

**MASARYKOVA UNIVERZITA**  
**PŘÍRODOVĚDECKÁ FAKULTA**  
ÚSTAV TEORETICKÉ FYZIKY A ASTROFYZIKY

# **Bakalářská práce**

**BRNO 2020**

**MATÚŠ LABAJ**

**MASARYKOVA  
UNIVERZITA**  
PŘÍRODOVĚDECKÁ FAKULTA  
ÚSTAV TEORETICKÉ FYZIKY A ASTROFYZIKY

---

# **Particle-in-Cell simulace termální fronty**

Bakalářská práce

**Matúš Labaj**

Vedoucí práce: prof. RNDr. Marian Karlický, DrSc.

Brno 2020

# Bibliografický záznam

**Autor:** Matúš Labaj  
Přírodovědecká fakulta, Masarykova univerzita  
Ústav teoretické fyziky a astrofyziky

**Název práce:** Particle-in-Cell simulace termální fronty

**Studijní program:** Fyzika

**Studijní obor:** Astrofyzika

**Vedoucí práce:** prof. RNDr. Marian Karlický, DrSc.

**Akademický rok:** 2019/2020

**Počet stran:** 8 + 62

**Klíčová slova:** Termální fronta, Simulace, Sluneční atmosféra, Particle-in-Cell, Astrofyzika plazmatu

# Bibliographic Entry

**Author:** Matúš Labaj  
Faculty of Science, Masaryk University  
Department of Theoretical Physics and Astrophysics

**Title of Thesis:** Particle-In-Cell Simulation of the Thermal Front

**Degree Programme:** Physics

**Field of Study:** Astrophysics

**Supervisor:** prof. RNDr. Marian Karlický, DrSc.

**Academic Year:** 2019/2020

**Number of Pages:** 8 + 62

**Keywords:** Thermal front, Simulation, Solar Atmosphere, Particle-in-Cell, Plasma astrophysics

# Abstrakt

Fenomén termálnej fronty vznikajúci v koronálnych slučkách je jedným z mnohých súčasných problémov slnečnej astrofyziky. Práca sa zameriava na simulácie termálnej fronty použitím Particle-In-Cell metódy aplikovanej na prechod medzi horúcou a chladnou plazmou. V práci je stručne objasnený fyzikálny podklad slnečnej fyziky, ktorý uvádza do problematiky koronálneho ohrevu, prechodovej oblasti a termálnej fronty. Ďalej je vysvetlený potrebný aparát fyziky plazmy a simulovaných javov, nutný pre pochopenie nastavenia a interpretácie simulácií. Nasleduje vysvetlenie princípu fungovania a konfigurácie fyzikálnych parametrov simulačnej metódy Particle-In-Cell použitím upraveného kódu TRISTAN. Posledná časť práce zohľadňuje doterajší výskum termálnej fronty, jej simulácie a predstavuje výsledky simulácií s odlišnými konfiguráciami rozhrania medzi horúcou a chladnou plazmou a rozdielného nastavenia tlakovej rovnováhy.

# Abstract

The phenomenon of the thermal front that forms in the coronal loops is one of many modern problems of solar astrophysics. The aim of this work is the simulations of the thermal front using the Particle-In-Cell method applied on the hot-cold plasma transition. The work briefly illustrates the physical background of solar physics, which introduces into the problematics of coronal heating, transition region and thermal front. Moreover, the necessary fundamentals of plasma physics and simulated phenomena are explained to understand the configuration and interpretation of the simulations. This is followed by an explanation of the principles and physical properties configuration of the Particle-In-Cell simulation method. A modified TRISTAN code is used. The last part of the work addresses the present research of thermal front, its simulations, and discusses the simulation results for different configurations of the initial hot-cold plasma interface and the initial configuration of pressure equilibrium.

