.

Regarding the density in the simulation, the density of the neutrals was given as an input and was treated uniformly (azimuthally) in the Tang et. al. (2012) [39] simulation, according to the following formula:

$$n_n(z) = n_{n,0}(1 - {}^{z}/_{L})^2, \tag{3.7}$$

where $n_{n,0}$ is the initial (at inlet) neutral density, L is the discharge chamber length and z is the axial position. This is in line with the 2D axisymmetric approximation (as the formula looks only at the axial direction). From this, the mass flow rate was assumed to be changing only along the discharge chamber's length (z).

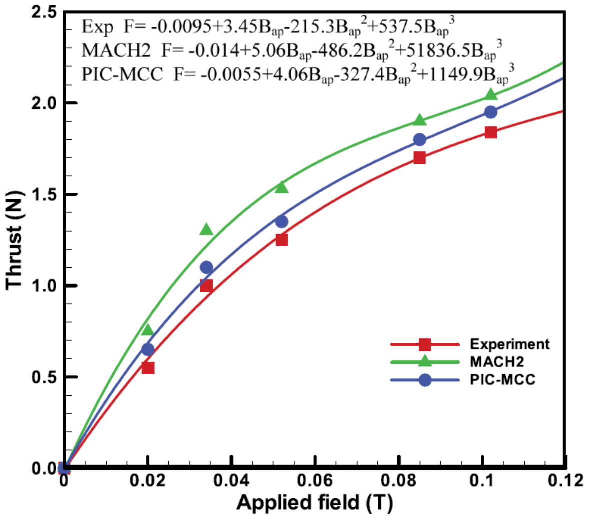


Figure 3.4: Comparison between MACH2 and PIC-MCC codes for applied magnetic field range of [0-0.12] T, with Argon propellant and discharge current of 1 kA [39].

3.7.4. Temperatures And Velocities

Through simulations it was observed that a higher applied magnetic field causes the ions to exit faster the discharge chamber, thus, the density of ions decreases not only downstream (towards exit), but with higher applied B-field as well. As such, higher applied magnetic field, discharge current and mass flow rate were shown to increase the ion and electron velocity and momentum in the simulations. Looking at the current distribution on the three components, it was observed that the radial component has by far (around 99%) of the total current, which was also supported by the high radial electric field. Increasing the applied B-field can slightly increase the percentage of radial current. Furthermore, lower applied magnetic fields cause higher currents at the extremities (entrance and exit of the discharge chamber), while higher applied B-fields cause stronger currents along the electrodes and more uniform overall [39]. It should also be stated that Tang et. al. (2012) [38] showed that with increasing applied magnetic field, discharge current and mass flow rate, the maximum of the charged particles' velocities slightly shifted in the upstream direction, closer to the backplate.

There were a range of values for the electron temperature (higher velocity indicates higher temperature) which may depend mostly on the thruster geometry. Serianni and Myerst [2] showed that near the anode the electron temperature is about 2 eV, and 5 eV in the bulk plasma, while Sankaran et. al. (2003) calculated the values near anode to be between 0.75 eV and 3.5 eV and Diamant measured values of about 2.5 eV near the anode [33]. Tang et. al. (2012) [39] explained that using the classical temperature formula ($T = {}^{mv^2}/_{k_B}$, where m is the mass of a particle, v is its velocity and k_B is the Boltzmann's constant) and other characteristics of the particles from the simulation, the temperature of the electrons and ions can be found (see Figure 3.5). The plots show that the electrons have an average temperature in the order of 1-2 eV, similar with other research papers, increasing with increasing applied B-field, while ions have a very low temperature close to the backplate and with a spike towards the exit, ranging from 0 eV to 12 eV, and with no obvious relationship with the applied magnetic field. The explanations come from the acceleration nature of charged particles. Electrons are accelerated electromagnetically and have most of the input power in the upstream region, where their temperatures peak. Through elastic and inelastic collisions along the chamber channel, they transfer the energy to the ions. After obtaining energy through collisions, ions are then accelerated electromagnetically [39].

In addition to the relationship between thrust and applied B-field, discharge current, mass flow rate, electrode radii, Tang et. al. (2012) [38] also showed how the velocity of charged particles varies with applied field, discharge current and mass flow rate. While for higher discharge current the axial, radial and azimuthal velocities for both ions and electrons increased almost linearly, the relationship is not so simple for the other two parameters - applied magnetic field and mass flow rate (see Figures 3.6

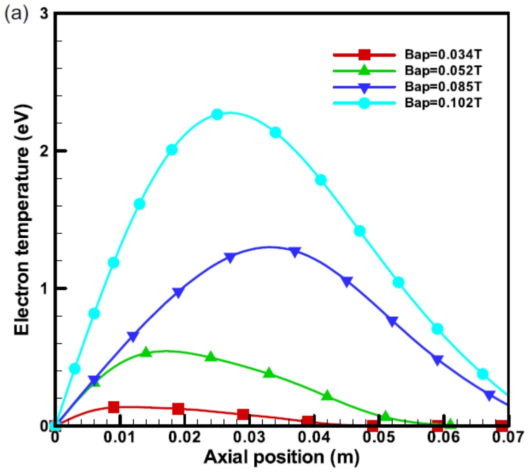
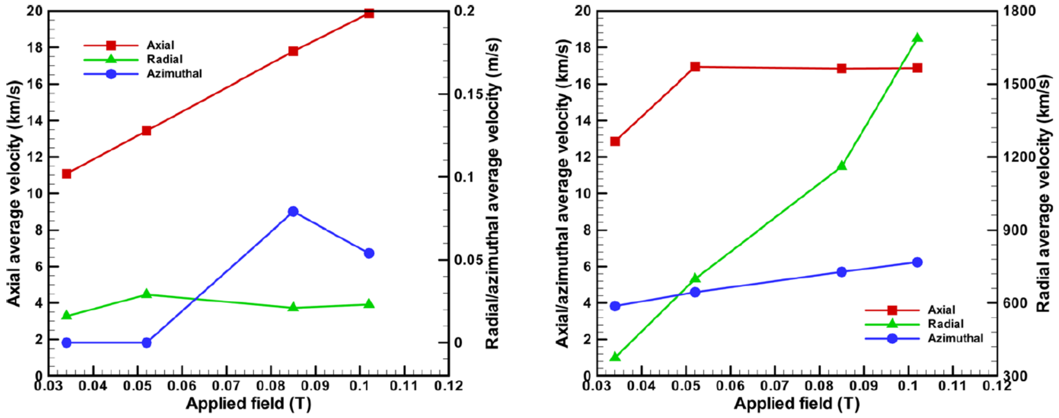


Figure 3.5: The electron (left) and ion (right) temperatures through the along the discharge chamber axis, at different applied magnetic field strengths [39].

and 3.7).

In the case of higher applied magnetic field, the velocities used to increase as well, although not necessarily linearly. For the case of higher mass flow rates, while some velocity components increased, the axial velocity for the electrons saw a decrease. In the case of ions the most important component is the axial one, while for electrons it is the radial component as it plays an important role in current conduction in plasma. It was also noticed that mean ion and electron velocities reach values higher than 10^4 m/s, respectively, $3{\cdot}10^5$ m/s [38]. This also showed that the classical equations still hold, as relativistic effects start becoming noticeable from 10^6 m/s onward. Analysing mean velocity values of the components it was concluded that the highest current density component is j_r, due to the radial electric field. This means that the dominant force is $-j_rB_z$ and *"electrons carry most of the current"*. Moreover, it was shown that with increasing applied magnetic field, both the azimuthal and axial current components decrease [38]. This would need more research as with increasing applied B-field, the Hall thrust mechanism should get stronger, which comes from a stronger applied field and induced Hall current.

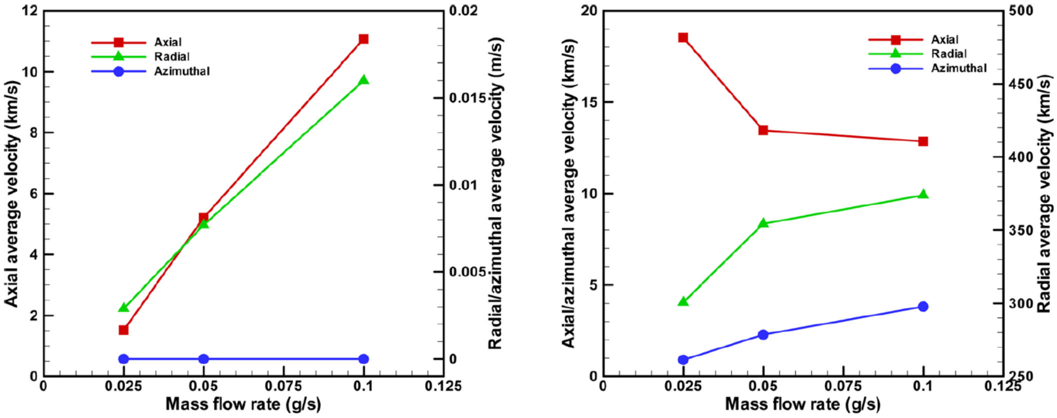


(a) Ion average velocity

(b) Electron average velocity

Figure 3.6: Axial, radial and azimuthal velocities of ions (left) and electrons (right) for an applied magnetic field range of [0.034-0.102] T, with Argon propellant with a mass flow rate of 0.1 g/s and discharge current of 1 kA [38].

While the acceleration mechanisms of SF-MPD thrusters are rather well understood, those of AF-MPTs need much theoretical background and experimental tests. Models such as that of Mikellides and Turchi [38] or the modified Tikhonov model seen in the Literature Study [23] cover only part of the operation range. in addition, older models such as these are tailored for regulated discharge current



(a) Ion average velocity

(b) Electron average velocity

Figure 3.7: Axial, radial and azimuthal velocities of ions (left) and electrons (right) with an applied magnetic field, with Argon propellant for a mass flow rate range of [0.025-0.1] g/s and discharge current of 1 kA [38].

operations (high current, low voltage), which is not the operation mode of SUPREME. As such, better models for a wider range of operation parameters needs to be developed and a better understanding of how each vector of each acceleration mechanism influences another thrust vector. For example, the radial component of the self-field mechanism pushes particles towards the cathode, increasing their radial velocity, thus, increasing the radial discharge current which increases the axial self-field acceleration. Furthermore, understanding of the acceleration mechanisms is further complicated in an AF-MPD as previously mentioned. The Hall acceleration can cause both positive and negative contributions. In the upstream region where usually the magnetic field is converging, the Hall mechanism has a negative impact on thrust. This may be countered by the self-field and gas dynamic components, but the scientific community is still analysing this aspect. In addition, the non-electromagnetic component (gas dynamic) needs to be taken into account as well. It is affected by plasma pressure as the plasma heats up through Joule heating and similarly to an electro-thermal arcjet, the plasma expands in the nozzle and accelerates [38]. Tang et. al. (2012) [38] used a full PIC model for the simulations and explained the two available methods for total thrust computation. The first method would be the use of the following formulae:

$$T_{sf} = \int_V \sqrt{(j_r B_\theta)^2 + (j_z B_\theta)^2}\,\mathrm{d}V, \tag{3.8}$$

$$T_{sw} = \int_V (-j_r B_z + j_z B_r)\,\mathrm{d}V, \tag{3.9}$$

$$T_H = \int_V \sqrt{(j_\theta B_r)^2 + (j_\theta B_z)^2}\,\mathrm{d}V, \tag{3.10}$$

$$T_{gd} = \int_S (kn_i T_i + kn_e T_e + kn_a Ta)\,\mathrm{d}S, \tag{3.11}$$

$$T_{total} = \frac{T_{sf} + T_H}{2} + \sqrt{\left(\frac{T_{sf} + T_H}{2}\right)^2 + T_{sw}^2 + T_{gd}^2}, \tag{3.12}$$

where k is a constant, n is the number density, T is the temperature and *i*, *e* and *a* refer to ions, electrons and neutrals. The total thrust formula comes from a simplification that all the energy is converted to axial energy. However, computations showed that a classical summation of the four terms is slightly more accurate than Equation 3.12. While the second method would be the use of the classical rocket formula and yielded results that are closer to the experimental data:

$$T_{total} = \dot{m} u_{exit} + pA_e, \tag{3.13}$$

where u_{exit} is the mean velocity at the exit where the sum of the product of the number density at a specific velocity at the exit with that velocity is normalised by the sum of those number densities. p is the plasma pressure, while A_e is the exit area.

3.8. Theory Summary

This section started by explaining the importance of anode power deposition and the much larger losses resulting from the anode over cathode. The most important source for anode power deposition is the electron current conduction, followed by smaller sources such as cathode radiation and convective heat transfer from the plasma. Considering the SUPREME thruster will operate in steady state mode, the thermionic current density was also introduced together with its relationship with the net current density from the anode and the electron current density through the anode. The former one is directly influenced by the anode fall voltage. Then, the current density of the cathode surface was discussed.

The types of propellants that can be used was presented, along with their characteristics. The propellant choice is a rather complex decision around four main factors: specific impulse level (including Alfvén velocity), the plume deposition level, ionisation levels and cost. The most likely propellant for the SUPREME thruster is Argon due to it having low first ionisation level (but larger than Lithium), no deposition problems, not a very heavy atom compared to other propellants and a low cost.

Then, the erosion caused by sputtering (ion erosion and deposition in the region) was presented, with the peak sputtering yield achieved for 45° for various materials and energies. A few measures to decrease the consequence/gravity of this phenomenon are the use of special materials and coatings, propellants with large atoms and low first ionisation level with high ionisation values for the next levels.

In the second part the operational parameters were discussed. The advantage of higher thrust in steady-state operation were stated, in contrast with the better use of pulsed-state for thruster testing. The differences between these operation types are decreased with an increase in operating power.

Several relationships were presented between thrust and discharge current (linearly), applied magnetic flux (linearly), anode radius (parabolically), mass flow rate, decreasing cathode radius, increasing electrodes length. The AFV plays a very important role in the functioning of an MPDT and a possible measuring method during experiments, was introduced through the use of an emissive Langmuir probe. To measure the anode temperature an optical pyrometer can be used. Sputtering can be measured by using a profileometer and a spectrographic analysis. Tests should also be conducted with an LaB_6 cathode for the power range required by SUPREME.

In space plume ID can be achieved by measuring the plume brightness with the optical pyrometer and by detecting a luminous cathode jet. Other quantities that can be measured to aid plume ID were also presented.

In the third part the structure of the code was discussed, with focus on the inputs and outputs needed for the software. Given the relationship between the Hall parameter, the induced Hall current and the discharge radial current, the first and second parameters can be easily computed at each cell. This in turn would result in the radial current value, which can be linked to the electric field values and R_m, the magnetic Reynolds number (the reader is advised to check the Literature Study [23] for more information regarding R_m).

The thermionic current can also be computed at each cell on the boundary, with predefined parameters together with the temperature of the surface and the Electric field. In addition, given the different theories regarding the electric field in an AF-MPDT, the AFV shall be computed (and given as output) in order to use different models to link it with the radial E-field. Computing the density of species at each cell is very important for results such as the AFV values, anode starvation, field shielding etc. Considering the operating mode of SUPREME, the total discharge voltage would be given as input. Also, the energy sputtering yields and the sputtering threshold should also be provided as inputs, which depend on the materials chosen and the tests conducted. The three boundaries were also stated: backplate, inner walls and the free boundary.

An updated formula for the self-field thrust in the partially ionised regime was also presented. The section ended with a conflicting opinion. While it is widely agreed in the scientific community that the Hall thrust is linearly proportional with the induced Hall current and the applied B-field, some of the papers presented state that the applied magnetic field does not influence the Hall thrust (higher B-field → lower Hall current), statement which might be disagreed by the majority of the scientific community.

The main two benefits of the PIC-MCC approaches over MHD are: considering nonlocal dynamic effects of all the particles and more accurate and easier plasma transport coefficients. Also, MHD works in the context of low Knudsen numbers, while kinetic approach works for high high values (larger than 1). Furthermore, there are two types of PIC models: full and hybrid. While the hybrid might be easier to develop as it would treat the neutrals and ions kinetically and the electrons as a fluid, the magnetic field condition, $\nabla \wedge \vec{B} = \mu_0 \vec{J}$, should not be approximated to be zero. The use of the hybrid approach has to respect the three assumptions discussed.

The (fully kinetic) PIC-MCC methods have the simulation time limited by the electron-time scale. To reduce the computational power multiple solutions were suggested: the use of super-particles, consider heavy particles to have a lighter mass for acceleration convergence and increase the Debye length.

Given that the materials of the thruster, the propellant type are known, they can be used to compute the mean free path (MFP) and have it as an input in the simulation. However, the MFP changes with the density of particles and the cross-section of the collision. Thus, a more accurate approach would be to compute it at each interaction. In addition, the cross-section is influenced by the energy of the colliding particles. Thus, it may be more beneficial to calculate the cross-section at each interaction as well. Also, the four types of particle-surface interactions have to be implemented: sputtering (erosion), SEE, electron absorption, ion neutralisation, scattering.

An important aspect in plasma simulation is the ionisation treatment. As plasma in MPDTs are in ionisational nonequilibrium, the code should use a multilevel equilibrium ionisation models. Also, ionisation cannot be treated in fluid models and an approximation of full ionisation is made for the CFD approach. In addition, there can be difference between theoretical models and experiments due to the continuum assumption starting to fail in low density regions.

The electric field field was shown to be constant at a specific radius along the symmetry axis and its axial field strength shape was stated (similar to Gaussian that peaks in the middle of the thruster). The induced B-field presented a similar shape that peaks in field strength in the middle. Tests are still needed in order to understand if there is a correlation between applied and induced magnetic fields that can cause this. Also, tests at very low induced and applied B-fields yielded swirl as the main thrust component, which is not expected in the case of SUPREME. If density is treated uniformly in the azimuthal direction, a simple density formula along the axial direction can be used. However, more accurate models or simulations might yield more complicated density levels. Higher applied magnetic field increases thrust, which increases the rate at which ions exit the chamber, which adds to the lower expected density towards the thruster exit. In addition, other relationships were analysed with regards to changes in magnetic fields.

Researchers have shown that the electron temperature is on average 2-3 eV, but it varies with the location (higher in the bulk and lower near the anode). The electron temperature peaks close to the middle of the thruster and it increases with increasing applied B-field (higher velocity). The temperature of ions do not have such a clear effect from the applied B-field, but some researchers have shown it ranges from 0 to 12 eV along the thruster. These differences come from the many types of charged particle acceleration in the thruster.

Through simulations, the relationship between velocity components and the three main operating parameters were analysed: an increase in discharge current causes a linear increase in the velocities (axial, radial, azimuthal) of the charged particles. An increase in the applied B-field also caused in increase in these velocity components, although not linearly. On the other hand, the mass flow rate caused only some components to increase, while the axial velocity of the electrons decreased. The results discussed shown values of 10^4 m/s for ions and $3 \cdot 10^5$ m/s for the electrons. The importance of the radial velocity component of the electrons was emphasised also by the fact that the electrons carry most of the current.

Finally, the need for better models was emphasised, in the context of steady-state, high regulated voltage AF-MPD thruster. The influence of different thrust components of the three velocity components was briefly explained, together with the contrast of the Hall thrust that has a negative contribution in the chamber entrance and positive towards the exit. The two models used with the full PIC approach by some researchers were also shown: computation of each of the four thrust components and their sum and the more accurate method of starting from the rocket equation for thrust.

With the theoretical foundation from this chapter, additional requirements were formulated, such as pertaining to the current (thermionic), the phenomenons around the electrodes etc. In addition, this chapter also helped with a better understanding of the simulation results from Chapter 5 and possible developments of the next iterations of a simulation code.

4 Software Design Through A Systems Engineering Approach

This section goes into detail about the design of a plasma simulation software for an AF-MPDT using systems engineering (SE) principles. The framework of the SE approach comes mainly from the lecture notes of Prof. Eberhard Gill [14], but also from the guidance of the TU Delft and NSS supervisors guiding this project and from the interactions with the relevant parties involved. The stakeholder/system requirements were detailed in Section 2.1 and here the lower level requirements are detailed. Attention was given to the product (plasma simulation software) verification and validation (V&V) methods. The focus on V&V of this thesis was on the verification of requirements (if a requirements cannot be verified, it is not a good/acceptable requirement), while validation was briefly discussed. This is due to the long roadmap that plasma simulation softwares for AF-MPDTs have, which would be done in iterations and would finish after this thesis project ended. The diagrams used to help guide the code design were context diagram, Input-Output (IO) matrix and sequence diagram with some aspects that stem from the Design Option Tree which were explained. As the focus of the thesis was not related with the actual code development of the product, detailed description or architecture of the software was not given. Other SE aspects that were discussed include fault tolerance (although very limited as this is a software only product not developed as part of this thesis), mitigation and risk management, system costs, which were treated as a design variable.

4.1. Tests Definitions

Before the types, methods and levels of tests are detailed, the verification and validation notions have to be explained. Verification provides the proof of compliance, for each requirement and is seen as a quality control process. Thus, through verification it is ensured that the product follows/complies regulations, the design specifications and the product documents.

This procedure shall be applied in all stages: of the product development, scale-up and production and the requirements that are presented in this document can be verified at different stages as well, as the software evolves. This is mostly an internal process and such is the case here. The NSS company would provide the systems engineering design for a plasma simulation software (the output of this project), with all the relevant requirements, to a simulation software development company. As such, that company would conduct the verification procedure, before the final/agreed version of the code is sent.