ZADÁNÍ  
BAKALÁŘSKÉ PRÁCE

Akademický rok: 2019/2020

<b>Ústav:</b>	Ústav teoretické fyziky a astrofyziky
<b>Student:</b>	Matúš Labaj
<b>Program:</b>	Fyzika
<b>Obor:</b>	Astrofyzika

Ředitel Ústavu teoretické fyziky a astrofyziky PřF MU Vám ve smyslu Studijního a zkušebního řádu MU určuje bakalářskou práci s názvem:

<b>Název práce:</b>	Particle-in-Cell simulace termální fronty
<b>Název práce anglicky:</b>	Particle-in-cell simulation of the thermal front
<b>Jazyk závěrečné práce:</b>	

**Oficiální zadání:**

Kontext: Během sluneční erupce vzniká ve vrcholu magnetické smyčky oblast zachyceného horkého plazmatu. Ve směru kolmém k magnetickému poli drží horké plazma magnetické pole. Způsob zachycení horkého plazmatu podél magnetického pole však zůstává stále otázkou. Pro toto zachycení plazmatu Brown et al. 1979 navrhl mechanismus s termální frontou. Li et al. (2012), Karlický (2015) a Rober-Clark (2018) studovali tento mechanismus pomocí metody Particle-in-Cell a ukázali, že tok tepla je na rozhraní teplého a chladného plazmatu potlačen pomocí kinetických efektů. Cíle: Seznámení studenta se základními kinetickými procesy v oblasti termální fronty. Student použije Particle-in-Cell model TRISTAN k simulaci termální fronty pro několik různých počátečních poměrů teplot, hustot nebo distribučních funkcí rychlostí částic. Tyto počáteční podmínky v kódu nastaví, vytvoří sadu skriptů na zpracování dat ze simulace, spočítá simulace a vytvoří grafy, které budou použity v bakalářské práci. Doporučená literatura: Birdsall-Langdon: 1985, Plasma Physics via Computer Simulation, McGraw-Hill, New York Matsumoto-Yomura: 1993, Computer Space Plasma Physics, Terra Scientific Publishing Company, Tokyo Karlický, M., 2014, Plasma Astrophysics, MFF UK Praha, vysokoškolská učebnice.

<b>Vedoucí práce:</b>	prof. RNDr. Marian Karlický, DrSc.
<b>Konzultant:</b>	Mgr. Jan Benáček, Ph.D.
<b>Datum zadání práce:</b>	18. 1. 2020
<b>V Brně dne:</b>	20. 1. 2020

Souhlasím se zadáním (podpis, datum):

.....  
Matúš Labaj  
student

.....  
prof. RNDr. Marian Karlický, DrSc.  
vedoucí práce

.....  
prof. Rikard von Unge, Ph.D.  
ředitel Ústavu teoretické fyziky a  
astrofyziky

# Poděkování

V prvom rade by som rád poďakoval rodine za ich prítomnosť, neoceniteľnú podporu a postrehy pri písaní mojej práce. Ďalej by som rád prejavil veľkú vďaku doktorovi Janovi Benáčkovi, za jeho trpezlivosť pri vysvetľovaní problematiky a samotného kódu dlho predtým, než som vedel, že bude predmetom mojej práce, a jeho príkladnej dostupnosti v čase nepríjemnej situácie okolo svetovej pandémie. Moje poďakovanie tatiež patrí pánovi profesorovi Marianovi Karlickému za jeho poznámky, ktoré mi pomohli problematiku lepšie uchopiť. V neposlednom rade by som rád poďakoval Národnému superpočítačovému centru IT4Innovations, za poskytnutie výpočetného výkonu superpočítača Salomon.

# Prohlášení

Vyhlasujem, že som svoju bakalársku prácu vypracoval samostatne pod vedením vedúceho práce s využitím informačných zdrojov, ktoré sú v práci citované.

Brno 1. 6. 2020

.....  
Matúš Labaj

# Contents

<b>Introduction</b>	<b>1</b>
<b>1 The Sun</b>	<b>2</b>
1.1 The Sun and its Internal Structure	2
1.1.1 The Sun as a Star	2
1.1.2 The Core of the Sun	3
1.1.3 Radiative Zone	3
1.1.4 Tachocline	3
1.1.5 Convective Zone	4
1.2 Atmosphere of Sun	5
1.2.1 Photosphere	5
1.2.2 Chromosphere	5
1.2.3 Transition Region	6
1.2.4 Corona	7
1.3 Historical Observations of the Solar Corona	9
1.4 Coronal Heating Phenomena	9
1.4.1 Magnetic Field Reconnection	10
1.4.2 Magnetoacoustic Waves	10
<b>2 Physics of Plasma</b>	<b>12</b>
2.1 Concept of Plasma	12
2.2 Classification and Attributes of Plasma	13
2.2.1 Types of Plasmas	13
2.2.2 Debye Shielding	14
2.2.3 Plasma Oscillations	14
2.2.4 Cyclotron Motion	15
2.3 Equations Describing Plasma Phenomena	15
2.3.1 Kinetic Description	16
2.3.2 Magnetohydrodynamic Description	17
2.3.3 Collisional and Anomalous Resistivity	18
2.3.4 Induction Equation	19
2.3.5 Plasma- $\beta$ Parameter	19
2.4 Double Layer Phenomena	20
2.4.1 General Description of Double Layers	20
2.4.2 One-Dimensional DL Model	21

<b>3</b>	<b>Particle-In-Cell Code</b>	<b>24</b>
3.1	Fundamentals of the PIC Method . . . . .	24
3.1.1	Mathematical Formulation of PIC . . . . .	24
3.1.2	Implementation of the Equations Using the Leap-Frog Algorithm . . . . .	25
3.1.3	Stability of the Simulation . . . . .	26
3.2	TRISTAN . . . . .	26
3.2.1	Field Update . . . . .	26
3.2.2	Particle Update . . . . .	27
3.2.3	Current Decomposition . . . . .	28
3.2.4	Speed Limits . . . . .	29
3.2.5	Boundary Conditions . . . . .	29
3.2.6	Parallelization . . . . .	29
3.2.7	Model Workflow . . . . .	30
3.2.8	Improvements implemented in the Code . . . . .	31
<b>4</b>	<b>Hot-Cold Plasma Transition</b>	<b>33</b>
4.1	Thermal Fronts . . . . .	33
4.2	Previous Theoretical Research . . . . .	33
4.3	Thermal Front Formation Simulations . . . . .	36
4.4	Unequilibrated and Equilibrated Pressure Models . . . . .	39
	<b>Discussion and Future Insights</b>	<b>49</b>
	<b>Appendix</b>	<b>51</b>
	<b>Bibliography</b>	<b>60</b>

# Introduction

The Sun as a star is a focal point of modern astrophysical research. We currently possess a lot of information about phenomena on the Sun. We understand their formation, evolution, and ongoing processes. However, there is a plethora of questions left to be answered that are needed to correctly describe the solar phenomena. This work will be devoted to the one of these questions – the problem of interaction of plasmas with different temperature on the kinetic level and the impact of pressure equilibrium on its stability. These problems directly relate to the problem of the thermal front that is not yet sufficiently described. We will simulate the region using Particle-In-Cell code TRISTAN for various initial conditions and try to interpret the data into various evolution graphs.

The solar atmosphere is in the state of plasma. We will devote one of the upcoming chapters to describe plasma, its characteristics, ongoing processes, and phenomena needed to understand solar atmosphere accordingly. Furthermore, we will address the structure of the Sun and its vital elements such as solar activity, eruptions, ongoing and past research, and their respective consequences and outcomes.

We are severely limited in the information we can gather from solar atmospheres, as they are only provided via electromagnetic waves and particle beams. Therefore we are unable to gather sufficient data from the corona. We choose a different approach, which is conducting Particle-In-Cell (PIC) simulations that are set to meet the conditions of given solar plasma. Using PIC, we can calculate the evolution of the studied hot-cold plasma transition region. To achieve accurate results, we have to correctly assume initial conditions and run the simulation on a sufficient amount of processors. We will discuss the matter in the later part of the work.