There are four product verification methods used in this project. The first method consists of tests designed to ensure that the relevant component fulfils the corresponding requirement. This is the preferred method, however, it is the most time consuming and costly out of all the methods. Secondly, there is analysis which consists of providing analytical solutions as proof for the requirements to be fulfilled. The third method is inspection, in which the relevant component is analysed (most common, visually) to check if it meets its corresponding requirement. Lastly, there is the similarity/demonstration method in which the requirement fulfilment is demonstrated by performing a check of the component or of a specific action. If the requirement is similar to another already verified requirement, then it follows that this requirement to be verified fulfils its documentation.

The description of the verification work and its success criteria are described in Section 4.3.

Validation provides evidence that the product accomplishes the expectations of the stakeholders and represents a quality assurance process, as opposed to verification (quality control). This is most likely an external process and corresponds to the fitness of purpose label for the purpose of stakeholders, NSS in this case. As such, the validation can be done in the later stages of the product development.

Similar to verification, there are four product validation methods. First, there is the end-to-end information system testing which is meant to assess the fit for purpose level of the compatibility of information systems involved. The second method is the mission scenario test which consists of implementing the use/flight-like conditions and check if the hardware and software (just software in this case) can execute the mission. Then there is operations readiness tests which involves demonstrating the compatibility of all the units involved (software, hardware, ground segment). This is however not really suited just for the plasma simulation software product, but for the software in the context of the Digital Twin. The fourth method is the stress-testing and simulation which analyses all the results to check the robustness of the system and how it affects the performance of the product, and the fault conditions.

Now that the verification and validation methods were discussed, the three major test types have to be introduced. The first and most common type are the functional tests. This type is responsible for verifying all the functional requirements at all system levels and it was the main type to be used in relation to the software for this project. The second type are the integration tests, which oversees the mechanical, electrical and/or software integration. This type was also relevant for the project in the context of the software integration within the Digital Twin. The third type are environmental tests in which the products are tested for fulfilling their functions while resisting constraints in their operational time frame. This was not a relevant part for this project as the a fully operational software was not developed by the time this project was finished. While the environmental tests type might seem redundant, a simulation software might shed some life on the backtracking phenomenon and the consequent charge buildup. This is especially important if SUPREME would be used on space stations where humans would live.

In addition to the test types there are also test levels. The first of the four levels is the development or demonstration level tests. They concern new configurations and validating novel designs. This level was almost over the entire thesis project. The second level is represented by the qualification tests that demonstrate the fitness of the design manufacturing according to the requirements. This was slightly approach towards the end of this project. The final two levels are acceptance tests and operation tests. The former aims to demonstrate the level of specification agreements with the product, while detecting the remaining errors, faults, defects. The later level implies the demonstration of an exact replica of the product in the target environment and under the mission constraints to achieve conditions as close as possible to the real mission.

4.2. Lower Level Requirements

For a complete design of the software, the lower level requirements that flow from stakeholder/mission requirements seen in Section 2.1, have to be detailed. In order to do this, the theoretical sections served as the basis, while discussion with the multiple stakeholders involved and exterior parties provided for a more accurate systems engineering design.

The following requirements were found, each having one of the four (test - T, analysis - A, inspection - I, demonstration - D) verification methods. For each requirement its type was either functional (FUN), performance (PER), interface (INT), constraint (CONSTR - mentioned in Section 2.1) and design (DES). Functional requirements are representative of the functions that the (sub)system/product has to fulfill. Performance ones complement the functional requirements with technical values giving a more detailed view into the (sub)system. Interface requirements describe the interaction between the plasma software subsystems, system with the other DT systems. Constraints were already discussed previously, while Design requirements detail the construction, structure of a (sub)system.

Similarly to Section 2.1, the identifiers used were: time (TIME), acceleration mechanisms (AM), thrust accuracy (TACC), losses accuracy (LACC), erosion accuracy (EACC), anode fall voltage (AFV), user interface (UI), cost (COST), high applied magnetic field (HAMF), magnetic Reynolds number (MRN), diagnostic (DIAG). In the next part (Section 4.3) an explanation of each requirement and its verification methods are provided.

1. NS-TIME-DES-1:The software shall use code specific accelerated processing. → T/I
2. NS-TIME-DES-1-1: The software shall use GPU parallelization. → T/I
3. NS-TIME-DES-1-2: The software shall use a super-particle representing at least 10 particles. → T/I
4. NS-TIME-DES-1-3: The code shall use algorithm optimisation. → T/I
5. NS-TIME-DES-2: The software shall have multiple physics levels of the models. → I/D
6. NS-TIME-DES-2-1: The software shall have simplified physics models running faster. → T/I
7. NS-TIME-DES-2-2: The software shall allow disabling at least one of the acceleration mechanisms in a simulation. → T/I
8. NS-TIME-DES-3: The software shall deliver the corresponding outputs for each of the three acceleration stages: discharge, confinement, divergence. → I/D

9. NS-AM-FUN-1: The models used shall include the swirl acceleration mechanism. → T/A/I
10. NS-AM-FUN-1-1: The swirl acceleration mechanism shall have the two azimuthal components. → A/I
11. NS-AM-FUN-1-2: The software shall check what percentage of the azimuthal kinetic energy is converted into axial kinetic energy in the magnetic nozzle. → I/D
12. NS-AM-FUN-2: The models used shall include the Hall effect acceleration mechanism. → T/A/I
13. NS-AM-FUN-2-1: The Hall acceleration mechanism shall have the radial component. → A/I
14. NS-AM-FUN-2-2: The Hall acceleration mechanism shall have the axial component. → A/I
15. NS-AM-FUN-3: The models used shall include the gas-dynamic acceleration mechanism. → T/A/I
16. NS-AM-FUN-4: The models used shall include the self-field acceleration mechanism. → T/A/I
17. NS-AM-FUN-4-1: The self-field acceleration mechanism shall have the radial component. → A/I
18. NS-AM-FUN-4-2: The self-field acceleration mechanism shall have the axial component. → A/I
19. NS-AM-FUN-5: The software shall treat the interaction between the induced and the applied magnetic fields. → T
20. NS-AM-FUN-6: The software shall consider at least four types of collisions in the first iteration. → I/D
21. NS-AM-PER-6-1: The software shall consider ionisation collisions. → T/I
22. NS-AM-FUN-6-2: The software shall consider elastic collisions. → T/I
23. NS-AM-FUN-6-3: The software shall consider excitation collisions. → T/I
24. NS-AM-FUN-6-4: The software shall consider momentum exchange collisions. → T/I
25. NS-AM-FUN-6-5: The software shall consider charge exchange collisions from the second iteration. → T/I
26. NS-AM-FUN-7: The software shall consider boundary collisions. → T/I

27. NS-TACC-FUN-1: The software shall ensure that the electric field in the anode sheath is higher than in the bulk region. → D
28. NS-TACC-FUN-2: The software shall ensure that the electric field in the cathode sheath is higher than in the anode sheath. → D
29. NS-TACC-PER-3: The accuracy of electric field simulation results shall be less than 10% from experimental values. → T
30. NS-TACC-FUN-3-1: The software shall compute the radial electric field. → A/D

31. NS-LACC-PER-1: The accuracy of the anode losses results shall be less than 10% from experimental values and models. → T/A
32. NS-LACC-PER-2: The accuracy of the electron backtracking level should be less than 10% from experimental values. → T
33. NS-LACC-FUN-3: The software shall consider the current density on the cathode surface. → A

34. NS-EACC-PER-1: The accuracy of the anode erosion results shall be less than 10% from experimental values. → T
35. NS-EACC-PER-2: The accuracy of the cathode erosion results shall be less than 10% from experimental values. → T
36. NS-EACC-PER-3: The accuracy of the insulator/backplate erosion results shall be less than 10% from experimental values. → T

37. NS-AFV-FUN-1: The software shall consider the level of electrons escaping through the anode. → D
38. NS-AFV-FUN-2: The software shall consider the anode starvation. → D
39. NS-AFV-FUN-3: The software shall take into consideration multiple inputs that influence the anode fall voltage. → I/D
40. NS-AFV-FUN-3-1: The software shall consider the influence of mass flow rate on the anode fall voltage. → I
41. NS-AFV-FUN-3-1-1: The software shall simulate cathode mass flow rate. → D
42. NS-AFV-FUN-3-1-2: The software shall simulate anode mass flow rate. → D
43. NS-AFV-FUN-3-2: The software shall consider the influence of applied magnetic field on the anode fall voltage. → I
44. NS-AFV-FUN-3-3: The software shall consider the influence of the plasma density on the anode fall voltage. → I
45. NS-AFV-FUN-3-4: The software shall consider the influence of plasma temperature on the anode fall voltage. → I
46. NS-AFV-FUN-3-5: The software shall consider the heat flux to the anode. → D
47. NS-AFV-FUN-4: The software shall compute the anode sheath thickness. → D
48. NS-AFV-FUN-5: The software shall compute the cathode sheath thickness. → D
49. NS-AFV-FUN-6: The software shall compute the anode sheath voltage. → A/D
50. NS-AFV-FUN-7: The software shall compute the cathode sheath voltage. → D
51. NS-AFV-PER-8: Anode power deposition shall consider at least 2 sources. → I/D
52. NS-AFV-PER-8-1: The software shall ensure that the electron current conduction in the anode accounts for at least 60% of the anode power deposition. → T
53. NS-AFV-PER-8-2: The software shall ensure that the electron current conduction in the anode accounts for up to 95% of the anode power deposition. → T
54. NS-AFV-PER-8-3: The software shall ensure that the cathode radiation accounts for at least 5% of the anode power deposition. → T
55. NS-AFV-PER-8-4: The software shall ensure that the cathode radiation accounts for up to 35% of the anode power deposition. → T
56. NS-AFV-PER-8-5: The software shall ensure that the convective heat transfer from hot plasma accounts for up to 5% of the anode power deposition. → T

57. NS-UI-DES-1: The software shall allow to run pre-configured simulations. → I
58. NS-UI-DES-2: The software shall allow to run saved simulations. → I
59. NS-UI-DES-3: The software shall allow to save simulations. → I
60. NS-UI-DES-4: The software shall allow to put inputs before running the simulation. → I
61. NS-UI-DES-4-1: The software shall allow the user to input an initial propellant temperature. → I
62. NS-UI-DES-4-2: The software shall allow the user to input an initial propellant velocity. → I
63. NS-UI-DES-4-3: The software shall allow the user to input an initial propellant density. → I
64. NS-UI-DES-5: The software shall allow the user to change inputs during the simulation. → I
65. NS-UI-DES-6: The software shall save characteristics of the materials from previous simulations. → I
66. NS-UI-DES-7: The software shall show completeness percentage. → I

67. NS-UI-DES-8: The software shall allow the user to input both eV and J units for conversion by the kernel. → I
68. NS-UI-DES-9: The software shall allow the user to select a surface by pressing on it on the visual object. → I
69. The software shall allow the user to change the physical models used as a plug-in type. → I
70. NS-UI-DES-11: The software shall allow to user to modify the physical models without replacing them. → I
71. NS-UI-DES-12: The software shall offer the choice of a fixed electrode potential. → I
72. NS-UI-DES-13: The software shall offer the choice of a changing (floating) electrode potential. → I
73. NS-UI-INT-14: The software shall provide outputs as a CSV file. → D
74. NS-UI-INT-15: The software shall provide outputs as an Excel file. → D
75. NS-UI-INT-16: The software shall be able to get all the inputs from the DT. → T
76. NS-UI-INT-17: The software shall provide outputs that can be used by the DT. → T

77. NS-COST-INT-1: The first iteration of the software shall cost up to [TBD] euros. → I
78. NS-COST-INT-2: The second iteration of the software shall cost up to [TBD] euros. → I
79. NS-COST-INT-3: Adding specific processes shall cost up to [TBD] euros. → I
80. NS-COST-INT-4: Maintenance and run costs shall be up to [TBD] euros. → I
81. NS-COST-INT-5: Personalised help shall be up to [TBD] euros. → I

82. NS-HAMF-DES-1: The software shall be designed to run in a voltage regulated mode. → T
83. NS-HAMF-DES-2: The software shall be designed to run in a steady-state operation mode. → D
84. NS-HAMF-DES-3: The software shall allow updates in the magnetic field strength inputs during the simulation run. → T

85. NS-MRN-FUN-1: The software shall compute the plasma parameter. → A/D

86. NS-DIAG-FUN-1: The software shall allow the user to set diagnostic points on surfaces. → I
87. NS-DIAG-FUN-1-1: The software shall allow the user to set the diagnostic points by inputting their coordinates. → I
88. NS-DIAG-FUN-1-2: The software shall allow the user to set the diagnostic points by selecting them on the visual object. → I
89. NS-DIAG-FUN-2: The software shall allow the use of tracking particles. → I/D

4.3. Requirements Verification Analysis

As previously discussed, the requirements found have to be verified by one or more of the four methods. The first batch of requirements refers to the time taken from the start of the simulation until its end that results in output files and parameters. The top requirement NS-TOP-CAP-1 can be tested by running the simulation and timing the runtime.

For the simulation to be faster, the software shall be codded using good programming practices (NS-TIME-DES-1), defining classes instead of creating many objects with similar functions, defining functions instead of writing multiple times similar processes, using GPU parallelisation, superparticles, using stored data whenever possible etc. In addition, the simulated interactions can be split into many easier computations and be ran in parallel with many functions using the properties of GPUs (NS-TIME-DES-1-1). Both the previous parent and daughter requirements can be inspected (i.e. looking at the code and see how it is structured, whether it is modular, if GPU use was implemented) and they can also be tested by conducting a simulation and measuring the time before and after these measures. The use of superparticles (NS-TIME-DES-1-2) is common in MCC simulations, where typically ten particles are averaged into one. A balance must be reached between the time saved and the accuracy of the simulation. To further decrease the computation power, thus, the time, as previously explained an optimised code shall be used (NS-TIME-DES-1-3). In addition, libraries can be used that would result in a more efficient and faster code. Both child requirements can be tested by running the simulation and measuring the time taken and can be inspected by looking at the code to see if these methods were implemented.

The time is also impacted by the detail of the physics models used (NS-TIME-DES-2), either the accuracy of each formula and its implementation or the use of certain formulae. The physics models used can be inspected in the code to ensure a range of options is given. A demonstration of this requirement can be conducted by checking the output file for the accuracy against known standards and checking certain parameters influenced more or less by the changes. The two child requirements that follow are the option of using simplified physics models (NS-TIME-DES-2-1) and the option of not involving certain acceleration processes (NS-TIME-DES-2-2), such as the self-field acceleration. Simplified models would reduce the accuracy of certain equations such as the transferred heat to the anode, losses present, while removing one of the acceleration mechanisms would have an important impact on the accuracy of the thrust. Testing them can be done by conducting simulations and compare the time saved. In addition, the code rules can be inspected to see if simplified versions of the models were implemented.

As mentioned previously, it is faster and cheaper to store and reuse data than rerun a simulation and it is also faster. As such, requirement NS-TIME-DES-3 ensures that after each of the three stages in an AF-MPD thruster the software saves the corresponding output files and continues with the simulation. This way, if the simulation is stopped suddenly, the previous stages are not lost. The code can be inspected for the output file creation switch and the output files can be checked to see the three separate stages.

The second batch of requirements refers to the four acceleration mechanisms that shall be included in the simulation. The top requirement NS-TOP-CHAR-2 can be verified by demonstration/checking the output file for the individual thrusts and compare them to what would be expected. This requirement can be verified by analysis and inspection of the code to see if the implemented thrust formulae are present and are correct according to the physics models used.

Looking at requirements NS-AM-FUN-1, 2,3 and 4, they can be tested by conducting simulations and check if the particles are exiting as expected. Also, the output files can be checked to see if the values are within the expected ranges for the individual thrust mechanisms. All four can also be verified by inspection and analysis. In this context, this batch, inspection refers to checking if the requirement has been implemented in the code in any form, while analysis means checking if the formulae within the code match the formulae of the physics models used. Going to the first child requirement, NS-AM-FUN-1-1, according to the interaction between the discharge current and the applied magnetic field, there are two azimuthal components. This should be inspected by checking if they are implemented in the code and according to the analysis of the physics models, the code shall use the correct formulae. The second child requirement, NS-AM-FUN-1-2, refers to the conversion process of the azimuthal kinetic energy (KE) from the swirl to useful axial KE. The code can be inspected for such an implementation. Given that there is not an exact formula, this requirement shall be demonstrated by checking the output files from the simulation. A certain percentage of the KE before conversion shall be transformed into useful KE, however, in the validation methods the correct percentage/optimisation shall be decided. Both NS-AM-FUN-2 and 4 have similar child requirements (NS-AM-FUN-2-1, NS-AM-FUN-2-2 and NS-AM-FUN-4-1, NS-AM-FUN-4-2). Their verification is similar to that of req. NS-AM-FUN-1-1 and they can be inspected by checking for their presence in the code and can be verified through analysis by checking their correctness according to the physics models used. Another important aspect is the interaction between the self-induced and applied magnetic fields. As previously explained in Section 3.7.3, the interaction process is not properly tested and as such, tests have to be conducted to see if the simulation runs smooth in the context of a simplifying assumption: magnetic fields interact constructively. There are five types of interactions involved in this case, however, for a faster code development, at least four shall be implemented in the first software version. This requirement (NS-AM-FUN-6) can therefore be verified through inspection of the code: the four classes/packages corresponding to the interactions shall be present. Also, this requirement can be demonstrated by looking at the output files and flag the different types of interaction. The corresponding child requirements (NS-AM-FUN-6-1, NS-AM-FUN-6-2, NS-AM-FUN-6-3, NS-AM-FUN-6-4 and NS-AM-FUN-6-5) can be tested by checking if the simulation does not encounter run errors with any of the interactions. In addition, their corresponding classes/packages in the code can be inspected to see if the interactions are implemented (see Section 3.7.1 or the Literature Study report [23] for details). The last part of this batch is about the boundary conditions in the code: NS-AM-FUN-7. By inspecting the code it the functions with the corresponding boundary conditions can be checked. In addition, this requirement can be tested by checking the simulation for errors and whether the code correctly interprets the boundary conditions.

The third batch refers to the thrust accuracy of the simulation compared with previous experimental results achieved. Thus, requirement NS-TOP-CAP-3 can be tested by running a simulation with the design parameters of a previously tested thruster and compare the resulted thrust accuracy of the simulation with that of the experimental results. As can be seen in Section 3.6 the electric field is higher, closer to the electrodes and it is lower and shielded (thus, approximately constant) in the bulk region (NS-TACC-FUN-1). This requirement can be demonstrated by checking the output files and looking at the electric field values at different points and throughout time (through the simulation). In addition, the electric field in the anode region shall be small than in the cathode region (NS-TACC-FUN-2), which can be demonstrated by looking at the output file and compare the values between the electrodes regions. Requirement NS-TACC-PER-3 comes from the need of understanding how the electric field is influenced by charged particles. In addition, since the top requirements are designed for 5% accuracy, the accuracy of the E-field topology can be worse. As such, a value of 10% was chosen based on Figure 3.3a. Only the radial component was used for this judgement as it is by far the dominant component in terms of magnitude. This requirement can be verified by comparing laboratory measurements with the results seen in the output files. Its child requirement (NS-TACC-3-1) involves the computation of the radial component of the electric field (the dominant component) using the formula stated in Section 3.7.1. Analysis can be used to compare the implemented formula with the one found in the research of this project. In addition, this requirement can be easily demonstrated by checking the output files for the radial electric field values.

The forth batch refers to the losses occurring during the thruster's mission. Top requirement NS-TOP-CAP-4 emphasises that the software has to simulate losses up to 10% either from experiments or models. Similarly to NS-TOP-CAP-3 and NS-TOP-CAP-5, it can be tested by running a simulation and comparing the results of the output files with those from the available data. As electrode loses that are of thermal nature can be tested in the laboratory, they are the ones compared against the simulation. Thus, requirement NS-LACC-PER-1 can be verified through testing by conducting a simulation and comparing the results of the anode temperature and of the losses with the experimental values. In addition, through analysis the implemented thermal loss model can be checked whether it is in line with the physics models used. If experiments allow for measuring the escaping particles, this can also be used further in the development of the code as a checking procedure. Going to requirement NS-LACC-PER-2, the backtracking of the electrons can be measured in the laboratory by counting the total charge in the back of the thruster, from the electrons moving on the magnetic field lines. This can be tested by running a simulation and comparing the experimental measurements with the output files. Lastly, requirement NS-LACC-FUN-3 pertains to the different densities of A/m^2 that the cathode has along its axial direction. This requirement can be demonstrated by checking that the output files show different values and that it matches what it is expected according to the physics models used.

The fifth batch refers to the erosion experienced by the thruster. As explained above, top requirement NS-TOP-CAP-5 can be verified through testing by comparing the output values with the experiments. The three lower level requirements NS-EACC-PER-1, NS-EACC-PER-2 AND NS-EACC-PER-3 pertain to the three types of physical boundaries. These requirements can be tested in a straightforward manner by running a thruster in a laboratory and analysing the erosion sustained in that time. Also, a simulation for that thruster with the same operating parameters shall be conducted and the erosion results shall be compared with those of the thruster.