In the last part of the paper, we will interpret the data in various time evolution figures. By the means of data analysis, we will comment on the occurring physical effects. Last but not least, we will discuss the consequences and future possibilities of using the PIC method.

# Chapter 1

## The Sun

### 1.1 The Sun and its Internal Structure

#### 1.1.1 The Sun as a Star

The Sun is a star of the main sequence. It is a G2V star based on its spectral class. However, it is the distance, with Sun being the closest star to the Earth, that makes it extraordinary and vitally important for the existence of life on the Earth. Furthermore, it is a prime candidate for the research and understanding of physical plasma and behavior of other stellar objects in space. The base characteristics of the Sun are concluded in the Table 1.1. The Sun, the same as the other stars, has a coherent shape of a sphere because it satisfies the condition of hydrostatic equilibrium – the force caused by the gas pressure and radiation pressure directed out of star is equal to the gravitational force directed to the center of the star. From the chemical point of view, the Sun consists mainly of hydrogen (90 %), hydrogen (8 %), and other elements (2 %) [6], such as oxygen, carbonium, potassium, silicon, and others. The Sun consists of several layers (Figure 1.1), which we will discuss further.

Age	$4.5 \cdot 10^9$ years
Mass	$2.0 \cdot 10^{30}$ kg
Diameter	$1.4 \cdot 10^6$ km
Gravitational acceleration	$274 \text{ m} \cdot \text{s}^{-2}$
Escape velocity	$618 \text{ m} \cdot \text{s}^{-1}$
Equatorial rotation period	$\approx 26$ days
Mean density	$1.4 \cdot 10^3 \text{ kg} \cdot \text{m}^{-3}$
Radiant flux	$3.8 \cdot 10^{26}$ W
Effective surface temperature	5785 K

Table 1.1: Fundamental physical parameters of the Sun [7].

## 1.1.2 The Core of the Sun

The core of the Sun is the source of thermonuclear reactions that release the vast majority of Sun's energy. Light cores of Hydrogen are being fused into the heavier core of Helium, releasing energy ( $\approx 26.8$  MeV) that is eventually radiated from the solar surface. The core of the Sun takes up to a quarter of the star's radius. Temperature of the core is  $\approx 1.5 \cdot 10^7$  K. The estimated core pressure is  $2.5 \cdot 10^5$  Pa, the mean density is  $1.5 \cdot 10^5$  kg  $\cdot$  m $^{-3}$ .

## 1.1.3 Radiative Zone

In the distance up to around 0.68 of the solar radius, the region is called the radiative zone. This zone is in radiative equilibrium, therefore the energy in this region is transferred by thermal radiation. Equation of heat transfer is expressed as change of temperature  $T$  with distance from core  $r$

$$\frac{dT}{dr} = -\frac{3\kappa\rho L}{16T^3 4\sigma\pi r^2} \quad (1.1)$$

where  $\kappa$  denotes the absorption coefficient (also called *opacity*) of star's material,  $\sigma$  is the Stefan-Boltzmann constant,  $\rho$  is the density and  $L$  is luminosity. The temperature in the radiative zone is expected to be in the range of  $7 \cdot 10^5 - 2 \cdot 10^6$  K. It is apparent from Equation 1.1 that with increasing distance, the temperature is decreasing.

## 1.1.4 Tachocline

Tachocline, which existence was derived by the data provided via the SOHO probe, is layer following the radiative zone in distance 0.68 – 0.72 of solar radius from the core. Most of the information about this region are yet to be discovered. However, we do know that it is the borderline zone between the radiative zone and the following convective zone. What makes the role of the region significant, is that it is located in between two zones with varying rotational velocities. Under the tachocline, the Sun rotates more-less as a solid