The sixth batch of requirements pertain to the anode fall voltage aspects. The top requirement NS-TOP-CAP-6 ensures that the AFV shall be considered in the plasma simulation software. This can be verified by demonstration: checking the output file would show the impact it has on thruster efficiency, erosion, losses as they would be affected differently by different AFV values. The electrons in the plasma have a wide range of values, mostly following a Maxwell-Boltzmann distribution. As such, more energetic electrons can leave the thruster through the anode causing loss in efficiency and perhaps charge build-up problems. Requirement NS-AFV-FUN-1 ensures this is considered in the code and its implementation can be demonstrated by checking the output files for electrons that are deleted from the simulation when they leave the simulation region. For better visuals, the code can flag in the output files each event when an electron is deleted in the anode region. An element that contributes to AFV is anode starvation (NS-AFV-FUN-2): when the density of particles is much lower in the anode region that the rest of the thruster. This can also be demonstrated by looking at the output files which should show a lower density in this region. There are multiple parameters influencing AFV and they shall be accounted for in the software (NS-AFV-FUN-3). This can be demonstrated as the output files should contain the corresponding parameters, but an inspection of the code modules would show the relationships between the anode fall voltage and these parameters. The child requirements NS-AFV-FUN-3-1, NS-AFV-FUN-3-2, NS-AFV-FUN-3-3 and NS-AFV-FUN-3-4 all refer to the parameters that influence the AFV. They can be verified by inspection: the corresponding modules shall include their relationship with the AFV. This would allow researchers to understand how changing these parameters in a certain thruster geometry would affect the anode fall voltage. In addition, the two lower requirements NS-AVF-FUN-3-1-1 AND NS-AVF-FUN-3-1-2 ensure that both anode and cathode mass flow is considered in the software. These can be demonstrated by looking at the output files for the injected mass flow rates compared to what the software received as inputs (they should be equal). Child requirements NS-AFV-FUN-3-5 provides a very useful output, the heat flux to the anode. This is needed in order to calculate the needed cooling systems such that the thruster efficiency is optimised and to ensure the superconducting system is not affected by the anode radiation. In addition, cooling to an appropriate temperature would extend the anode lifetime. This can be demonstrated by checking the output heat flux value. Two other important aspects in the AFV and the thruster efficiency are the electrode sheath thickness (NS-AFV-FUN-4 AND NS-AFV-FUN-5). By checking the output files from the simulation, the values can be demonstrated to show non-zero values. The anode sheath voltage is required as explained in the theory part of this report, thus, requirement NS-AFV-FUN-6 ensures its direct computation. This requirement can be demonstrated by demonstration, checking the output file for its values) and through analysis by comparing the implemented formulae with the physics models desired. The cathode sheath voltage is also relevant (NS-AFV-FUN-7) as it might show a connection with the AFV and it might provide an explanation for the main electrons acceleration mechanism (highest velocity were shown to be in this region, although some teams might not directly agree with this). This requirement can be demonstrated by showing the cathode sheath voltage values in the output files. The anode power deposition is the main power dissipation process as explained in Section 3.2: NS-AFV-PER-8. Three sources for it were stated, but a minimum of two shall be implemented in the first software iteration. There shall also be the possibility of turning on and off processes such tailored processes and notice what the outputs might suggest, but this is emphasised by the need of a modular code. This requirement can be verified through inspection if there are at least two modules implemented and through demonstration by checking the number of anode deposition sources in the output file. The following child requirements (NS-AFV-PER-8-1, NS-AFV-PER-8-2, NS-AFV-PER-8-3, NS-AFV-PER-8-4 and NS-AFV-PER-8-5) refer to the three anode power deposition sources stated in Section 3.2 and within ranges found by Myers and Soulas (1992) [30]. Thus, they act a ranges for the simulation and a starting point for the operation parameters. However, the modular design of the software has to, once again, be emphasises: there shall be an easy, fast and straightforward way of turning off these ranges and their impact on the simulation if after multiple simulations and comparisons against laboratory results are conducted. All these five requirements can be verified by testing: conduct simulations in order to check from the output files how far the simulation results are from laboratory results.

The seventh batch is a very important one, but that has not had much attention in the field of plasma simulations for AF-MPDTs. This is because user interface does not come among the first places in a novel field (cutting edge of research). However, to ensure much faster advancements of this technology, designing the software with a reliable and easy to use graphical user interface (GUI) is vital. Top requirement NS-TOP-CHAR-7 refers to the design of the GUI such that stakeholders without coding knowledge can easily use the software. Like most of the requirements within this batch, the corresponding top stakeholder requirement can be verified by inspection of the GUI, but also by demonstration of ease of use by a person with no coding knowledge. That person would try to run simulations and change the different details of the simulation. Most of the requirements are verified by inspection by looking at the GUI and testing different functionalities of the software. As the inspection method is similar or the same for most of these requirements, most of the explanations in this batch would focus around the need for those specific requirements. Out of all the operation parameters and thruster geometries used, it is very likely that only a few of them would be used more often as they are more promising or small changes are needed around these parameters. As such, the software shall allow a user to load pre-configured simulations either with some operation parameters, geometry or at a certain stage in the simulation (NS-UI-DES-1). Also, sometimes it might be necessary to stop the simulations and continue later or a problem might cause the simulation to stop unexpectedly. Thus, the software shall allow the user to load any simulation available (NS-UI-DES-2). In order to load simulations, they have to be saved first, giving rise to requirement NS-UI-DES-3. While it might be the case that demo simulations software run with certain parameters, for advancements of the AF-MPD technology to occur, the software shall allow the user to input the desired parameters for each simulation or the software shall be able to use input parameters from another module within the Digital Twin (NS-UI-DES-4). This last requirement has the following child requirements: NS-UI-DES-4-1, NS-UI-DES-4-2, NS-UI-DES-4-3 which add to the total types of inputs possible, three others that pertain to the propellant and should not automatically run with predetermined inputs. In different context, such as changing the applied magnetic field during the simulation/run (as explained in the Literature Study report [23]) there might be the need to change at least one parameter during the simulation. The software shall allow the user to pause it and conduct the desired changes (NS-UI-DES-5). In most of the cases, while the geometry and the operation parameters might change quite frequently and without respecting certain ranges, the materials used for the thruster parts and the propellant types are much more limited. As such, the software shall allow the user to save all the corresponding details about these materials and propellants for future simulations (NS-UI-DES-6). To have all the details at hand and for the sake of work organisation, the completeness percentage shall be available to the user such that plans can be made around the time left for the simulation or to detect if there is an error that is causing the code to stagnate or show erroneous values (NS-UI-DES-7). There is some freedom given to the software developer team, here, as the percentage can be set to show completeness according to their definition. When working at the interaction between aerospace engineering and physics, different parameters are found in literature in units of Joules or eV. The software shall allow both units and convert in the back such that it is easier for the user to use the software (NS-UI-DES-8). Similar to tracker particles, there are tracker regions or perhaps the user plans to change certain whole regions or points of the thruster. While the software shall allow the input of an exact location on the thruster, that might not always be required. As such, the software shall allow the user to select a region or point on the thruster by simply pressing with the mouse on it (NS-UI-DES-9). As the technological readiness of the AF-MPDTs increases, along with knowledge from simulations and laboratory tests, the physical models used, such as the four main forces, would also improve or change. It can be thought of as an improvement over the entire operation range or different formulae for different operation ranges. Thus, by allowing the physics models to work as plug-in components (NS-UI-DES-10), even users with no experience can run the simulation and test different physical models they received. For NS-UI-DES-11, by modifying the physical models directly in the software, the user would save time. Depending on the consensus achieved by technical team, a user might want to run a thruster with either fixed or variable electrode potential and the code shall allow this (NS-UI-DES-12 and NS-UI-DES-13). This is mostly dependent on the design of the whole propulsion system, including superconductors, propellant tank etc.

The output data has to be easily readable by users and to be used by the other components in the digital twin or other data processing packages. As such, the software shall provide the output files in both CSV (NS-UI-DES-14) and Excel formats (NS-UI-DES-15). These last two requirements can be verified through demonstration by checking the type of the output files. As mentioned earlier, after the simulation software validation as a standalone and the successful integration within the DT, the software shall be able to receive and use inputs not only from users but from other parts of the DT (NS-UI-DES-16). Thus, test parameters as would come from the DT parts would be given as an input to the code in order ot verify this requirement. In addition, for the successful integration within the DT, the software shall also be able to provide the output parameters/data in a format the the other parts of the DT can use (NS-UI-DES-17). The type of data required by the DT would be given at a later stage. The last two requirements can be verified by testing the software within the DT framework.

In every systems engineering project, system costs should be treated as a design variable. This is especially relevant in the start-up environment which has limited founding. As such, the available plasma simulation software budget is limited and uttermost effort has to be put such that investments such as these would be supported through a ESA funding grants or European projects grants. In addition, as mentioned previously, the start-ups do not benefit of a large time-to-market range as large companies do. Thus, investments have to be conducted, but in an organised manner. As top requirement NS-TOP-CHAR-8 states, attention to costs shall be given through the entirety of the project's lifetime. This can be inspected by setting agreements with the software developing teams, ahead of time for multiple iterations at once and applying to as many grants as possible. All the lower level requirements can be verified through inspection by agreeing and signing contracts with the responsible parties. Especially in the context of developing a plasma simulation code that would be at the forefront of the research, the first two iterations (first covering the most important aspects and second correcting the issues and improving on the details) are the most important ones and the ones with the highest costs. As such, there should be a cut-off or a desired amount for first iteration (NS-COST-INT-1) and for the second iteration (NS-COST-INT-2). These costs can also be given a cut-off by the relevant grants/projects won. In addition, it is quite likely that there would be a need for the implementation of additional processes that perhaps are not part of what was agreed in the discussions for the iterations. As such, there shall be an approximate cost given for an additional process (NS-COST-INT-3). Another very important part are the maintenance costs which might be necessary during the use of the software and the run costs (NS-COST-INT-4) which imply the physical infrastructure (HPC, personal computers) and human resources needed. In order to reduce the unexpected costs that may come up, there shall also be an agreement on how much personalised help, outside the iterations, would cost and how much can the definition of personalised costs extend.

The ninth batch refers to the new type of running an AF-MPD thruster: at high applied magnetic fields, which was allowed by the use of high temperature superconductors. The previous simulations of AF-MPDTs involved low applied B-fields compared to the desired values for SUPREME and as such, there is no experience on simulation codes for high applied B-fields. Top requirement NS-TOP-CHAR-9 can be tested by running a simulation and making sure that there are no errors in the simulations. In addition, rough thrust values can be computed for the operating parameters used and compared with the output. A very important aspect of SUPREME is that it is used in high regulated voltage (NS-HAMF-DES-1). This requirements can be verified by testing the simulation: check if there are no errors that cause simulation to stop, check that the discharge current is constant while the voltage changes to the desired values, as it results from the output files. In addition, for higher performance as explained in Section 3.5, the SUPREME thruster would be ran in steady-state mode. The software shall be designed to accommodate this (NS-HAMF-DES-2) and this requirements can be verified by demonstration: checking the output files for the presence of thermionic current and the fluctuations simulated, especially for the discharge current and voltage. For example, as explained in the Literature Study report [23], the startup of the thruster might benefit from lower applied magnetic fields (faster startup and less erosion), followed by an increase to achieve high thrust efficiencies. Thus, requirement NS-HAMF-DES-3 is needed to ensure this option and it can be tested by varying the applied magnetic field input to the simulation, checking that the software does not crash and checking the output files for a smooth transition to higher applied B-field.

An important parameter for the AF-MPD thrusters is the magnetic Reynolds number (R_m) that influences the specific impulse, the acceleration of charged particles, the electric field and discharge current. As such, top requirement NS-TOP-CAP-10 ensures it is computed in order for the output files to yield its effect. This can be demonstrated by checking the output files. Another parameter that is linked with R_m is the plasma parameter and it shall be computed by the software (NS-MRN-FUN-1). In order to verify this, the physics models code can be analysed that the formula matches the models desired. In addition, it can be demonstrated by checking the output files and compare its value with the expected one.

The eleventh batch deals with the aspect of software diagnostics. It is vital for any product to have diagnostic procedures in place that might help the improvement procedures. This might be more complicated for software compared to hardware products as an error in the data might not be such a straightforward procedure of going back to find the problem. The top requirement NS-TOP-CAP-11 ensures that procedures and modules are put into place to understand the output and the possible errors that might come up in the simulation. This can checked by inspection of the GUI of the options to be ticked or the options of interacting with the geometry. In addition, it can be demonstrated by looking at the output files that there is extra output for what was necessary and that belongs to the diagnostic procedures. Previously explained requirement NS-UI-DES-9 builds on the option of interacting with the thruster structure, while requirement NS-DIAG-FUN-1 further builds on this by allowing the selected points to become diagnostic points. These selections shall provide detailed information from the simulation and the user shall be able to choose these points. This requirement can be verified by inspecting the GUI and how can the user interact with the geometry. This has two child requirements: choosing a specific point/region directly by pressing on the thruster geometry (NS-DIAG-FUN-1-1) and by inputting the exact coordinates of that point/region (NS-DIAG-FUN-1-2). Both can be verified by inspecting the GUI. The software shall allow the user to select points/regions on the thruster while there shall also be an option of setting these points by coordinates. However, the main advantage of diagnostic procedures is represented by tracking particles (NS-DIAG-FUN-2). The user shall be able to select a superparticle starting from a region in the thruster or from a certain time. These particles shall provide additional detailed information in the output files that would paint for the user a clear image of what is going on in the simulation.

4.4. Overview Diagrams

In addition to the requirements, there were a few additional steps for the design of a first iteration of a plasma simulation code. First, it needs to be put into the higher picture.

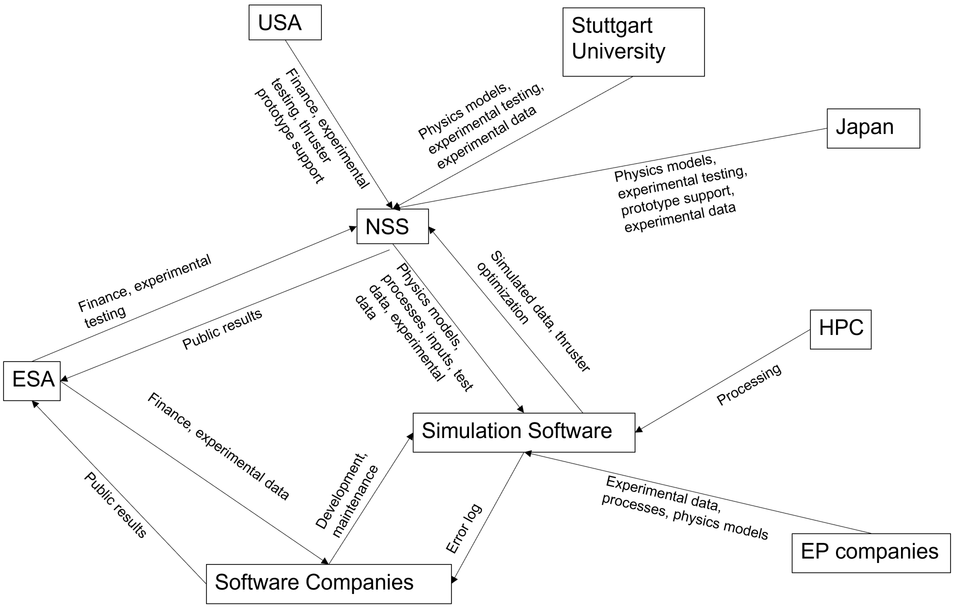
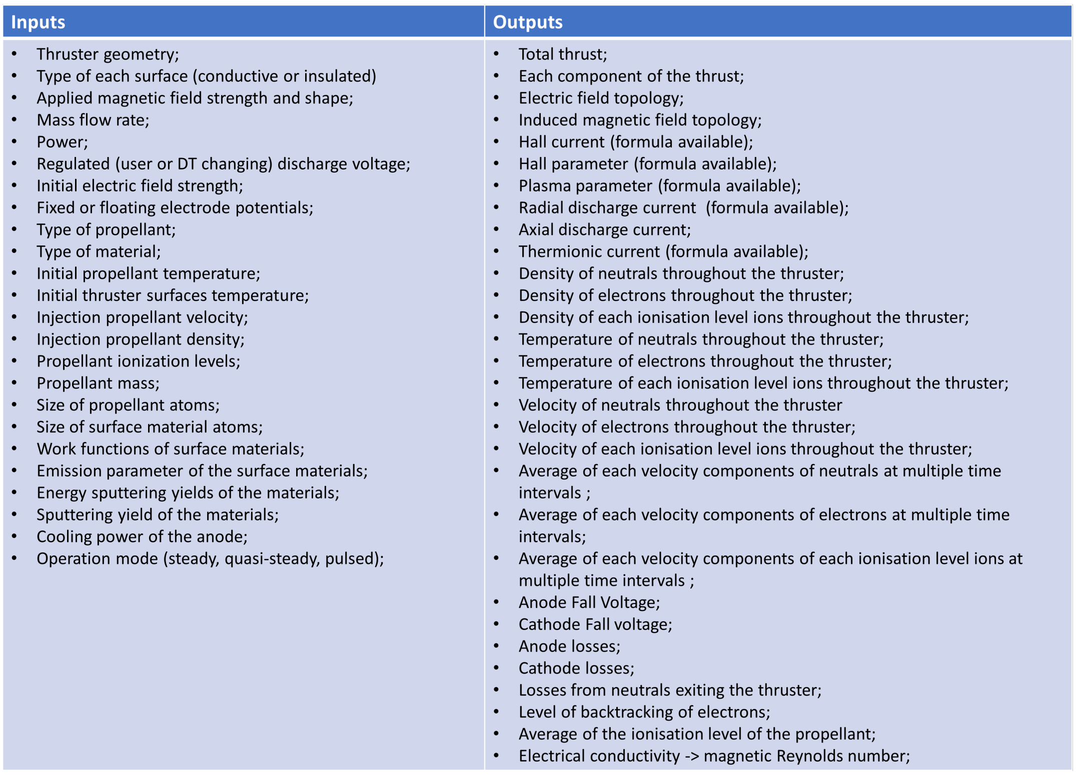

USA
Stuttgart University
Japan
NSS
HPC
ESA
Simulation Software
EP companies
Software Companies
Finance, experimental testing, thruster prototype support
Physics models, experimental testing, experimental data
Physics models, experimental testing, prototype support, experimental data
Finance, experimental testing
Public results
Physics models, processes, inputs, test data, experimental data
Simulated data, thruster optimization
Processing
Finance, experimental data
Public results
Development, maintenance
Error log
Experimental data, processes, physics models

Figure 4.1: Context diagram for the simulation software for NSS.

As such, the context diagram (see Figure 4.1) shows the related parts that have an influence on the software. Most of the physics models would come from Stuttgart University due to NSS collaborating with them. They might also be able to provide laboratory testing, in turn checking the quality of the simulation, as well as experimental data from previous tests. Trough its many research centres and universities, as well as experience with electric thrusters, Japan can also provide the same help as Stuttgart University, with the added help of building the SUPREME prototype, providing parts. USA also has experience with electric propulsion, thus, many testing and manufacturing facilities, which would be useful for the software and can speed up the plasma simulation software development. In addition, depending on the level of involvement desired, it is possible that there would be funding available as well. While stakeholders from USA and Japan are actively working with NSS, depending on the needs of the code before the full iteration is ready, as well as after, they can contribute more, by providing experimental data, theoretical help etc. However, these stakeholders do not interact directly with the software, but with NSS, which would process this data and provide the inputs for the code. In return, the software would provide the simulation data and the optimisation parameters. The software would most likely make use of high performance computing (HPC) farms to process the simulation and can take different physics models, experimental data and optimisation parameters from other similar electric propulsion companies that have already processed the relevant input. The software would also interact with the software companies responsible for its development. Thus, the error log would be sent to these companies and they would provide constant development and maintenance such that NSS can fulfil its goals with little to no delay. Another important stakeholder is ESA. They can interact directly with NSS to provide funding through grants and projects as well as experimental testing facilities, while NSS has to send ESA their results and progress. Depending on the type of funding scheme, ESA can also directly interact with the companies developing the simulation software by providing them with grants and experimental data from previous tests. In return, the software companies are expected to provide the results achieved. While no funding has been provided by ESA to NSS for plasma simulation software development, it represents a very relevant opportunity.

4.4.1. Input-Output Matrix

After introducing the context diagram for the software and explaining in detail the requirements (see Section 4.2), a clear Input-Output table was created.



Inputs	Outputs
• Thruster geometry; • Type of each surface (conductive or insulated) • Applied magnetic field strength and shape; • Mass flow rate; • Power; • Regulated (user or DT changing) discharge voltage; • Initial electric field strength; • Fixed or floating electrode potentials; • Type of propellant; • Type of material; • Initial propellant temperature; • Initial thruster surfaces temperature; • Injection propellant velocity; • Injection propellant density; • Propellant ionization levels; • Propellant mass; • Size of propellant atoms; • Size of surface material atoms; • Work functions of surface materials; • Emission parameter of the surface materials; • Energy sputtering yields of the materials; • Sputtering yield of the materials; • Cooling power of the anode; • Operation mode (steady, quasi-steady, pulsed);	• Total thrust; • Each component of the thrust; • Electric field topology; • Induced magnetic field topology; • Hall current (formula available); • Hall parameter (formula available); • Plasma parameter (formula available); • Radial discharge current (formula available); • Axial discharge current; • Thermionic current (formula available); • Density of neutrals throughout the thruster; • Density of electrons throughout the thruster; • Density of each ionisation level ions throughout the thruster; • Temperature of neutrals throughout the thruster; • Temperature of electrons throughout the thruster; • Temperature of each ionisation level ions throughout the thruster; • Velocity of neutrals throughout the thruster • Velocity of electrons throughout the thruster; • Velocity of each ionisation level ions throughout the thruster; • Average of each velocity components of neutrals at multiple time intervals ; • Average of each velocity components of electrons at multiple time intervals; • Average of each velocity components of each ionisation level ions at multiple time intervals ; • Anode Fall Voltage; • Cathode Fall voltage; • Anode losses; • Cathode losses; • Losses from neutrals exiting the thruster; • Level of backtracking of electrons; • Average of the ionisation level of the propellant; • Electrical conductivity -> magnetic Reynolds number;

Figure 4.2: All the inputs provided and the outputs expected for a plasma simulation software for NSS.

This table represents a more detailed set of parameters than what was used in the requirements. This way a more detailed picture of the software is painted.

4.4.2. Sequence Diagram

To better understand the string of actions that are required in a plasma simulation software that is accepted, a Sequence Diagram was designed (see Figure 4.3).

However, there is a limited amount of information that can be put on a diagram that does not become too confusing. As such, a proper explanation is given here. The start is given by the inputs from the user. In the first stages (before DT integration) the user would put the inputs manually, while the magnetic topology would be introduced as an (unchanging) input as well.

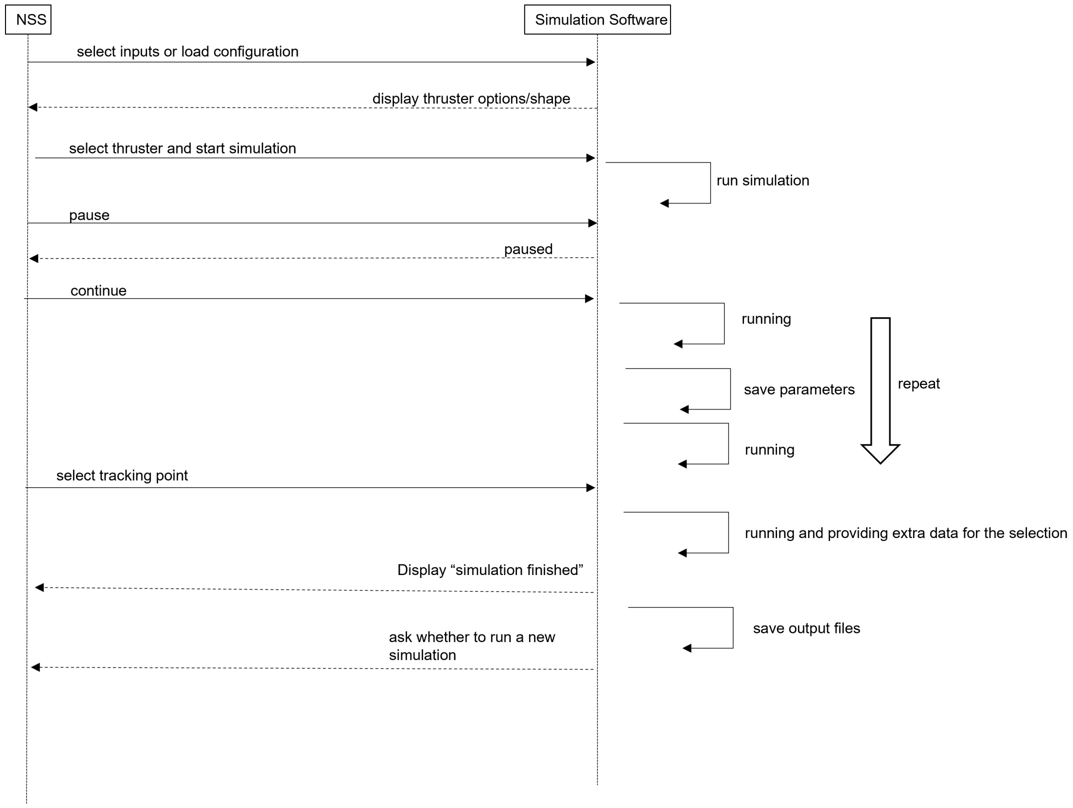

NSS
Simulation Software
select inputs or load configuration
display thruster options/shape
select thruster and start simulation
run simulation
pause
paused
continue
running
save parameters
repeat
running
select tracking point
running and providing extra data for the selection
Display “simulation finished”
save output files
ask whether to run a new simulation

Figure 4.3: The string of actions within the function of the plasma simulation software.

After the verification of the software and it’s validation before DT integration the integration itself would be considered. This would be followed by, naturally, by verification and validation of the integrated software product. After these successful procedures are conducted, the software shall be able to use inputs directly from other parts of the digital twin, so the entire SUPREME thrust simulation is automated. This is where the user would also have the option of loading a previously saved simulation at a certain point in time or a load configuration (to slightly tweak the inputs). Then, the user would be able to choose the thruster geometry and its corresponding characteristics (materials, properties etc.) as well as selecting tracking points or regions on the geometry of the thruster. Furthermore, according to software developers, the use of fully 3D or 2D axi-symmetric geometry in the simulation depends on the solver used by the company. While first iterations of the code can be used with 2D axi-symmetry, for an increased fidelity, 3D is preferred. However, changing the dimensions require a change in the solver, a very complex feat. As such, it might be more feasible to use a 3D solver from the first iteration. Tracking options (particle(s) or geometry) would also be available at each point in time of the simulation, not just at the beginning.

After all these aspects are decided and introduced, the user has to press the start button for the simulation to process the inputs and conduct the simulation. In addition, according to three type of regions identified (see Literature Study report [23]: discharge, confinement, divergence), at the end of each stage there would be an automatic saving procedure.

As can be seen in Figure 4.3, the user can pause and then continue the simulation at any point in time. Furthermore, there would be the option of selecting tracking geometry and tracking particles when pausing and the user would also be able to save the simulation. As previously explained, the software automatically saves the data at the end of each of the three steps, but if there is any need of stopping the simulation now and continue at a different time, the user would be able to save at any point as well.

As explained in the theory, it is faster to load the results of a simulation than rerun that desired part of the simulation. It is also cheaper as running HPC units has a high cost and consumes considerable energy (more sustainable to load files). In the name of sustainability, in later software iterations the user shall be able to pause or load, change some of the inputs and continue the simulation with the new inputs. This can be done to become a more sustainable company and when the change of some inputs does not have a strong impact on the simulation up to that point.

Very important to be mentioned, this report has no requirement on the type of solver used or the order of field computations. While the order of field computations is somewhat similar between companies, each software company has their preference when coding for the simulation. As the software product would have both PIC and DSMC module, the simulation would move the particles according to PIC and conduct the interaction according to the DSMC module. As put in Figure 4.3 the steps would be repeated until the program ends. Most likely, the steps would be linked to the distance travelled by an electron. It should be noted that a finer simulation, with more steps and smaller time steps would result in a longer simulation, while too few steps would result in a simulation that cannot be resolved.

After selecting the tracking particles or geometries, the software would provide additional data corresponding to the user's selections, at the end of each stage in the output files.

At the end of the simulation, before the integration, the software would provide the output files in the formats explained in the requirements. These would include detailed information of the simulation and a ring or circular sheet of charge, around the thruster, corresponding to the backtracking electrons. The charged particles would not be simulated for their entire backtracking journey, just the charge levels around the thruster, right after being ejected. At the end, the software would ask if the user wants to run a new simulation. After the software is integrated with the DT, the outputs would be automatically used/processed by the next parts of the DT framework.

4.5. Additional SE Aspects

As defined in Section 4.1, the four methods would be used to validate any chosen plasma simulation software. It is very important to be mentioned that the actual validation stage was not tackled in this report due to the lack of a chosen software product and its adaptation for the needs of NSS in the time frame of the thesis.

In the first method, the end-to-end information system testing, the software shall be validated by using artificial data as inputs. The artificial data represents the inputs that a thruster would have. Thus, functions of the software would be added and checked with each addition, ending with the test of all the functions as a whole. If there are no errors and the output files have accepted output (but not correct since not real data would be used) then it can be said that the software was correctly validated.

The second method, mission scenario test, the software shall be validated by using new experimental values. As detailed in Section 4.3, the verification methods are conducted using data gathered by NSS from previous experimental tests, and with different operating parameters and geometries. In the case of validation, the results would be compared with new experimental tests. If the simulation results pass the requirements when compared with the new experimental data, then it can be said that the results were validated.

Validation can also be thought in terms of operation readiness tests. Applied in the context of the plasma simulation software, this would result in validating its integrating in the digital twin framework. This is again, tested against the new experimental data. The integration process would be done in a bottom-up approach: the smallest, most basic functions would be integrated within larger, more complex functions. In addition, integration should be done in a concurrent manner. This would not be entirely suitable for a small subsystem of the entire system, where functions are done mostly sequentially. However, the integration process of all the components of the digital twin can be and shall be done in a concurrent manner, but this aspect is out of the scope of this report.

The last validation method, stress-testing and simulation is not as relevant as the previous three methods in the case of the simulation software. This is because the code was already tested in different conditions and with different data and parameters in multiple stages. What can be brought in addition, however, is test data again, but with the purpose of analysing where the code would fail: the time steps are too small - the grid is too fine, the discharge voltage or current are too high or too low, erosion is too high etc.

Furthermore, one of the reasons for not using part of the previous experimental data, but new for validation, lies in the capacity of getting much more tailored measurements from the new data, for the software.

Very important to be mentioned, validation can only be done on tests from an AF-MPD thruster with the same geometrical design and the same operating parameters as the SUPREME thruster. This is what was meant in this section by ”new experimental tests”. Verification can also be done with new experimental tests, but that do not have the SUPREME geometry or its operating parameters. It would not be possible to use old test data from SUPREME to validate the software because there is no built SUPREME thruster yet.

4.5.1. Fault Mitigation And Costs

This section relies more on understanding the definitions of aspects such as validation, which are going to be implemented when implementing a simulation code. However, this section is more attuned to the case of a start-up such as NSS. Looking at such an example, regular meetings at certain intervals (will be discussed shortly) are based on the discussion with a certain option with regards to consultation needs.

While verification looked mostly at checking if the software fulfils a certain requirement, the validation methods would test fault tolerance implementations of the software. Additional tests to prevent faults, errors and failures would result not only in a reduced cost down the line, but most importantly in a faster time to market, a vital aspect for start-ups. While most of the tests would have to be conducted in-house by the developing company, some tests can also be approached by NSS. One such example is testing operating parameters limits,introducing tracker particles or regions at each point in time for the simulations and a consistent behaviour of initial testing/initialisation with simple inputs and comparison with the expected outputs.

As previously said, system costs have to be seen as a design variable. Especially in the context of a start-up, there are limited funds available. Given the novelty of an AF-MPDT plasma simulation software, bottom-up (analytic - use the costs of each subcomponent and build up) and parametric models (using historical data and mathematical relationships) are not suitable for cost approximation. However, the analogy-based cost model (using past similar missions) might provide a starting point. While the AF-MPDT simulation code would be novel, there are similarities in the code with other electric propulsion technologies. Thus, a very rough approximation can be made by NSS.

However, official discussions are needed as every software company has different knowledge and would have different costs based on the knowledge needed for SUPREME simulation. Furthermore, once a company is chosen, the costs shall be settled for the first two iterations at least, maintenance and support, additional needs and training, as stated in the requirements. This would ensure that the developing company can approximate how many employees it can hire and the required time, while NSS can have costs almost fixed, which might be ideal in a limited funds scenario.

In addition, it can be very useful for NSS and the chosen company to have regular meetings every 2-3 weeks to discuss the software progress. Given its novel nature, it might be very likely there will be time delays and perhaps additional costs. Having regular meetings would ensure these time changes are accounted for and that the additional costs can either be covered with additional time or costs cuts are needed from other departments. For example, if additional funds are needed in the plasma simulation department, it can be a great idea at a specific time, to take funds from manufacturing or other departments. Analysing the outputs of the simulations can give investors a further motivation to use their capital in NSS in a way that having the parts produced (that can have wrong materials or geometries) might not.

As such, the development of different subsystems can be done concurrently, but good contact with the development of each subsystem can lead to efficient cost distribution and time delay preparations.

5 Simulation Options

The last part of the thesis was split into two components. The first component pertained the differences and similarities between three of the stakeholders that can provide a plasma simulation code (as obtained from interviews with the respective stakeholders), while the second part dealt with basic simulations of just plasma sheets in 2D/3D environments. This meant that only the PIC approach was used, with no collisions (no DSMC) and only electrons and ions as particles. This was a much simpler simulation than an AF-MPD simulation, which has more types of particles and more complex phenomenons. Put simply, the cases used would represent only a part of the full AF-MPDT simulation. However, this initial testing stage was used to provide initial understanding on the potential of an open-source code.

5.1. Description Of Main Options

The three options presented are denoted as option S, option B and option J. The reason for not mentioning their names is to ensure that there are no copyright issues. Companies would tend to prefer a more positive description based on their marketing texts. By covering their names it is ensured that their reputation is not influenced, for example if there are certain functions not implemented yet. Openly putting their names would also require the signing of contracts that restrain what can be discussed.

Option J is a single developer which has experience with simulating plasma in Hall thrusters and is currently conducting a PhD in the field of plasma simulation. Option B is a small start-up that has some experience in plasma simulations for electric propulsion thrusters, works on the basis of consultancy with open-source codes and now focuses on other plasma applications. Lastly, option S is a young company larger and older than B, with specialised people for many areas. However, according to their schedule, they are to release their first working version of the code, that is focused on Hall Effect thrusters and ion thrusters in June of 2022. They work on the basis of software licenses.

A few requirements would be selected that would be used for an initial evaluation of the products, after a more detailed analysis of these stakeholders.

The code from J is designed to be more modular, with approaches such as PIC, MCC seen as modules. Thus, changes, even those on a more fundamental level, do not require that additional modules are updated to fit the changes, but just the main file. The current development of the approaches includes PIC, while DSMC is in progress. MHD approach is also in plan, however no exact time frame is settled. MHD is not really of relevance for this software, but is mentioned for completeness. The meshes developed are in 2D, rectangular and symmetric cylinder. For the second mesh, the processes simulated must not break the symmetry, so care has to be given to parameters such as the azimuthal current. However, a 3D mesh is in the development.

With regards to the particles used, electrons, protons and multiple ionised ions as well as a range of neutrals are already implemented. Although the work currently conducted with this software is used for simulations of the space environment (mainly due to the Sun), the software can easily scale from a density of 10^9 to 10^{18} - 10^{20} particles. As previously said, DSMC is in development, so none of the interactions stated in the requirements (NS-AM-FUN-6-1, NS-AM-FUN-6-2, NS-AM-FUN-6-3, NS-AM-FUN-6-4, NS-AM-FUN-6-5) are implemented. A time frame suggested for these are four months from the start of the implementation, which would also include relevant tests to provide a complete product.

The software can receive as input the applied magnetic field, the electric field applied is self consistent and can update in time (affected by the interaction with the charged particles). Previous developments would allow the self-induced magnetic field to be simulated, which would be done with a starter grid (induced magnetic field grid) to conserve energy. In addition, the code also allows that at the border of the simulation, protons and electrons can disappear or be created. J software has also an additional function that is in the fine tuning stage: a particle reformer. This class allows for an automatic change of the mesh, such that there is a balance between precision and time. This does not affect the physics of the simulation and the mesh regions have finer meshes as well.

While erosion is not implemented, it can be included in those four months of total development time needed. However, a geometry change that would change the processes might require even more time. The easiest approach suggested was the use of an erosion formula in the post-processing stage.

The only losses accounted at the moment are losses from particles getting erased when going out of the simulation range. In addition, the software can also process the heat flux from the plasma which is needed in order to know the cooling power required for the anode.

The order of movements for the code is given by the speed of the electrons. Thus, at time 0 all particles move a step, then for a number of time steps only electrons move, then the electrons and the protons. This trend continues for another set of steps until all three, the electrons, protons and ions move at once again. A very important part of the simulation is the discharge, which is a difficult process to develop and has not been implemented in the code from J. However, a very important aspect is that the code from option J already satisfies requirement NS-UI-DES-1 as unlike for option S, this software can continue the simulation where it left off.

Furthermore, implementing additional processes would extend the total time required to 6-8 months for option J. It should also be considered that it is likely that the capability at the end of this time frame would still be lower than that of the other two options at the end of their required time.

Lastly, simulating electron backtracking is possible but not implemented yet. This would also involve changing the weight of the superparticles.

Option B was a university spin-off for three years and is a start-up since 2021. This comes with strong collaboration between them and the University of Stuttgart, which are also researching the AF-MPDTs and can result in better physics models and tests. However, unlike the other two options, they are transitioning away from space propulsion technologies to vacuum plasma flows systems. The code from B is open-source and services are paid on the basis of consultancy. A small addition must be made, while most of the code is made open-source, not all details, especially the newer implementations such as the discharge are public.

They have previously simulated Hall Effect thrusters, ion thrusters and have AF-MPD knowledge from their work with and at Stuttgart University. Further emphasis is put on the active collaboration between option B and Stuttgart University, and one such recent success is the development of a natural way of simulating the discharge in an AF-MPD thruster. On the lines of their collaboration with the university, they use master's or PhD students to develop certain parts of their simulation software for certain applications. No such available students are present, so a higher cost would be charged by them in order to hire specialised personnel if NSS desires to have further developed or tailored to their specific needs, code.

The code developed by B can either be used as a hybrid code (lose all electron dynamics) or with a fully kinetic baseline. The second option means they are resolving the Debby length frequency and it is computationally intensive. This further spawns two options: they either use assumptions to decrease the computational power and time required or use an HPC. Currently, the code developed by B works only on the HPC cluster from Stuttgart University and additional time is required to be adapted to Azure by Microsoft or AWS by Amazon (although it is expected that this adaptation procedure not to be long, but no exact time frame was given).

The software uses a 3D mesh and a limited (up to 90°) 2D mesh is also possible. Their software uses a 3D solver, thus, switching to 2D is a slow procedure. There are a number of species that can be simulated and examples such as atomic N, atomic O, their product NO, 5 species air model, methane and CO_2 have already been implemented. With the exception of complex molecules, a wide range of particles can be simulated. Chemical reactions are also supported and tests are conducted by B, such as aluminium ionisation tests, every week. Their software can implement particle ionisation with one of three methods: Arrhenius equation [5], quantum kinetic model using DSMC (ionisation level threshold) and cross-sections, which can vary depending on the particle type and energy. For the last option, the reaction is energy-dependent for each pair and an energy table can be used to simplify the simulation, as there would be different cross-sections for each reaction type. Then, an interpolation method can be used to find the exact cross-section.

The software from B has also already implemented the five interactions (NS-AM-PER-6-1, NS-AM-PER-6-2, NS-AM-PER-6-3, NS-AM-PER-6-4, NS-AM-PER-6-5) with rotational, vibrational and electronic types for the excitation interactions. The charged-charged particles interactions (Coulomb) have not been implemented and their implementation would add additional six months for the code development.

Their software has a different approach to the simulation of induced magnetic field: they would refine the time steps to such an extent that the simulation has a resolution suitable for the speed of light. This is, of course, a very computer intensive approach. Thus, their code would simulate the four main forces.

The code can receive as inputs the applied magnetic field and the electric field, which updates in time (interaction with charged particles). The plasma sheaths corresponding to the electrodes can be simulated, but stakeholder B plans to do more research for the anode fall voltage to provide a more accurate simulation. Unlike the code from J, the grid (coarse or fine) has to be set manually, but there are plans to automate this in the next iterations.

Erosion was not implemented, but similarly as for the code from J, an erosion rate formula can be used on the output data, so no changes to the geometry while the simulation is running. The losses can be implemented only from simple models that can account for particles getting absorbed. In addition, the code would be able to compute during the simulation the heat flux to the anode.

Unlike the software from J, the approach of B with regards to the backtracking particles is different. Instead of using the same code with a different superparticles weight, a second code was suggested, which is a very simplified version of the main code from B (no collisions as well) and that would take as inputs the charged particles from the output and would compute the distribution of the backward going particles. This method would result in the main code having a limited domain, as the second, simpler code would take the exit domain (space behind thruster).

Option S is a start-up with a few years of experience and that unlike option B, uses only its employees (many with PhD studies) to develop products. Their main focus is on Hall Effect thrusters and ion gridded thrusters, however, if an agreement is reached, focus can be diverted to an AF-MPDT tailored code. The code from S also has fully kinetic capabilities, with 2D and 3D solvers and hybrid capability with 3D a solver. The 2D mesh is axisymmetric and is usually used in HET devices to speed up the simulation, but comes with the downside of not being able to reproduce the erosion patterns.

In terms of hardware they are using, there are two types, each with two levels of computing power. Unlike the other two options, the code from S would have the GPU parallelization capability which would greatly reduce the simulation time. It is not clear whether this option would be available with the official code in June of 2022 or shortly after. Thus, they have a CPU only execution hardware or both CPU and GPU execution, with each having minimum requirements or recommended hardware. This should be taken into consideration in the total costs, as it might increase the cost of cloud computing or in-house hardware.