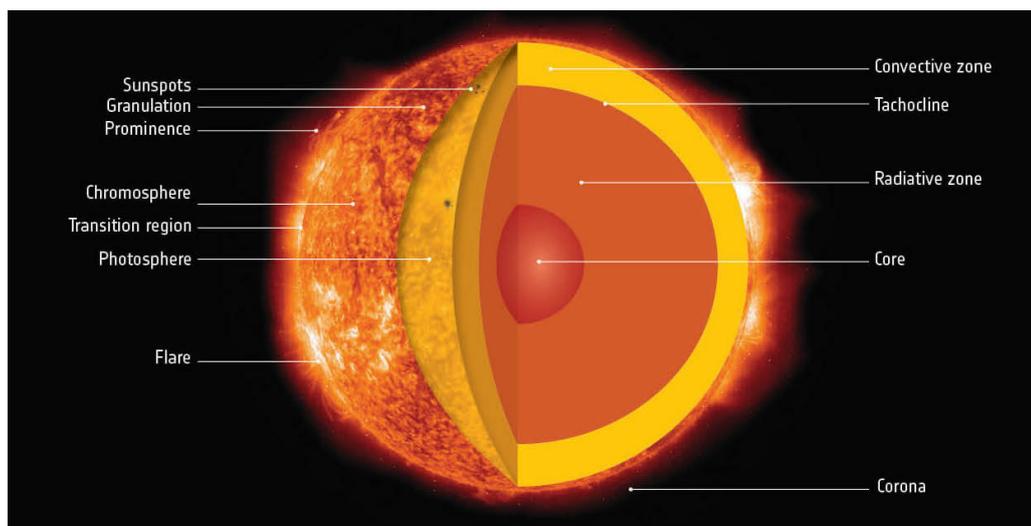


Figure 1.1: Structure of the Sun [40].

object, the rotation over the tachocline is of differential character (Figure 1.2), therefore the angular velocity is dependent on the distance from Sun's core. It is also commonly accepted that due to differential rotation, the region of tachocline generates magnetic fields through so-called *magnetohydrodynamic dynamo* (we will discuss the matter in further chapters).

### 1.1.5 Convective Zone

The convective zone follows the tachocline and ends nearly at the surface of the Sun. The solar plasma in this region is no longer hot enough to support transferring energy through radiation, consequently, the energy is transferred by convection, as the term for this region suggests. The temperature at the top of most parts of the region decreases up to around 6000 K. We can describe the transfer of energy using the following equation:

$$\frac{dT}{dr} = \left(1 - \frac{1}{\gamma}\right) \frac{T}{p} \frac{dp}{dr}. \quad (1.2)$$

The left side of Equation 1.2 describes the temperature gradient, the right side stands for the adiabatic gradient, therefore the decrease of gas temperature and pressure due to its expansion. The pressure gradient on the right side is given by

$$\frac{dp}{dr} = -G \frac{M}{r^2} \rho, \quad (1.3)$$

which is the equation for hydrodynamic stability of star where  $G$  is gravitational constant,  $M$  is the mass of the star,  $r$  its radius and  $\rho$  is the mass density dependent on the radius. We see, that the Equation 1.2 defines the same physical quantity as the Equation 1.1. If the absolute value of temperature gradient of the Equation 1.1 exceeds one of the Equation 1.2, the energy is transferred by convection. This inequality is commonly known

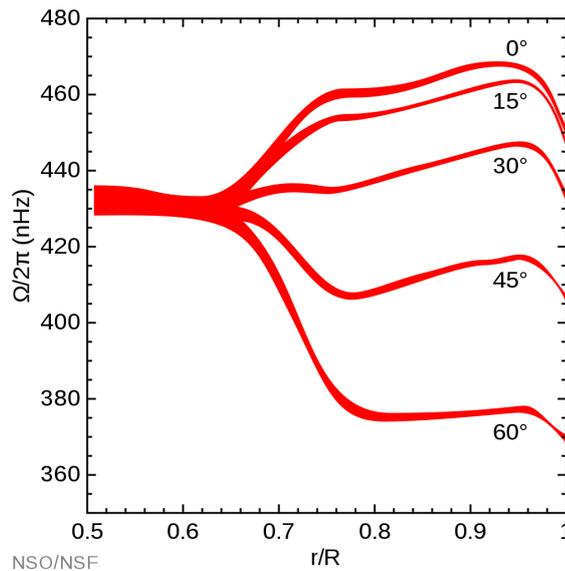


Figure 1.2: Dependency of angular velocity on the distance from core [41].

as *Schwarzschild's criterion*. It is worth mentioning that a small part of the energy is transferred by conduction, but the impact is negligible, therefore we do not take it into account.

## 1.2 Atmosphere of Sun

### 1.2.1 Photosphere

The photosphere is the lowest region of the solar atmosphere and also the directly observable surface of the Sun. Its temperature of  $\approx 5800$  K corresponds with the spectral classification of the white color. The photosphere is also the source of many phenomena, such as sunspots, granulation, and supergranulation. The granulation is the effect of energy transfer from lower layers of the Sun – the peaks of thermal streams coming from the convective zone are the cause of the visible solar granulation. The newest, most detailed view of solar granulation (Daniel K. Inouye Solar Telescope) is depicted in Figure 1.3.

### 1.2.2 Chromosphere

The chromosphere is the region of Sun's atmosphere that follows the photosphere. It is not observable directly by the human eye, however, it can be seen during the eclipse of Sun as a red glow around the solar disc. The chromosphere consists of the so-called *low chromosphere*, filled by neutral gas with a temperature of  $\approx 4200$  K and *high chromosphere* filled with fully ionized plasma with a temperature of  $\approx 20000$  K. It is estimated to be between 2000 up to 2500 km high. The dynamics of the chromosphere are therefore governed by kinetic pressure. The most distinctive phenomena occurring in the chromosphere are floccula fields and spicules, which are dynamic jets of hot plasma material that erupted from lower regions. Spicules can be usually found around the edges of supergranulation

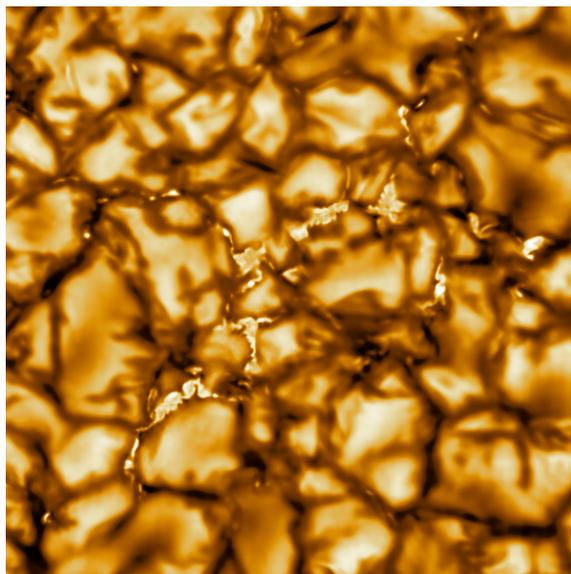


Figure 1.3: Solar granulation of the photosphere (2020) [42].

formations, as seen in Figure 1.3, with a strong magnetic field. The process of spicule formation is not satisfactorily described at this moment, however, they are suspected to be the manifestation of magnetic flux tubes that transfer the plasma above the Sun's surface (Figure 1.4). At this time, it is essential to mention the plasma- $\beta$  parameter that will be explained in later Chapter 2.3.5. The plasma- $\beta$  parameter of the chromosphere goes from high to low values and subsequently, a plasma with kinetic collisions becomes nearly fully collisionless plasma.

### 1.2.3 Transition Region

The very thin region dividing the chromosphere and solar corona is called *transition region*. This region is where the relatively cold plasma of the upper chromosphere comes to contact with the hot plasma of solar corona. Since we do not possess enough information about this region, the simulations that we will discuss in further chapters aim to replicate the environment to keep track of ongoing physical processes. The thing that strikes us most about the transition region is the sudden increase in temperature. It has proven to be difficult to correctly measure the layer's thickness – it is estimated to be thick in order of hundreds of kilometers. The process of steep temperature increase happens fully in this region with temperature going from above stated  $\approx 20000$  K up to 1 – 2 MK.