The code can simulate all five processes, including double ionisation. However, in terms of particles, they can simulate everything except for complex molecules. Like the other two options, the software can take as inputs the applied magnetic field and the electric field which updates itself according to the interaction with the charged particles. With regards to the self-induced magnetic field, there were two options suggested. Either use a 10^{-12} seconds time step to resolve the speed of light and simulate the B-field or use a proprietary method of simulating the induced B-field before the rest of the simulation. No additional details can be provided as it might infringe on their intellectual property rights. The code from S has also implemented the discharge, however, in an nonphysical way. The anode fall voltage is expected to be simulated accurately as the company has experience with simulating plasma sheaths. Also, detailed voltage output results can be shown. In addition, the software can treat the electrodes as either floating or fixed potential.

Similarly with the other two options, the erosion rate is using a formula and it does not consider changing geometry during the simulation. However, in addition to the other options, the code considers the impact of the eroded particles in the plume. With regards to the losses, the software can compute the unidirectional heat flux to the electrodes. Furthermore, it can compute at the electrode's surface the electrons absorbed, ions scattered, energy and momentum transfer and the thermionic electron emission. With the exception of electrons absorbed, the other mechanisms are already considered by the other two options in the interactions implemented. However, ion implementation is not covered and nor is the mid to long term sticking of particles, that create a coating.

On the subject of backtracking particles, both option B and option S have the same approach: use a second code that uses as inflow the outflow of the main code. Again, this is a much simpler version so additional time, costs and other resources are expected to be negligible in comparison to the main code.

In Figure 5.1 the traits of the three options are summarised.

5.1.1. Cost Of Options

As explained previously, option B has an open-source code on GitHub and it is monetising it's functioning with paid consulting services. It should also be stated that not all functionalities, such as the newer or more complex developments are available as open-source. Furthermore, it was stressed by B that a couple of months are needed for a team to be able to understand and use the code from B for their applications.

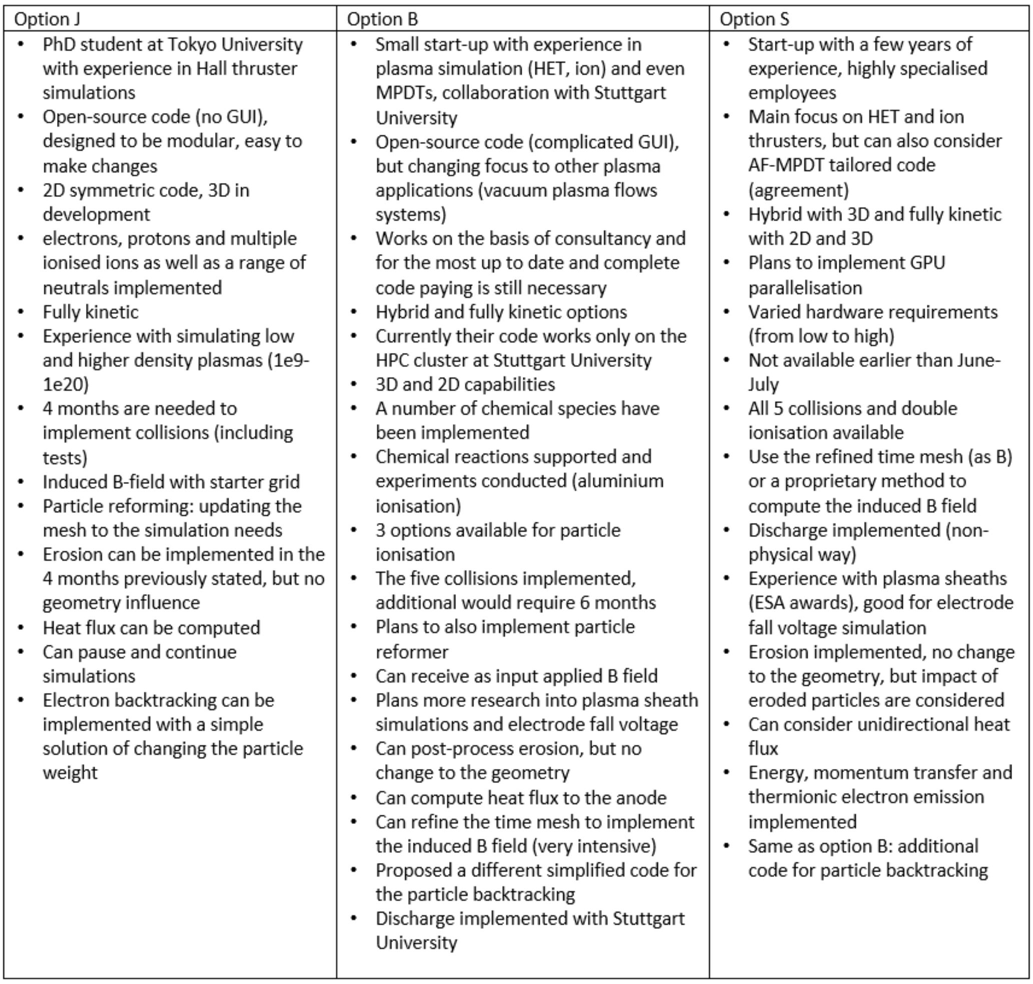
Option B offers two paid options: provide support and provide support with training. The first option consists of replacing the individual work of tailoring their code for the clients' business, as B would adapt their code with all their developments. This adapted code would be given as it is and it would be up to the clients to learn how to use it, how to obtain additional outputs etc. With the second option of both support and training, B would also allocate time with its consultants and developers to train the personnel of their clients. As an example, an offered made by them to NSS for half a day of workshop/-training and 10 hours of support was for €5000 (euros net).

Option S on the other hand, works with licenses. There are no open-source codes available and the interaction with them is governed by the signing of an NDA (non-disclosure agreement). Thus, it is expected that charging a one-time fee (additional fees are possible but it is expected to require much fewer resources than the main license) that would be much larger than a few training and support offers from B. Furthermore, more time resources would also be necessary. Since option S is currently developing the codes and did not have any interaction with AF-MPD thrusters, a number of meetings might be required to set in detail all the requirements, the time frames and the framework of the code.

In addition, option S not having a collaboration with an institute/university research AF-MPDTs similar to option B with Stuttgart University, NSS would have to make sure that all the theoretical information about AF-MPDTs is shared with them.

Both options B and S recommended a collaboration through EU or ESA funding. Applying for research funding from these institutions has the potential of allowing NSS to continue its operations and without having to give shares from the company or having investor expectations.

On the other hand, option J would have an open-source code that he can develop during his PhD.



Option J	Option B	Option S
• PhD student at Tokyo University with experience in Hall thruster simulations • Open-source code (no GUI), designed to be modular, easy to make changes • 2D symmetric code, 3D in development • electrons, protons and multiple ionised ions as well as a range of neutrals implemented • Fully kinetic • Experience with simulating low and higher density plasmas (1e9-1e20) • 4 months are needed to implement collisions (including tests) • Induced B-field with starter grid • Particle reforming: updating the mesh to the simulation needs • Erosion can be implemented in the 4 months previously stated, but no geometry influence • Heat flux can be computed • Can pause and continue simulations • Electron backtracking can be implemented with a simple solution of changing the particle weight	• Small start-up with experience in plasma simulation (HET, ion) and even MPDTs, collaboration with Stuttgart University • Open-source code (complicated GUI), but changing focus to other plasma applications (vacuum plasma flows systems) • Works on the basis of consultancy and for the most up to date and complete code paying is still necessary • Hybrid and fully kinetic options • Currently their code works only on the HPC cluster at Stuttgart University • 3D and 2D capabilities • A number of chemical species have been implemented • Chemical reactions supported and experiments conducted (aluminium ionisation) • 3 options available for particle ionisation • The five collisions implemented, additional would require 6 months • Plans to also implement particle reformer • Can receive as input applied B field • Plans more research into plasma sheath simulations and electrode fall voltage • Can post-process erosion, but no change to the geometry • Can compute heat flux to the anode • Can refine the time mesh to implement the induced B field (very intensive) • Proposed a different simplified code for the particle backtracking • Discharge implemented with Stuttgart University	• Start-up with a few years of experience, highly specialised employees • Main focus on HET and ion thrusters, but can also consider AF-MPDT tailored code (agreement) • Hybrid with 3D and fully kinetic with 2D and 3D • Plans to implement GPU parallelisation • Varied hardware requirements (from low to high) • Not available earlier than June-July • All 5 collisions and double ionisation available • Use the refined time mesh (as B) or a proprietary method to compute the induced B field • Discharge implemented (non-physical way) • Experience with plasma sheaths (ESA awards), good for electrode fall voltage simulation • Erosion implemented, no change to the geometry, but impact of eroded particles are considered • Can consider unidirectional heat flux • Energy, momentum transfer and thermionic electron emission implemented • Same as option B: additional code for particle backtracking

Figure 5.1: The table summarising the three options.

While the code and outputs can be used by NSS at any time and with no monetary requirements, the intellectual rights remain to J. In addition, after the PhD is finished, further support for the code would require NSS to suggest a job offer. In addition, option J may seem quite advantageous if NSS plans to have more time to raise funds, while still acquiring knowledge. In addition to the much lower costs of option J, having an employee with specialised knowledge can ensure a better retention and distribution of knowledge in the company. In addition, having someone with experience would ensure that if NSS receives funds and signs a contract with a company, there would be less consulting sessions required by NSS from the chosen company.

It is very important to be mentioned that while all three options have a lot of potential, options B and S employ many more people for their projects and have more experience in the field of electric thrusters.

5.1.2. Options Trade-Off

The trade-off criteria resulted from the requirements in Sections 2.1 and 4.2 as well as from interviews with the NSS leadership to understand better the needs for the code. Furthermore, the criteria are not weighted and a Graphical Trade-off table is chosen. The reason for this lies in the different needs NSS can have at each point in time. Thus, this report can have a more useful result by explaining the reasons behind the most suitable choice at each given moment in time. The criteria chosen are:

1. Time for development;
2. Cost of achieving a fully operational software;
3. Hardware required.

The time required for an operational software development was chosen as the first criteria as the three options have a range difference of a few months between them. This is a very important aspect as a code that is ready sooner can provide data faster and might result in obtaining funding earlier. It can also result in more inside knowledge until funding is achieved or until a contract is signed with a customer.

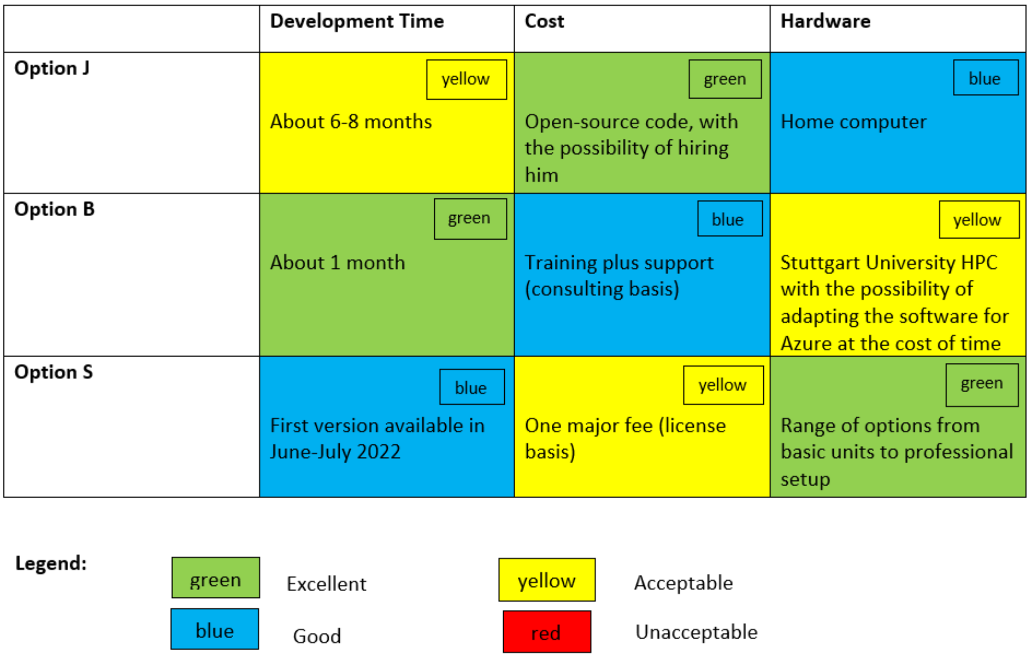
The second criteria, the cost, plays a significant role, especially for younger start-ups such as NSS. With limited funding, the cost can have a great impact on the financial planning of the company, on what projects NSS accepts and how resources are moved between sub-departments.

The third criteria can act as a constraint on the availability of running the software at all or in a timely manner. In addition, more complex options might allow yielding simulation results faster.

The theoretical background or theoretical knowledge achieved by each option is not considered in the trade-off because none of the 3 options has extended experience or knowledge in the field of AF-MPDTs and this type of thruster is still under theoretical analysis at institutes and laboratories around the world. In addition, the main functionalities needed are covered to a great extent by the requirements explained in this document and they can be seen as a part of the required time to implement a functioning version of the software. Theoretical models would be provided by NSS as new research is conducted, either in-house or in collaboration with institutes and universities such as Stuttgart University. Time required for each option to conduct a full simulation was not considered, although, it is partly included in the Hardware criteria. Also, after a successful operating software is achieved, future iterations can take this into consideration.

Based on the three criteria discussed in Section 5.1.2, the Graphical Trade-off table can be seen in Figure 5.2.

As can be seen, in the time section, only option B has the *Excellent* classification as their software can be adapted in about 1 month (although a few features would still need more time to be implemented). Thus, this represents the fastest option of getting closer to optimisation and then prototype manufacturing. The other two options have decreasing grades: option S have the classification *Good*, as they would require a few months longer, while option J is slightly worse with *Acceptable*, requiring 6-8 months. Furthermore, given their current status, it is probable that they would require additional time for implementing all or most of the functionalities covered by the requirements in this document. Option J has only one developer and it is uncertain how long can potential delays be, especially when



	Development Time	Cost	Hardware
Option J	yellow About 6-8 months	green Open-source code, with the possibility of hiring him	blue Home computer
Option B	green About 1 month	blue Training plus support (consulting basis)	yellow Stuttgart University HPC with the possibility of adapting the software for Azure at the cost of time
Option S	blue First version available in June-July 2022	yellow One major fee (license basis)	green Range of options from basic units to professional setup

Legend:

green	Excellent	yellow	Acceptable
blue	Good	red	Unacceptable

Figure 5.2: Graphical Trade-off table for the three options.

implementing more complex requirements. The given time frame was accepted in consultation with option J. Option S on the other hand, while it has a large team, their focus is on HET and ion thrusters and any product launch has a chance of being delayed. Considering that Option B works in collaboration with Stuttgart University, they are ahead in terms of AF-MPDTs processes testing.

The next criteria is cost. Obviously, option J has the *Excellent* classification as the highest cost predicted is the salary if NSS hires him. Then comes option B with *Good* classification, as the cost can be spread into multiple tranches and it can be expected to have initial training at the beginning only. About support, this is expected to be required at the beginning as well, with rarer meetings later if a new process needs to be implemented. Option S provides an *Acceptable* only option, as they require a large sum for the license. However, this is expected to cover the full development of the initial stage of the product, including testing and a basic training for NSS to use the software. No complex training is needed as option S would provide a fully usable GUI.

Regarding the hardware, option S has the widest range. Thus, their software can run on a basic and inexpensive hardware if simulation time is less important than reducing costs or it can run on a professional hardware is simulation time is the most important. This gives a range of options that is suitable for a start-up with the possibility of upgrading depending on the funding. In addition, the configurations mentioned might be available on cloud computing options, which would reduce upfront cost for NSS. The next option, with a *Good* classification, is option J, which can run on a home unit. This reduces costs dramatically. However, this also carries very long simulation times. A simple simulation of the plasma sheath, with just electrons and protons takes about 30 minutes. Thus, lacking the range of options seen for S. Finally, option B has the worst hardware options as it can currently run only on the Stuttgart University HPC centre. With little time however, it might run on Azure as well. This would fit in the one month time frame corresponding to option B.

A few important aspects should also be discussed. It was mentioned earlier that theoretical knowledge is somewhat similar with all three options. This is not entirely accurate, but not very relevant to the trade-off process. Currently, option B has the most knowledge, followed by option S and then option J. With this situation, it can be seen that option J can develop a code faster than option S, although the last has a team working on the projects. It should be considered however, that option S is focused on HET and ion thrusters. In addition, it is expected that options B and S conduct much more rigorous, complex and resource intensive tests than option J, given their larger plasma simulation experience, larger teams and larger funding. Thus, accuracy is expected to be larger for options B and S. Given the diversity of professions working for option S and the difference in focus (option B focus moves to general plasma applications, especially on Earth activities, while option S focuses on electric propulsion simulations), it is expected that on a larger time frame, option S would provide more accurate and stable software packages.

The choice of an option lies on the interests of NSS. Thus, if funding (considering unlimited funding) is available and the need for a software is not urgent, i.e. the first complete software can be ready later than June, then the obvious choice is option S. This is also supported by the fact that on a longer time frame, option S is expected to provide much more reliable and accurate software.

However, if limited funding is available (can pay in tranches/consultations) and time is of the essence, option B is the right choice. It should be noted that most of the implementation of new processes might be conducted within master and PhD thesis. However, they currently have the necessary knowledge and skills of developing an accurate code that covers almost all processes.

If no funding is available and time is not a pressing issue, the right option might be option J. This would provide NSS with a working code in a few months time, as well as develop on plasma simulation knowledge within the company.

However, a better option at the moment might be to go with option B for all the benefits mentioned earlier, but also hire/use option J to decrease the time for the training of NSS personnel to use the software from B. Furthermore, also using option J should decrease the frequency of consultation sessions with option B, as he should be able to take care of small processes that appear later or less complex errors that might appear. This way would also ensure a higher retention of knowledge within NSS (associative learning).

It is further emphasised that an accurate and robust code, especially for an electric thruster with a low TRL requires considerate resources such as time of at least a few months, considerable funding mostly for developers, physicists, needed for a large and skilled team. Also, funding is needed for hardware, that is most of the time specialised hardware.

5.2. Plasma Sheath Simulations

For this section of the report, a simple, collisionless plasma sheath simulation would be analysed, from the option J. In addition, comparisons with the same simulation test case from option S were added in this section. However, given that option S is a 3D simulation with more advanced code practices and a GUI, the simulation with this software was not achieved in less than 12 hours (with a standard home laptop with a 9th generation Intel i7-core). Furthermore, on option S it is not possible to pause the simulation, further impairing the achievement of results with it. For completeness, a brief explanation of the simulation test case with the professional option S was provided in this report.

The two simulations were set-up following the same set of test case characteristics. The 3D geometry of the test case is a rectangular cuboid with the two opposite faces along the length of the cuboid being the inflow and the outflow regions. In addition, the outflow region is a conductive wall. Looking at the cuboid perpendicular to the length, the left wall is the inflow region, while the right wall is the outflow region. However, as would be explained, the geometry used was in 2D. Thus, the cuboid simplifies to a rectangle, where the left side is the inflow wall maintained at 0 V and the right side is the direction in which the ions move, being the conductive wall with negative potential.

In this case, the dimensionality of the simulation (1D, 2D, 3D) is not important and given the time constraints, unlike the code from S which is in 3D, the code from J is in 2D. This case provides a geometric symmetry and together with the desired quantities to be measured and plotted in this report, there is no need for more than 1D simulations. Thus, the types of plots chosen would be taken along a line (at height = 0), in the middle of the width of the bottom rectangle and along the length of the cuboid, from one wall (inflow) to the other (outflow). It is expected that the quantities chosen to be plotted vary depending on the location along the width and along the height, but this does not provide useful information for this thesis. Considering that simulations can have numerical issues (such as for option J) and there are local effects, the parameters are not the same to the directions perpendicular to the main path of motion (straight line perpendicular to the inflow and outflow walls). This is in contrast with the fact that there is the same number of particles inflowed at each time step from the entire inflow wall. As such, the average is interesting for this report. Considering that the simulations done are in 2D, it is better to have the average value along the width at each point in the length of the domain. So, an advantage of a higher dimensionality is that it would allow the user to see how the plasma inside the cuboid can vary in different regions (i.e. different species densities). However, higher dimensionality also requires more processing power.

First, the size of the rectangle, respectively the rectangular cuboid were set-up in term of the Debye length (it represents the size of the electrostatic effect of charge carrier in plasma or conductors, after this distance the charge is screened [25]). The values used here is $\lambda_D = 3.0862 \cdot 10^{-4}$ m, as the plasma sheath is treated as a whole in this example and not as part of the MPD as discussed in Section 3.2. Thus, the length of the mesh (between the inflow/outflow walls) was set-up to be $50\,\lambda_D$, while the width and height were $5\,\lambda_D$. Thus, the mesh from the J code has a size of $50{\cdot}5$ (while the mesh from the S code had a size of $50 \cdot 5 \cdot 5$) - each side was multiplied by the Debye length. It should also be stated that a minimum advisable length between the two walls was $30\,\lambda_D$, otherwise there would be imbalances in the plasma. At smaller lengths, the inflow of particles is very close to the outflow and the electrons can be repelled directly. Thus, leading to a charge imbalance. Also, the number of diagnostic points to have an accurate sampling was set to be larger than 50 (the length of the path - multipliers of λ_D), but it differs between the two options. The difference between the numbers of diagnostic points comes from the flexibility of analysing, changing and most importantly, running the code from option J within the previously mentioned deadline. Thus, the number of points was personalised such that there was not an overflow of data that might slow the code, but that it provided distributions that seemed continuous and not value jumps from large differences. This was implemented on more qualitative grounds and it did not follow an exact quantitative value.

The walls have a potential difference between them: inflow wall has a fixed potential at 0 V, while the outflow (conductive) wall has a floating potential of -10 V. Floating means that charged particles are being neutralised and absorbed by the conductive wall and this wall changes the potential to a higher value (in this case) that corresponds to the steady-state of the system. Thus, this negative potential difference is what makes ions move to the conductive wall (this is in addition to the other velocities that cause ions to move in the direction of the conductive wall). With regards to the other (side) walls, they are treated with a Neumann boundary condition of 0 V/m gradient of the potential. The software from option S uses periodic boundary conditions (PBC) in its simulations, not only for this test case, but for full thruster simulations. This results in particles colliding to the side walls and getting absorbed, followed by being injected back in the domain from the opposite wall. On the other hand, in the case of option J, the particles simply bounce back from the collision wall. Thus, using PBC allows for a smoother simulation.

The time discretisation consists of time steps each of $8.86 \cdot 10^{-11}$ seconds. The number of steps was increased until the simulation reached the steady-state. No plots were provided as they would have high variations before reaching steady-state. Any system first starts with the transient state, which would see fluctuations in quantities such as the potential and the electric field. With time, the steady-state is reached, and the potential and the electric field become stable with time.

As previously said, the inflow wall is on the left side and the density of particles coming from it is 10^{14} particles/m^3 per time step (for each species, electrons and ions). It should be noted however, that due to the applied potential and by solving the Poisson equation the distribution of particles does not remain constant through time and along the domain. The particles are given an initial temperature that stays constant throughout the simulation. Thus, electrons were set to have $T_e = 2000$ K (or about 0.17eV as

1 K = $8.617 \cdot 10^{-5}$ eV [11]), while ions were set to have $T_i = 0$ K, to not affect the Bohm criterion (more would be explained later). The electron temperature controls the Debye length according to:

$$\lambda_D = \sqrt{\frac{\epsilon_0 k_B T_e}{e^2 n_e}} = 3.0862 \cdot 10^{-4}, \tag{5.1}$$

where ϵ_0 is the vacuum permittivity, k_B is the Boltzmann's constant, e is the charge of the electron (in Coulomb) and n_e is the electron density.

This velocity cause the Maxwell-Boltzmann velocity distribution. In addition, the electron velocity is used for computing the Bohm velocity (criterion) of the ions, corresponding to the pre-sheath region, as explained in the Literature Study [23]:

$$v_B = \sqrt{\frac{k_B T_e}{m_i}}, \tag{5.2}$$

where k_B is Boltzmann's constant, the temperature of the electrons was used (T_e), and the mass of the ions is m_i. Thus, the total velocity of the electrons come from the Maxwell-Boltzmann velocity distribution and from the interaction with the potential. The ions however, their total velocity is given by the Maxwell-Boltzmann velocity distribution, the potential and a drift velocity chosen by the user. The choice of drift velocity would be explained later. According to the Bohm sheath criterion [22], the magnitude of the total total ion velocity has to be larger than the Bohm velocity for a collisionless sheath.

There is another option possible when setting up this test case: empty simulation domain or full simulation domain. This refers to the number of particles that are already in the domain before the simulation starts. For option S, this option was not available in their GUI, for this test case. This was motivated by the fact that in a standard plasma sheath, for example in a thruster, there are no particles initially in the domain.

However, whether the simulation domain is filled with particles or is empty, it would not affect the physics. The advantage of using a filled domain lies in the simulation reaching the steady-state faster.

Furthermore, a technique previously explained, used to make the simulation faster, is using superparticles. However, option S provided a formula to compute the maximum theoretical value for the particle weight such that there is a value of about 100 superparticles in the volume of a Debye sphere:

$$weight = \frac{V_\lambda \cdot \rho}{100} = \frac{\frac{4}{3} \cdot \pi \cdot (7.4325 \cdot 10^{-4})^3 \cdot 10^{14}}{100} = \approx 123, \tag{5.3}$$

where V_λ is the volume of the Debye sphere and ρ is the density of a species. As such, for option J, a particle weight of 100 for each superparticle was set, in order to have more accurate results. When the particle weight increases, accuracy of results decreases. However, for option S it was accepted to have a larger particle weight (200) in order to decrease the large simulation time, with the accepted issue of worse accuracy than using a lower weight. More details about the choice of particle weight are provided later.

The theoretical result that should be achieved from these simulations can be seen in Figure 5.3, where a collisionless plasma is moving towards the conductive wall.

The continuity in this collisionless environment holds so the product of particle density and their respective average velocity is constant ($n_1 v_1 = n_2 v_2$). There are three assumptions [22] that were employed: electrons have a Maxwellian distribution with T_e, ions are cold (temperature of ions is much smaller than that of electrons and was considered zero as previously stated) and at the inflow region the densities of the two species are equal (10^{14} particles/m^3). In Figure 5.3, n_e, n_i and n_0 refer to the electron, ion

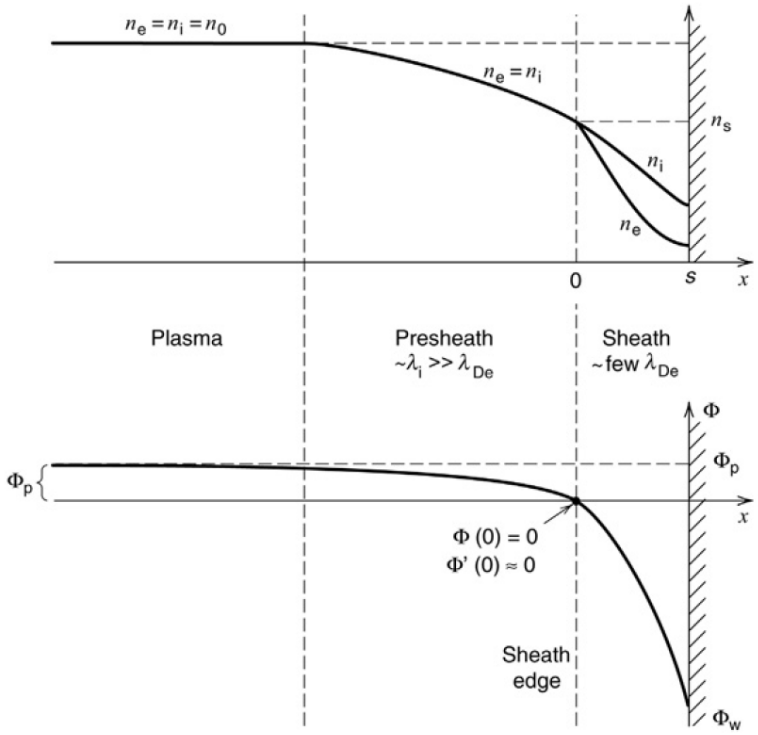


Figure 5.3: Theoretical behaviour of a collisionless plasma, sheath and pre-sheath in contact with a conductive wall [22]. The pre-sheath is quasi-neutral, having equal electron and ion densities. However, the sheath is non-neutral. There is a negative potential reached by the wall ($-\phi_w$), as the ions fall on it (ions have a reaction time larger than of electrons). Electrons with an energy lower than $e\phi_w$ are reflected. It can also be seen that there is a potential drop in the pre-sheath region, causing the ions to accelerate to a terminal velocity known as Bohm velocity [25].

and initial particle densities, while the index "s" refers to the sheath region. Here, the Debye length is noted as λ_{De} and the wavelength corresponding to the ions by λ_i. The potential is denoted as Φ, with the index "p" referring to the positive potential and the index "w" to the minimum potential reached at equilibrium. In the test case described earlier, the inflow wall has a fixed potential, thus, the exact shape of the plot should be moved down so that the potential starts at 0 V. The floating potential of the outflow wall would accumulate charges and would result in a different end potential (Φ_w) than at the start (-10 V).

The species used in the example from Figure 5.3 were electrons and an ionised hydrogen atom [22]. However, since protons/hydrogen atoms were not implemented in option S, the protons were changed with single ionised helium atoms for both options. Thus, the Bohm velocity computed with Equation 5.2 had to be divided by 2, as the mass of a helium atom is considered to be exactly the mass of 4 protons. All the other parameters mentioned in this section remained the same.

5.2.1. Option J

Option J used Python programming for the entire code and has no GUI. Thus, running the code was done through a Python terminal like Anaconda [3] and the data was analysed and plotted using the visualisation ParaView software [32]. Running the software on a home laptop with a 9th generation Intel i7-core requires slightly more than 10 hours. Running the simulations on a more powerful home laptop (newer processor, better cooling) it can take the software even 5 hours and a half. It was expected that the code J would run faster than code S as there is no GUI that also requires processing power and the simulation domain was in 2D, unlike code S which was in 3D.

The code was updated a number of times with the most important problem being the species densities, especially at the entrance of the domain. After the inflow was solved to be constant along the inflow wall, to solve the issue of putting much fewer particles than 10^{14} particles/m^3, there was an additional pre-entrance box created. This box allowed the creation of a higher number of particles and allowed the simulation to get closer to the 10^{14} value injected.

There are four types of plots that were used as they provided the best way to analyse the quality of the simulations:

1. Potential and electric field in the outflow wall ($x = 50\lambda_D$) varying with time;
2. Potential and electric field along the path between inflow and outflow walls ($x = 0$ and $x = 50\lambda_D$), averaged over the steady-state;
3. Density of each species along the path between inflow and outflow walls ($x = 0$ and $x = 50\lambda_D$), averaged over the steady-state;
4. Particle current/flow in the outflow wall ($x = 50\lambda_D$) varying with time.

For all these plots, the diagnostic line was taken at height = 0, from left (inflow wall, $x = 0$ m) to the right (outlfow wall, $x = 50 \cdot \lambda_D = 0.01543088$ m) and in the middle of the width ($x = 5 \cdot \lambda_D = 0.001543088$ m), as the domain is symmetric. Taking the diagnostic data at height = 0 ensures a better comparison with other 3D cases. Thus, as the electric field is a vector, if the path of particles (length of domain) is considered as the x-axis, then only the x-direction of the electric field vector was plotted.

As the ions are helium atoms, the Bohm velocity changed accordingly (two times slower than having hydrogen atoms as the ions) to the value of 2031.55 m/s, using Equation 5.2. The total ion velocity was then computed by summing the velocity due to the potential, the Maxwel-Boltzmann velocity (if $T_i \neq 0$ K) and the ion drift velocity, which for a reason that would be explained later, this is known as the $x1$ case.

In the next sections, there are two cases analysed. The first is the example case, with a higher particle temperatures, corresponding closer to an AF-MPDT, while the second case has lower temperatures. However, given that the length of the domain in the example case which was not larger than 30 Debye lengths, its results cannot be treated as interpretable as it is expected that local issue may arise in the plasma. For the second case (the test case), the domain length was larger than 30 Debye lengths and

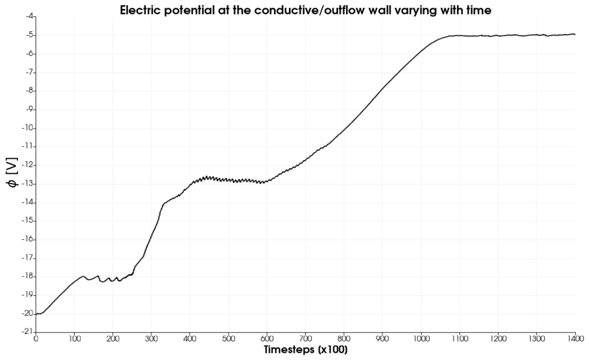


Figure 5.4: The electric potential at the outflow wall, varying through time with ion drift velocity = Bohm velocity ($x1$).

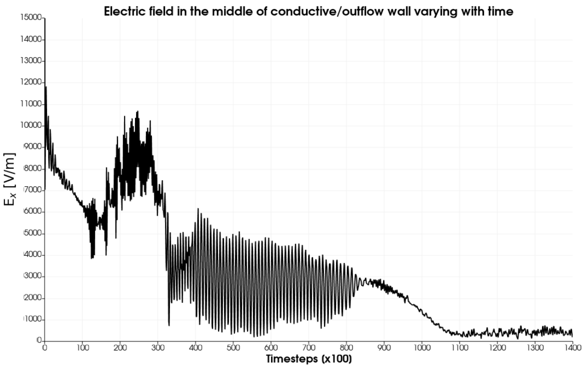


Figure 5.5: The electric field at the outflow wall, varying through time with ion drift velocity = Bohm velocity ($x1$).

represents the comparison scenario against the theoretical case.

5.2.2. Example Case For J

At first, there was an example test case that had values closer to a real case and that was provided by option S. As such, the electron temperature was $T_e = 11600$ K ($\sim 1eV$), the ions has a temperature, although very small: $T_i = 200$ K, which created the Maxwell-Boltzmann distribution for ions as well. The higher electron velocity created a higher Debye length ($\lambda_D = 7.4325 \cdot 10^{-4}$ m, according to Equation 5.2). However, the simulation domain size was kept the same at a slightly larger size of 0.01848 m x 0.00154 m (compared with 0.01543088 m x 0.001543088 m). T_e also caused a Bohm velocity ($x1$) of 4892.62 m/s. The particle weight was kept the same at 100 particles/superparticle, the voltage at the outflow wall was -20 V. The number of steps was 140000 steps, but this was enough to reach steady-state. It should be noted that this case was done in order to understand to better set up a test case and to achieve steady-state faster. However, given that the length of the domain used was 25 times larger than the Debye length in this case, it does not provide much real case interpretation. As explained in a previous section, the length of the domain has to be larger than 30 Debye lengths.

Looking at the potential and the electric field ($E_x = -{}^{\Delta\Phi}/_{\Delta x} = {}^{\Phi_{n+1} - \Phi_{n-1}}/_{x_{n+1} - x_{n-1}}$, where n is the number of the nodes) at the outflow wall, varying through time, it can be seen in Figures 5.4 and 5.5 that the steady state is reached after almost 1100 time step sets. The x-axis in these plots is in sets of 100 time steps, each step lasting $8.86 \cdot 10^{-11}$ seconds as mentioned in Section 5.2. Thus, steady-state is reached after about 107000 time steps, with the potential at the outflow wall starting at -20 V as in the setup and stabilising at -5 V, while the electric field stabilised around the value of 500 V/m. A filled domain of 10^{14} particles/m^3 density was used in the case only in order to achieve faster steady-state.

Two regions with more variations in the potential can be seen (Figure 5.4): between 100 and about 250 time sets (each time set is a 100 time steps) and between and between 400 and 750 time sets. This translates into large variations to the electric field plot (Figure 5.5 in the same regions, with a longer cool down period (it takes longer for the electric field to become more stable compared with the potential). It should also be noted that when a region in a plot is said to be more stable or unstable, it is in comparison with the rest of the plot. This longer cool down period of the electric field might be explained by its differential computation, which takes into consideration the potential at the current location and that of the previous location.

After making sure the steady-state is reached, the next plots were analysed, such as the electric potential and the electric field, throughout the length of the domain and at steady state. Throughout this report, when it is said that a parameter is plotted at steady-state, it means that the final value is achieved by averaging (at each location) all the values after the steady-state is reached. Thus, all the values (at each location along the path) from time step 10700 until time step 140000 are used to compute the average at their corresponding locations.

In Figure 5.6 the potential along the path can be seen, from the inflow region at $x = 0$ m up to the

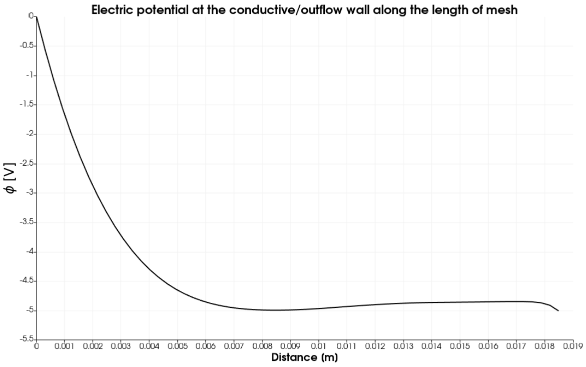


Figure 5.6: The electric potential along the path between the walls, at steady state, with ion drift velocity = Bohm velocity $(x1)$.

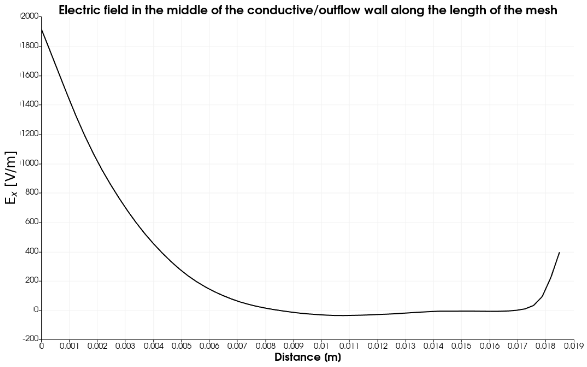


Figure 5.7: The electric field along the path between the walls, at steady state, with with ion drift velocity = Bohm velocity $(x1)$.

outflow wall at $x = 0.01848$ m. There is a steep decrease, followed by a mild increase and another final decrease to the same lowest level of -5 V. The potential at the inflow wall stays at 0 V as it was set as fixed in the code, while shortly after (from 0.007 m onward) the potential wiggles around the final, stable value. Looking at the electric field (along the length of the domain, x-direction) from Figure 5.7 it can be seen that up to 0.017 m the electric field follows the same shape as the electric potential. In the middle of the domain, for the $x1$ case, the electric field reaches the value zero as the potential is constant in this region. However, towards the end, as the potential saw a sharp decrease towards the end and the electric potential is positive in this case, it resulted a sharp increase close to the outflow wall in Figure 5.7.

The movement of particles affect the potential and the electric field, thus, it is important to look at their densities at steady-state, between the inflow/outflow walls. While the code was set to inflow 10^{14} particles/m^3 throughout the simulation, in Figure 5.8 it can be seen that at the entrance the ions start with a lower density, followed by a sharp increase right after. The electrons, on the other hand, star with a slightly higher density that also peaks at the same region as the ion's density. This might have been due to the Maxwell-Boltzmann distribution, but more investigation is needed for a conclusion.

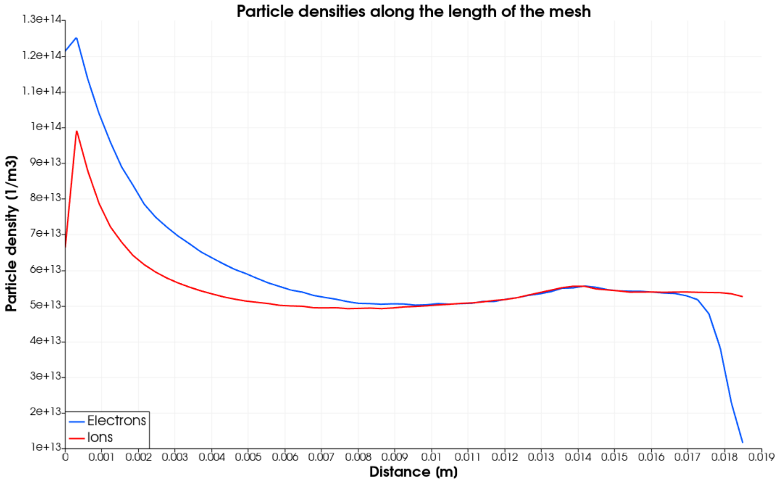


Figure 5.8: Densities of the two particle species (electrons and ions) along the path between the walls, at steady state, with ion drift velocity = Bohm velocity $(x1)$.

As a general rule of thumb, the higher electron density would be expected closer to the inflow as the potential of the outflow conductive wall is negative. In the middle region, the densities of the two species become almost the same, which would also explain the constant potential seen in Figure 5.6 in the middle of the domain. At the end of the domain, the electron density decreases sharply, as expected, while the ion density remains constant, while the start of a decrease might be seen.

However, comparing Figure 5.3 with Figures 5.6 and 5.8 it can be noticed that beside the second half of the electron density plot, none of them follow the theoretical shapes.

To analyse all the information, the flow of particles to the outflow wall was plotted as current against time. Again, the x-axis of Figure 5.9 represents time sets, each of 100 time steps. The current could be computed similarly to the electric field: $I = {}^{\Delta Q}/_{\Delta t} = {}^{(Q_n - Q_{n-1})}/_{(t_n - t_{n-1})}$, where Q is the total number of charges at the respective step, t is the time step value and n is the time step number.

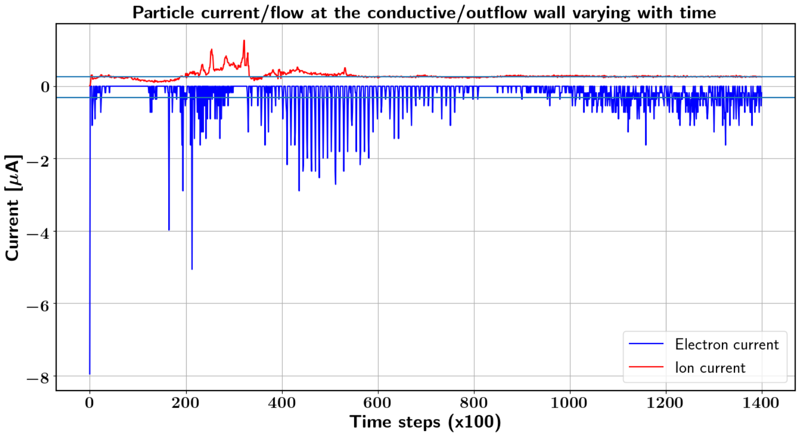


Figure 5.9: Flow of charged particles/current to the outflow wall, varying through time with ion drift velocity = Bohm velocity $(x1)$.

In the beginning of the simulation, the electron current/flow is seen having large variations for very brief time periods compared with the rest of its shape. In contrast, the ion current/flow to the conductive wall has a large value compared with the rest of the ion current plot and for a larger duration than the electron peaks: between 200 and 400 time sets. A possible explanation for this is that the electrons have a high temperature, so there is a considerable number of electrons in the higher velocity distribution tail. Together with the fluctuations in the potential in the transient state, a large number of short peaks in the electron flow to the conductive wall might be explained. However, when looking at the region after the steady-state and average the electron and the ion currents (bleu line), it can be seen that these two averages are at the same absolute values. This was expected as the system was negatively charged (imbalance of the species) and at the steady-state the currents sum to zero.

So, while it was checked that the steady-state was correctly reached with the help of Figure 5.9, Figures 5.6 and 5.8 are far from the expected theoretical shapes seen in Figure 5.3.

The plots shown (Figures 5.4 - 5.9) are simulated with a particle weight of 100 particles per superparticle. However, for faster simulation times the particle weight can be higher in this case. The only change resulted from this was that with a smaller particle weight the current peaks (Figure 5.9) can be seen more clearly/sharply. The maximum theoretical value for the particle weight such that there is a value of about 100 superparticles in the volume of a Debye sphere (using Equation 5.3): weight = 1719. Thus, a value of up to 1719 particles per superparticle was recommended, much larger than the value used.

There can be multiple problems that cause the deviation from the theoretical expectations, however, steady-state was reached. This can be seen in the constant potential and electric field from Figures 5.4 and 5.5 as well as the equal electrons-ions current from Figure 5.9.

Problems may have incurred from the domain length (0.01848 m) being about 25 times the Debye length ($\lambda_D = 7.4325 \cdot 10^{-4}$ m). What was noticed to have the greatest impact to achieving the theoretical shapes, was the change in drift velocity of the ions. As explained in Section 5.2, for this collisionless sheath, the magnitude of ion velocity has to be greater than the Bohm velocity. However, due to the Maxwell-Boltzmann distribution, the velocity of the ions could vary both in magnitude and in direction.

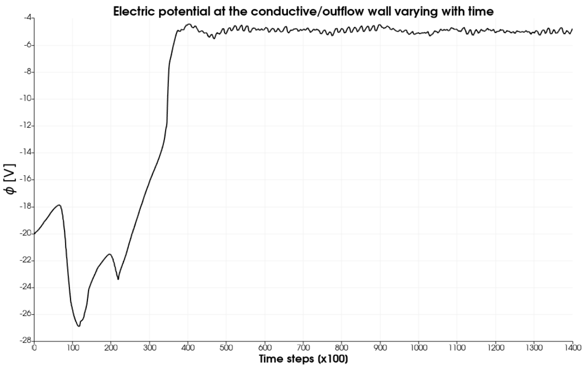


Figure 5.10: The electric potential at the outflow wall, varying through time with ion drift velocity = 6 x Bohm velocity ($x6$).

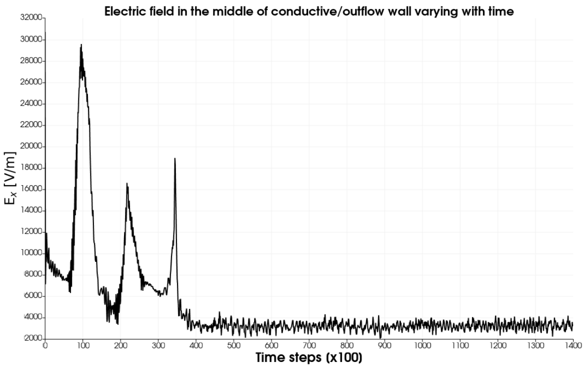


Figure 5.11: The electric field at the outflow wall, varying through time with ion drift velocity = 6 x Bohm velocity ($x6$).

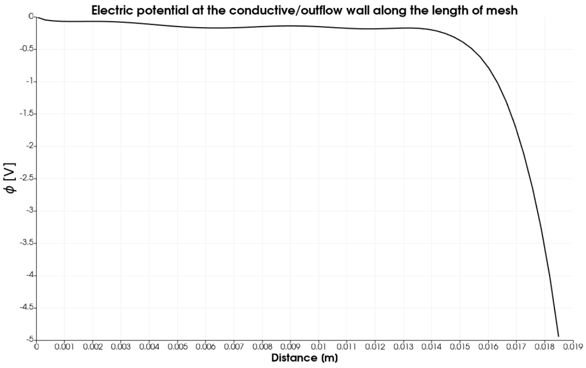


Figure 5.12: The electric potential along the path between the walls, at steady state, with ion drift velocity = 6 x Bohm velocity ($x6$).

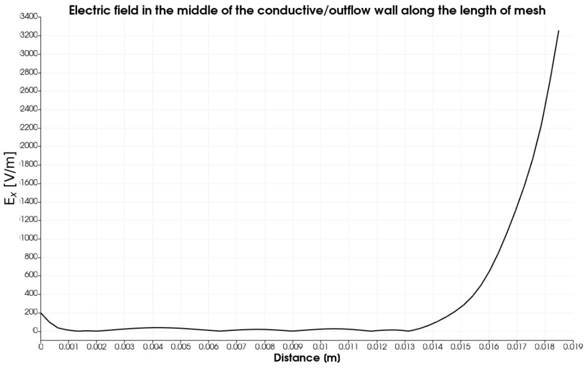


Figure 5.13: The electric field along the path between the walls, at steady state, with with ion drift velocity = 6 x Bohm velocity ($x6$).

So, a larger ion temperature can give an ion a higher velocity towards the outflow wall, decreasing the time to pass the domain or it can give a lower velocity, increasing the domain passage time. In addition, it was noticed that the ion velocity had to be much larger than the Bohm velocity to reach the theoretical expectations. Trying higher multiples of the Bohm velocity, it was noticed that at an ion drift velocity of six times the Bohm velocity the outcome is close to the expected results and the plots start having a similar shape to those in Figure 5.3. Higher ion velocity compared to Bohm velocity results in the Bohm criterion being overstepped more, thus, providing the expected results (the theoretical case). Analysing from a real case point of view, the ion velocity ranges in the interval 0 - 12 eV, so in the cases from this report, considering either $x1$ or $x6$ values, both are within the interval. The same type of plots as for the $x1$ case were plotted for the $x6$ case.

Figures 5.12 and 5.14 show that with the $x6$ case the drift velocity of ions was large enough and a smooth and accurate steady-state was reached. Thus, the theoretical case seen in Figure 5.3 was almost mirrored in this case. The only difference with the desired shape came at the outflow wall, where the electrons density saw a very sudden drop, while the ion densities remained almost unchanged. It should be noted that the ion drift velocity accounts not only for the $x6$ Bohm velocity, but also the potential and the Maxwell-Boltzmann contributions.

Furthermore, it can be seen that the steady-state is reached faster, at 40000 steps, much earlier than the 10700 steps required for the lower ion drift velocity.

5.2.3. Test Case For J

As seen in the previous chapter, using a high enough ion velocity would ensure that the theoretical case from Figure 5.3 is reached. However, having the length of the simulation domain less than 30 Debye lengths could cast doubt on the validity of the results due to imbalances in the plasma. A possible explanation for the results close to the theoretical ones might come from the high ion drift velocity: fast

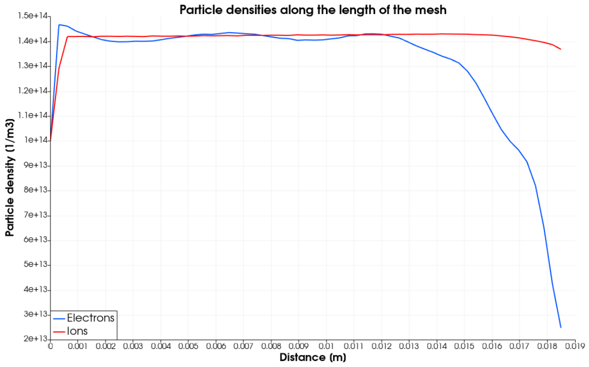


Figure 5.14: Densities of the two particle species (electrons and ions) along the path between the walls, at steady state, with ion drift velocity = 6 x Bohm velocity ($x6$).

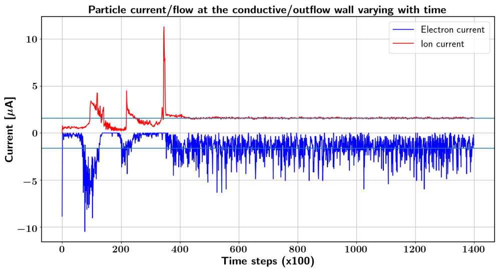


Figure 5.15: Flow of charged particles/current to the outflow wall, varying through time with ion drift velocity = 6 x Bohm velocity ($x6$).

enough ions might result in a smooth stream, even in a small domain.

In agreement with option S, the test case had a few changes compared with the example case. They were mentioned previously in Section 5.2, but were restated here in more detail. Firstly, the ion temperature was reduced to 0 K such that there is no Maxwell-Boltzmann impact on the ion drift velocity and the second assumption (out of the three mentioned in Section 5.2) is fulfilled. The temperature of the electrons was lowered to 2000 K. This resulted in a smaller Debye length of $\lambda_D = 3.0862 \cdot 10^{-4}$ m according to Equation 5.1. This resulted in the domain being larger than 30 Debye length multipliers, but for a slightly shorter time required, while not getting too close to the 30 lengths minimum, the round value of the domain length of 50 λ_D was chosen (0.01543088 m). Time step size, density and weight of the superparticles were kept the same. Given Equation 5.3 the weight had to be smaller than 123 (condition fulfilled), but not make the simulation take longer.

A potential value further away from 0 V resulted in more time needed for the simulation as it would take longer to reach steady-state. while a more negative potential would influence ions to have a higher velocity towards the outflow, the constant potential value could be reached much faster if the value of the outflow wall was changed from -20 V to -10 V. From multiple simulations with the rest of the parameters fixed, it was noticed that a less negative potential (closer to 0) reaches steady-state faster, decreasing the time needed for the simulation. This was in contrast with the less negative potential that would increase the ion drift velocity less than with -20 V.

With the assumed negative impact on the time required, the simulation domain was set as empty. Thus, there were no particles throughout the domain, at the beginning of the simulation, requiring more time to reach steady-state. Thinking of a thruster in space that has no initial particles, it made the test case more relevant. The initial case of using $x1$ Bohm velocity (initial/multiplied by 1, $v_B = 2031.55$ m/s) as contribution to the ion drift velocity provided results far from the theoretical expectations. However, a value of $x6$ Bohm velocity (multiplied by 6, $v_B = 12189.3$ m/s) ensures correct results. This might mean that in the example case with smaller than 30 λ_D domain length it was not the plasma instabilities that was causing problems, but the ion drift velocity being larger than the Bohm criterion. Considering the slightly smaller domain, but the slower particles, the total number of steps implemented was 450000, which explains the longer simulation time required.

Again, the potential varying with time and the corresponding electric field were plotted and can be seen in Figures 5.16 and 5.17.

The first aspect to be noticed in comparison with the example cases (both $x1$ and $x6$) are the increase of variations throughout the simulation. Between the example cases the increase throughout the simulation came from higher velocity. It was not fully understood why in the test case plots there are larger variations after steady-state is reached as the velocity from Figures 5.16 and 5.17 is smaller than in Figures 5.10 and 5.11. An explanation for the higher variations before steady-state, in the ex-

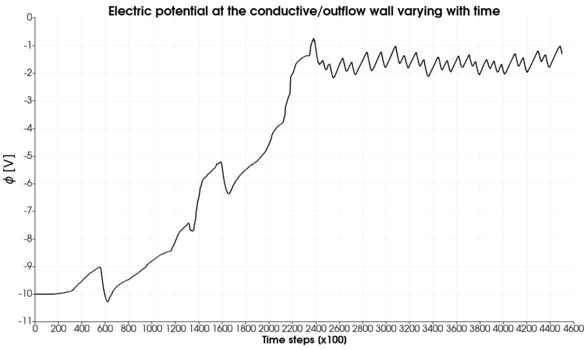


Figure 5.16: The electric potential at the outflow wall, varying through time with ion drift velocity = 6 x Bohm velocity ($x6$) - empty domain.

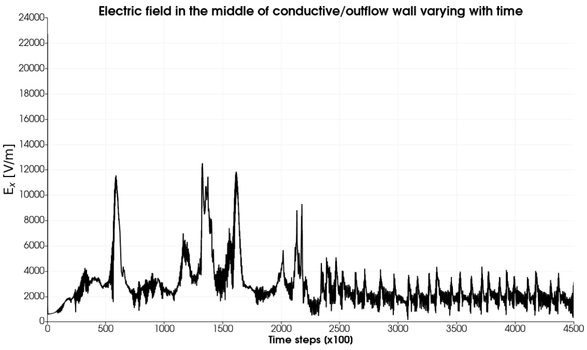


Figure 5.17: The electric field at the outflow wall, varying through time with ion drift velocity = 6 x Bohm velocity ($x6$) - empty domain.

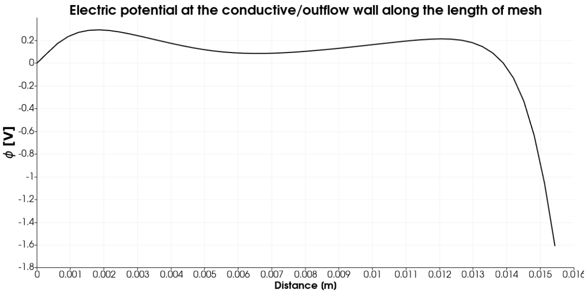


Figure 5.18: The electric potential along the path between the walls, at steady state, with ion drift velocity = 6 x Bohm velocity ($x6$) - empty domain.

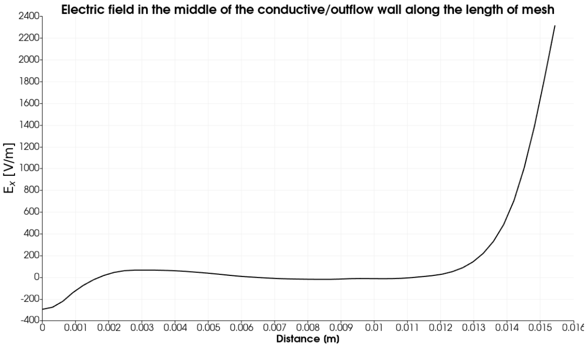


Figure 5.19: The electric field along the path between the walls, at steady state, with with ion drift velocity = 6 x Bohm velocity ($x6$) - empty domain.

ample case, might come from the instabilities seen in a small domain compared with the Debye length. However, but Figures 5.16 and 5.17 show that steady-state is reached from step cycle 2400 (240000 time steps). The larger variations at steady-state represent a further reason to test individual aspects from a thruster plasma simulation and understand how they would impact the performance of a thruster.

The potential throughout the domain (after steady-state is reached) and the corresponding electric field can be seen in Figures 5.18 and 5.19. These parameters are too different from the $x1$ example case, but they are very similar to the parameters from the $x6$ example case. Thus, the ion drift velocity played a significant role and it can be safely said that the $x6$ test case simulation results follow the theoretical expectations seen in Figure 5.3.

However, at the beginning of the domain (close to the inflow wall) a bump in the potential towards higher values can be observed. This results in the corresponding electric field starting from negative values. Since these two plots are taken as an average of the values at steady-state, it cannot be said that starting the simulations with an empty field played a role in this discrepancy. Looking at the densities throughout the domain (at steady-state) helped in understanding the bump.

To understand the accuracy of Figure 5.20, an understanding of Figure 5.14 is necessary. The $x6$ example case has achieved such a close shape to the theoretical model artificially. Due to numerical aspects of the code, the simulation was forced to add a higher density of particles such that there is a 10^{14} inflow density. This stabilised the densities, but also made them quickly rise above the set value, although particles are also absorbed at the outflow region. This buildup of density can be seen in Figure 5.20 of the test case, but it is delayed as there were no additional particles inflowed to keep the density to a desired value. Thus, a possible explanation for the bump is that at steady-state the much faster electrons go ahead, leaving an excess of ions at the beginning, creating the corresponding bump. In the middle of the domain, the electron density caught up and even overtook the ion density.

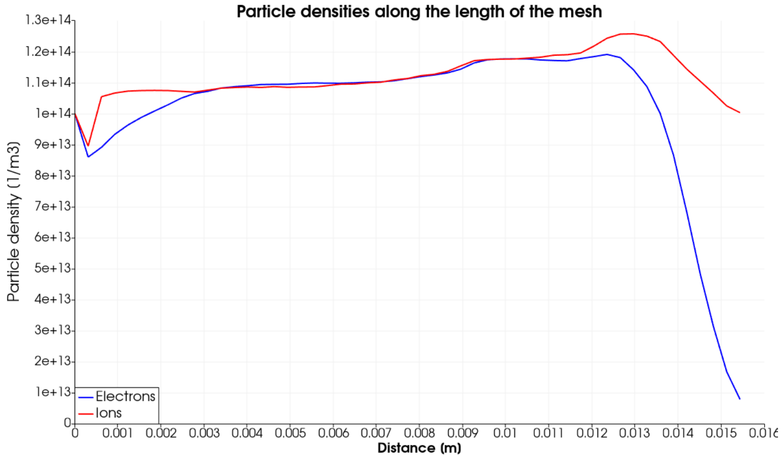


Figure 5.20: Densities of the two particle species (electrons and ions) along the path between the walls, at steady state, with ion drift velocity = 6 x Bohm velocity ($x6$) - empty domain.

The highest particles densities is reached at the end of the domain, right before the outflow wall. It can be seen that the ion density has a small decrease back to the initial value of 10^{14} particles/m^3, while the electron density drops suddenly to very low values. However, the higher value in electron density in the second half of the plot may be explained by the presence of ions which shield the electrons from the negative potential of the outflow wall.

While Figures 5.18 and 5.20 are not exactly following Figure 5.3, it can be said that they have similar shapes and the expected results are achieved. It should be further emphasized that similarly to the explanation in Section 3.7.4, at higher thrust (ions closer to the outflow wall) the density should decrease towards the exit (outflow wall).

To further support this, the current flow was plotted at the outflow wall (see Figure 5.21). Again, it can be seen that the averages of the ion and electron currents (bleu line) were the same, indicating that the steady-state was reached.

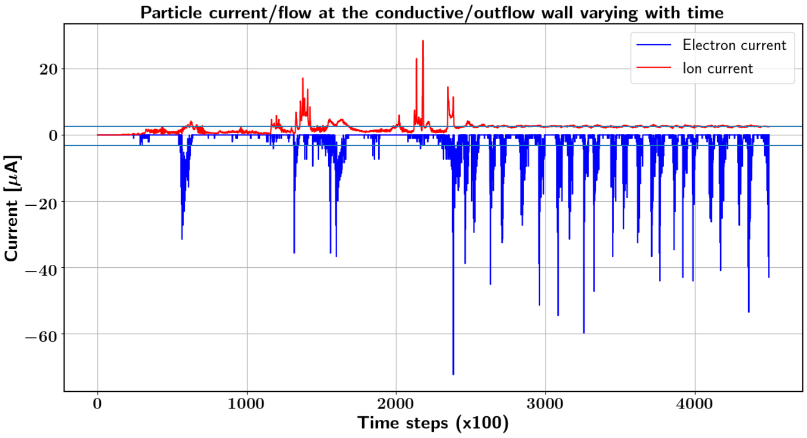


Figure 5.21: Flow of charged particles/current to the outflow wall, varying through time with ion drift velocity = 6 x Bohm velocity ($x6$) - empty domain.

These current averages are about the same at for the $x6$ example case. However, the more seldom, but much higher peaks may indicate that the particle weight is not as high compared to the corresponding case as for the example case. Both simulations used 100 particles per superparticle, but since the maximum weights computed with Equation 5.3 were 1719 (example case), respectively, 123 (test case). As the value of 100 is very close to the maximum value in the test case, it can be understood that the particle currents are not as smooth in Figure 5.21 as for the other cases.

Analysing the parameters of the J simulation with the test case yielded results close to the theoretical values expected. This pointed to the fact that the J code has potential in being used either as a standalone that is heavily tested at each stage and different cases and as an add-on to other codes. However, these simulations emphasise a very important aspect in plasma simulations that was only briefly touched by the requirements definition seen in Section 4.1: each component of the thruster plasma simulation should be developed individually and then integrated together while testing at each integration. It was clear even from the first simulations in this project that extended tests and verification of the quality of results are a necessity.

Furthermore, these simulations and the corresponding analysis should not be taken as a final product or as a prototype that can fulfil the requirements explained in this report. The simulation cases focused only on the plasma sheath, which is one of the most important parts of an AF-MPD thruster, but it was still only briefly tested and as an individual part. Further work would see more lower level requirements that would go into more detail on individual aspects of the plasma such as the plasma sheath. Also, having a full simulation software that can be verified and validated against the requirements was not feasible in this report, nor would it be feasible to ask multiple stakeholders to produce such a code and then decide which option NSS would choose. Even a basic AF-MPDT plasma simulation requires significant resources so a full software comparison was not feasible as a solution proposition.

In addition, there were some numerical problems that have to be analysed into more detail and that affect the density of the particles right after injection into the domain and the shape of the densities throughout the domain. If option J would be chosen, then multiple test cases have to be conducted and understand where the numerical issues arise from.

By conducting the example case simulation it was clear that the code from option J has the potential to conduct the basic plasma sheath simulations, even though the results are not relevant given the length of the domain. On the other hand, the test case simulation respected the domain length being greater than 30 Debye lengths, providing interpretable results. However, the test case simulation took much longer. Furthermore, it is useful to see how the results change from case to case. Another important result from the example case simulation was that a larger ion velocity was needed for the simulations, so the $x6$ ion value was implemented over the $x1$ value.

Thus, it can be said that the main research question (*What are the requirements of a plasma simulation software to describe as accurately as possible the acceleration mechanisms within a state of the art AF-MPDT such as the SUPREME thruster?*) was answered to a good degree. In addition, by finding and interacting with the stakeholders of each of the three options, the requirements gained further relevance from their knowledge. These requirements are appropriate for the first stage of a simulation software and there has to be work conducted with the chosen stakeholders in order to further develop them.

6 Conclusions

The main objective of the thesis project was to find suitable options of a plasma simulation software for the SUPREME thruster, based on research, available resources at NSS and interviews with the corresponding stakeholders. This section delves into the conclusions that support the main aim.

The literature analysis conducted in Section 3 yielded a number of conclusions that would be considered in a plasma simulation code. On the topic of electrode fall voltage, while there are losses in both electrodes, it is clear that anode losses are larger and thus, more important than the cathode losses. Together with the steady-state operation of SUPREME, the thermionic current should be considered and the impact of the heat flux, quantified. The software should cover the use of Argon as a propellant first, followed by complex molecules in later iterations. When conducting experimental tests, if sputtering is shown to play a large effect nevertheless, the use of coatings should be employed. The relationship between thrust and other operating parameters in the simulation should be compared with the current models for an AF-MPDT. Important information can result by computing the the radial discharge current, followed by determining the electric field and the magnetic Reynolds number.

The type of simulation has also provided important insight. Thus, the need of PIC-MCC simulations over the classical MHD resulted from two aspects: the nonlocal dynamic effects and the plasma transport coefficient. The plasma coefficient has gained increased interest in the field as the transport of energy in the plasma can lead to the understanding of a number of phenomenons in a thruster. The use of hybrid PIC over fully kinetic PIC was shown to not be a good approximation as the three needed approximations are not respected. While a fully kinetic approach would require more time, a few methods can be used to improve on this aspect, such as: the use of superparticles, give heavy particles a lighter mass for faster convergence and artificially change the vacuum permittivity to increase the Debye length.

Another aspect is the use of symmetry in the simulation that stems from using a 2D or 3D solver. A number of parameters can make use of the axial-symmetry to simplify the simulation and other experimental tests can be used to confirm some parameters. However, leaving an unconstrained simulation (fully 3D) might show in a more straightforwards manner if new physics models employed have errors. This would be especially beneficial for a still experimental thruster such as the AF-MPDT. In addition, the analysis of the results of simulations should pay more attention to the radial velocity of electrons as it carries most of the current.

As part of the work for this project, a number of systems engineering aspects were designed. The most important of these are the top-level requirements and the corresponding lower level requirements that represent the first stage in the collaboration with a stakeholders that would develop a plasma simulation software. In subsequent meetings, there would be additional and other lower level requirements that help design the project more in depth. Also, there might be specific requirements in building or testing some parts that would be subjected to other stakeholders that provide funding. Furthermore, other aspects were analysed, such as the Input-Output Matrix. This is of high importance as it is based on literature analysis and interviews with stakeholders such that the software product would allow a faster testing of new physics models.

Three plasma simulation options were investigated. Then they were compared and a trade-off analysis was conducted. This yielded not a single best option for every scenario, but different suggestions depending on the available resources at NSS.

To better understand the suitability of simulation codes, one of the options was used to conduct simulations of a collisionless plasma sheath. Simulating a part of a full AF-MPD thruster would provide a first glance at the simulation capability of a stakeholder and where some issues might lie. The parameters analysed and compared were the electric potential, electric field, charged particles densities and the current flow to the outflow wall. The first aspect noticed was the length of time required which was drastically influenced by the choice of parameters. Using one of the options that used a 3D solver required more powerful hardware than what was available for this project and has not provided results. Thus, this showed that one of the options might require more powerful minimum hardware than stated. However, the option used for the simulations reached steady-state and yielded results close to the theoretical values. It was observed that the results presented suffered from numerical issues which caused slight variations in the expected results and might have delayed the steady-state. From these simulation it was clear that each part of the thruster has to be simulated individually first and tested heavily.

Looking at such a simple case, the simulation software cannot be compared against the requirements described in Section 4.2.

6.1. Answering The Research Sub-Questions

The findings from this report can be better organised by answering the research sub-questions, building towards the main research question.

The first research sub-question (*What is the current state of AF-MPD research?*) was answered mainly in Section 3. Thus, in terms of the hardware, more geometrical aspects and ranges of operational parameters need to be analysed. Complex propellants still require tests and a new LaB_6 cathode also need testing for the case of SUPREME. The majority of AF-MPD thrusters tested do not run at steady-state, high-voltage with high applied magnetic field like SUPREME, which leaves considerable room for research in this area. This can be fasten by the use of a plasma simulation code. Fully kinetic PIC-MCC codes are starting to be used for AF-MPDTs as well, but so far they focus on simulating the thrust mechanisms, with little focus on the phenomenons happening inside the thruster.

The second research question was: *Is a 3D model preferred over a 2D one to achieve thrust, erosion, losses results with errors of up to 5% from the values in experiments?* While most of the simulation software uses a 2D mesh with axi-symmetry, as seen in Section 3.7.3, 3D meshes provide slightly more accurate results. However, more work needs to be done to understand the impact of the accuracy and give a more quantitative result. Researchers in the field of electric propulsion agree on the need for fully 3D codes and their higher simulation fidelity, as different phenomenons can produce non axi-symmetric effects. In addition, once a 2D solver was implemented, it is a complex and time consuming process to change the code to have a 3D solver.

The third question (*What are the plasma simulation codes applicable for the AF-MPD developed by the NSS?*) was answered through literature analysis of available codes and interviews with the corresponding stakeholders. There were three options identified (option B, option S and option J), each with their best suitability depending on the available resources at NSS. However, from this report it was suggested that option J should be used, followed by option B and that they should be used in parallel.

The fourth question (*How would a simulation software tackle the backtracking phenomenon of charged* *particles?*) has different answer es depending on each of the stakeholders' approach. While option J proposes the use of the same software as for the main simulation, but with a simplification in the operating and simulation parameters, options B and S suggest a second, simpler code that tackles the backtracking phenomenon particularly and that takes inputs from the main code.

Answering the previous question and conducting the simulations of the example and test case helped in answering the sixth research sub-question (*What are the boundary conditions?*). Considering the thruster functioning and the backtracking phenomenon are treated separately, for the main code, the boundaries would be inner walls (electrodes), the backplate (including the inflows) and the free boundary. In the first stage of the simulation, the software would compute the heat flux towards the electrodes, but would not be able to also determine the final electrode temperature based on the cooling mechanisms employed. Thus, the domain stops at the surface with the electrodes. In addition, experiments with AF-MPDTs have shown large vacuum chambers are needed. This translates into the free domain at the exit requiring a larger space. A trade-off should be conducted between the time required for a larger simulation domain and a cut-off after a large enough percentage of the particles have been demagnetised.

With regards to the backtracking software, the simulation domain can be treated in 2D, as the trajectory of the particles along the thruster is not important. What is of interest is the amount of negative charge (as electrons demagnetise much later than ions) that comes back and how it is distributed.

In order for a requirement to be valid it has to be verifiable. Question six (*How would the requirements be verified?*) focuses on this and Section 4.3 went into details on the logic behind each requirement and how they would be verified. The four verification methods explained in Section 4.1 were exemplified for each method and for each requirement. The methods would apply to a prototype of a plasma simulation software and would use a set of data from NSS from other MPDT experiments.

The seventh research sub-question (*Can open-source codes be useful for the AF-MPDT simulations?*) was explained by conducting simulations on a test case. An open-source code maintained by one person was able to simulate in a limited time (a few weeks as a secondary objective) a collisionless plasma sheath with results similar in the first part to the theoretical expectations. Numerical problems were apparent and represent a main point of focus at first, if pursuing this option. However, the results of the simulations show potential for this option to be chosen to work on the plasma simulation software for SUPREME, especially if a longer time frame is allowed.

6.2. Recommendations For Future Work

Although the code requirements design and softwares assessment has concluded successfully, there is a number of activities that can continue the work of this project.

On one hand, future work can focus on the physics required for understanding the phenomenons in an AF-MPDT, especially SUPREME and the plasma simulation. This would start by conducting laboratory tests and collect data in order to answer the additional research sub-questions stated in Section 1.1. Further tests and analysis can be conducted in order to judge the literature review from this report achieved by analysing the work of other researchers in this field. By this, the results from tests of older versions of AF-MPDT would be compared (judged) against SUPREME and its operational parameters. These tests can be done after a fully operational plasma simulation software is developed and results in a SURPEME prototype being built.

On the other hand, the future work can focus on the code aspects. After a software choice is done and as previously stated, together with the corresponding stakeholder more requirements can be designed. In addition, multiple parts from a full thruster can be simulated individually and compare with experiments or theoretical expectations. From this, new requirements on the desired accuracy can be designed. For example, a natural row of steps starting from the collisionless plasma sheath case is to go to a normal plasma sheath (with collisions) and then to anode and cathode sheaths. After the first stage is ready, the person continuing this work can heavily test the code against available data from past experiments and check if it fulfils the requirements. Then, the code can be used on the same data to test new physics models and improve at each iteration. Moreover, the need for a minimum viable product (MVP) is high in order to attract more funding to NSS. As such, after the development of a SUPREME MVP, the new software can be used on the data sets with the operational parameters of SUPREME.

Regarding the Digital Twin, future work is limited to providing integration support to the NSS employee responsible for putting all the individual codes in a fully functioning digital thruster.

As can be seen, there is a range of exciting avenues of the work than can be done. This is based on the need for a plasma simulation code for SUPREME and the development of a SUPREME MVP.

Bibliography

[1] Drew Ahern. “INVESTIGATION OF THE GEM MPD THRUSTER USING THE MACH2 MAGNETOHYDRODYNAMICS CODE”. http://eplab.ae.illinois.edu/Publications/MS/Drew_Ahern.pdf. MA thesis. Graduate College of the University of Illinois at Urbana-Champaign, 2013.

[2] Riccardo Albertoni, Fabrizio Paganucci, and Mariano Andrenucci. “A phenomenological performance model for applied-field MPD thrusters”. In: *Acta Astronautica*. Vol. 107. https://www.scopus.com/record/display.uri?eid=2-s2.0-84915771417&origin=resultslist&sort=plf-f&src=s&sid=0f875d1e3e663652a32503a40a87e7b9&sot=b&sdt=b&sl=83&s=TITLE-ABS-KEY%28A+phenomenological+performance+model+for+applied-field+MPD+thrusters%29&relpos=0&citeCnt=14&searchTerm=. Mar. 2015, pp. 177–186.

[3] *Anaconda*. https://www.anaconda.com/. 2022.

[4] Mariano Andrenucci. “Magnetoplasmadynamic Thrusters”. In: *Wiley Online Library* (2010). https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470686652.eae118.

[5] Syed Ali Ashter. *Thermoforming of Single and Multilayer Laminates*. https://www.sciencedirect.com/book/9781455731725/thermoforming-of-single-and-multilayer-laminates#book-info. EMD Millipore Corporation, 2014. DOI: 10.1016/C2012-0-02821-9.

[6] Manuel La Rosa Betancourt, Marcus Collier-Wright, and Maieul Girard et. al. “APPLIED-FIELD MAGNETOPLASMADYNAMIC THRUSTERS (SUPREME) as an Enabling Technology for Next-generation of Space Missions”. In: *Journal of the British Interplanetary Society* 72.12 (Dec. 2019). https://bis-space.com/shop/product/jbis-vol-72-no-12-december-2019/, pp. 401–409.

[7] Manuel La Rosa Betancourt et al. *Telco Minutes*. Neutron Star Systems. Feb. 2020.

[8] A. Boxberger, A. Behnke, and G. Herdrich. “Current Advances in Optimization of Operative Regimes of Steady State Applied Field MPD Thrusters”. In: *36th International Electric Propulsion Conference*. http://electricrocket.org/2019/585.pdf. University of Vienna, Vienna, Austria: IEPC-2019-585, Sept. 2019.

[9] M. J. Boyle. “Acceleration processes in the quasi-steady magnetoplasmadynamic discharge”. https://ntrs.nasa.gov/citations/19750005557. PhD thesis. Princeton University, 1974.

[10] Charles Chelem Mayigué and Rodion Groll. “A density-based method with semi-discrete central-upwind schemes for ideal magnetohydrodynamics”. In: *Archive of Applied Mechanics* 87.4 (2017). https://link.springer.com/article/10.1007/s00419-016-1216-7, pp. 667–683. DOI: 10.1007/s00419-016-1216-7.

[11] ANVICA Software Development. *Convert kelvin [K] to electron-volt [eV]*. Online. https://www.translatorscafe.com/unit-converter/en-US/energy/70-12/kelvin-electron-volt/. Feb. 2022.

[12] G. A. Dyakonov et al. “Experimental Research of High Power MPD-thrusters in Pulse, Quasistationary and Stationary Modes”. In: *30th Joint Propulsion Conference*. https://arc.aiaa.org/doi/10.2514/6.1994-2994. Indianapolis, IN, U.S.A., June 1994.

[13] M. Fertig et al. “Hybrid Code Development for the Numerical Simulation of Instationary Magnetoplasmadynamic Thrusters”. In: *11th Results and Review Workshop on High Performance Computing in Science and Engineering, HLRS 2008*. https://link.springer.com/chapter/10.1007%2F978-3-540-88303-6_40. Stuttgart, Germany: High Performance Computing in Science and Engineering, Sept. 2008, pp. 585–597.

[14] Prof. Dr. Eberhard Gill. *Lecture Notes*. 2019.

[15] Georg Herdrich. “Advanced Space Propulsion (Master Class)”. In: *3rd General Assembly & 5th Steering Committee Meeting*. Ed. by University of Stuttgart. https://discoverer.space/wp-content/uploads/2019/08/2018_Session-5_DISCOVERER_MasterClass_Herdrich-komprimiert.pdf. Discoverer, Nov. 2018.

[16] Georg Herdrich, A. Boxberger, and D. Petkow et. al. “Advanced scaling model for simplified thrust and power scaling of an applied-field magnetoplasmadynamic thruster”. In: *46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*. https://arc.aiaa.org/doi/10.2514/6.2010-6531. Nashville, TN: Aerospace Research Center, July 2012.

[17] Mengdi Kong, Haibin Tang, and Wenjiang Yang. “Analyses of Ignition Processes of an Applied-Field Magnetoplasmadynamic Thruster”. In: *53rd AIAA Aerospace Sciences Meeting*. Ed. by Sciotech Forum. https://arc.aiaa.org/doi/10.2514/6.2015-1612. Kissimmee, Florida: Aerospace Research Central, 2015.

[18] Gerd Krülle, Monika Auweter-Kurtz, and Akihiro Sasoh. “Technology and application aspects of applied field magnetoplasmadynamic propulsion”. In: *Journal of Propulsion and Power* 14.5 (1998). https://arc.aiaa.org/doi/10.2514/2.5338, pp. 754–763.

[19] Michael R. LaPointe and Pavlos G. Mikellides. “High Power MPD Thruster Development at the NASA Glenn Research Center”. In: *AIAA* (Aug. 2001). https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.205.7181&rep=rep1&type=pdf.

[20] Dan Lev and Edgar Y. Choueiri. “Scaling of Anode Sheath Voltage Fall with the Operational Parameters in Applied-Field MPD Thrusters”. In: *IEPC-2011-222*. https://alfven.princeton.edu/publications/publications/personnel/personnel/publications/lev-iepc-2011-222. Wiesbaden, Germany, Sept. 2011.

[21] Jian Li et al. “Particle Simulation Model for Self-Field Magnetoplasmadynamic Thruster”. In: *Energies* 12.8 (Apr. 2019). https://www.mdpi.com/1996-1073/12/8/1579.

[22] Michael A. Lieberman and Allan J. Lichtenberg. *Principles of Plasma Discharges and Materials Processing*. Second. https://www.wiley.com/en-us/Principles+of+Plasma+Discharges+and+Materials+Processing%2C+2nd+Edition-p-9780471720010. John Wiley & Sons, Inc, 2005.

[23] Tanase-Marius Manafu. “Literature Study”. MA thesis. TU Delft, 2021.

[24] Maris A. Mantenieks and Vincent K. Rawlin. “Sputtering phenomena of discharge chamber components in a 30-cm diameter Hg ion thruster”. In: *12th International Electric Propulsion Conference*. https://ntrs.nasa.gov/citations/19770004161. Key Biscayne, FL, U.S.A: NASA Lewis Research Center Cleveland, Jan. 1976.

[25] Manuel Martinez-Sanchez and Paulo Lozano. *Space Propulsion*. https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-522-space-propulsion-spring-2015/. MIT. 2015.

[26] Stéphane Mazouffre. “Electric propulsion for satellites and spacecraft: established technologies and novel approaches”. In: *IOPscience* 25.3 (2016). https://iopscience.iop.org/article/10.1088/0963-0252/25/3/033002.

[27] National Library of Medicine: National Center for Biotechnology Information. *PubChem*. online. https://pubchem.ncbi.nlm.nih.gov/. Oct. 2021.

[28] Rosa Muñoz. *Differential Sputtering of Electric Propulsion Grid Material by Ion Bombardment*. McNair Scholars Summer 2005 Research Program. https://projects-web.engr.colostate.edu/ionstand/publications/papers/McNair-2005.pdf. 2005.

[29] Roger M. Myers. “Applied-field MPD thruster performance with hydrogen and argon propellants”. In: *Journal of Propulsion and Power* 9.5 (1993). https://arc.aiaa.org/doi/10.2514/3.23691, pp. 781–784. DOI: 10.2514/3.23691.

[30] Roger M. Myers and George C. Soulas. “Anode Power Deposition in Applied-Field MPD Thrusters”. In: *28th Joint Propulsion Conference and Exhibit*. https://ntrs.nasa.gov/citations/19920071410. July 1992.

[31] Mihchael R. Nakles et al. “Experimental And Modeling Studies of Low Energy Ion Sputtering in Ion Thrusters”. In: *39th Joint Propulsion Conference and Exhibit*. `https://arc.aiaa.org/doi/abs/10.2514/6.2003-5160`. July 2003.

[32] *ParaView*. `https://www.paraview.org/`. 2022.

[33] K. Sankaran, S. C. Jardin, and E. Y. Choueiri. “Comparison of Simulated Plasma Flowfields to Experimental Measurements for a Gas-Fed Magnetoplasmadynamic Thruster”. In: *39th Joint Propulsion Conference and Exhibit*. `https://alfven.princeton.edu/publications/publications/personnel/personnel/publications/sankaran-jpc-2003-4843`. July 2003.

[34] SPARC Industries SARL. *VSTRAP Software User Manual v0.2*. Beta version 0.2. 20 Rue du Commerce, Technoport 1, L-3895 Foetz, Luxembourg, 2021 2021.

[35] Akihiro Sasoh, Hayato Kasuga, and Yoshiya Nakagawa et. al. “Electrostatic-magnetic-hybrid thrust generation in central–cathode electrostatic thruster (CC–EST)”. In: *Acta Astronautica* 152 (2018). `https://www.sciencedirect.com/science/article/pii/S0094576518302820`. ISSN: 0094-5765. DOI: `https://doi.org/10.1016/j.actaastro.2018.07.052`.

[36] James S. Sovey and Maris A. Mantenieks. “Performance and Lifetime Assessment of MPD Arc Thruster Technology”. In: *24th Joint Propulsion Conference*. `https://ntrs.nasa.gov/citations/19880020476`. Boston, Massachusetts, July 1988.

[37] Neutron Star Systems. *SUPREME TECHNOLOGY*. `https://www.neutronstar.systems/the-tech`. 2021.

[38] Hai-Bin Tang et al. “Study of applied magnetic field magnetoplasmadynamic thrusters with particle-in-cell and Monte Carlo collision. II. Investigation of acceleration mechanisms”. In: *Physics of Plasmas* 19.7 (2012). `https://aip.scitation.org/doi/full/10.1063/1.4737104?ver=pdfcov`. ISSN: 1070-664X. DOI: `10.1063/1.4737104`.

[39] Hai-Bin Tang et al. “Study of applied magnetic field magnetoplasmadynamic thrusters with particle-in-cell code with Monte Carlo collision. I. Computation methods and physical processes”. In: *Physics of Plasmas* 19.7 (2012). `https://aip.scitation.org/doi/10.1063/1.4737098`. ISSN: 1070-664X. DOI: `10.1063/1.4737098`.

[40] WebQC. *Chemical Portal*. online. `https://www.webqc.org/`. Oct. 2021.