*The Transition Region and Coronal Explorer* (TRACE) probe, launched in 1998, provided a unique view of the solar outer atmosphere. TRACE observations point towards a corona comprised of thin loops that are dynamic and continually evolve. These very thin strings are heated for a certain time (ranging from some minutes to tens of minutes), after which the heating ceases. Heating appears to occur mainly in the distance of 10 000 to 20 000 kilometers from the solar core [31].

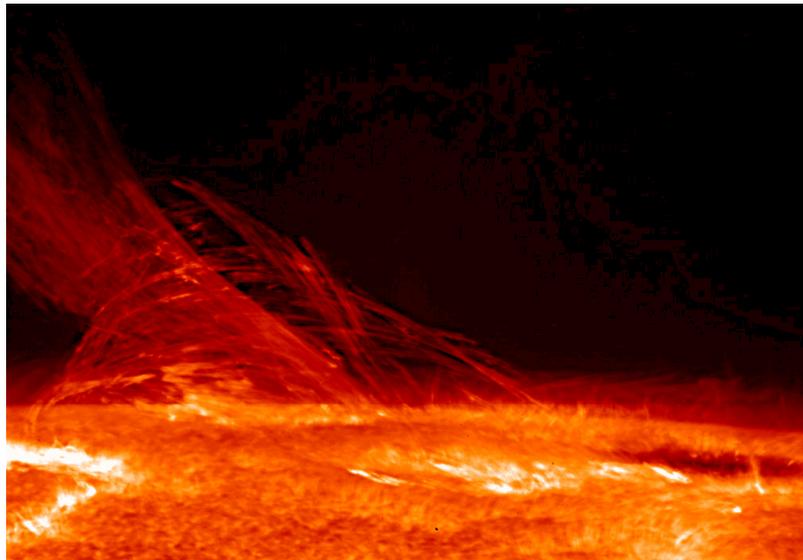


Figure 1.4: Picture of chromosphere captured by the Hinode solar observatory. The image reveals the filamentary nature of the plasma connecting regions of different magnetic polarity. (2007) [43].

The term *loop* is commonly used to describe bright coronal structures that are very long and thin. These loops outline the coronal magnetic field, primarily due to heat conduction and mass transport across the field being strongly suppressed. The loops contain flare-related heat flows, in which the so-called thermal fronts are generated.

## 1.2.4 Corona

The solar corona is the topmost layer of the solar atmosphere. It is not delimited by a fixed border as it extends into the solar wind. Its shape has a dynamic character as it is dependent on solar activity. Corona is not observable during typical daytime, as it is outshined by the core sunlight, though it can be observed during a solar eclipse (Figure 1.6). Eclipse observations provided us with first attempts to estimate density models of corona and also discovered the below stated *coronal holes*. In August 2018, the Parker Solar Probe was launched by NASA. The goal of the probe is to come as close as 10 solar diameters and trace how energy and heat move through the solar corona.

The average temperature of the corona is around 2 MK. However, the temperature is not distributed homogeneously. The cold spots, so-called *coronal holes* that are dominated by open magnetic field configurations, have a temperature lower than 1 MK. On the other side of the temperature spectrum are the so-called *active regions*, where the temperature gets as high as 6 MK.

The evolution of the plasma- $\beta$  parameter is shown in Figure 1.5. In the graph, we see that the transition region and inner corona region have plasma- $\beta$  parameter  $< 1$ , followed by outer corona, where the plasma- $\beta$  parameter is  $> 1$ . This means that the local magnetic field is very weak. This happens in the case of so-called *magnetic cusps* where thanks to the high temperature the dynamics are governed by thermal kinetic pressure. As a result, the flowing plasma can carry the plasma with its motion being perpendicular to the magnetic field lines. If the plasma- $\beta$  parameter is  $< 1$ , it is the magnetic pressure that governs the coronal plasma dynamics – the plasma can only flow along the magnetic field lines.

The huge temperature gradient between the solar chromosphere and corona (in the transition region) has not yet been satisfactorily described. Current theories apply the means of the magnetic reconnection processes or heating by waves in the solar plasma.

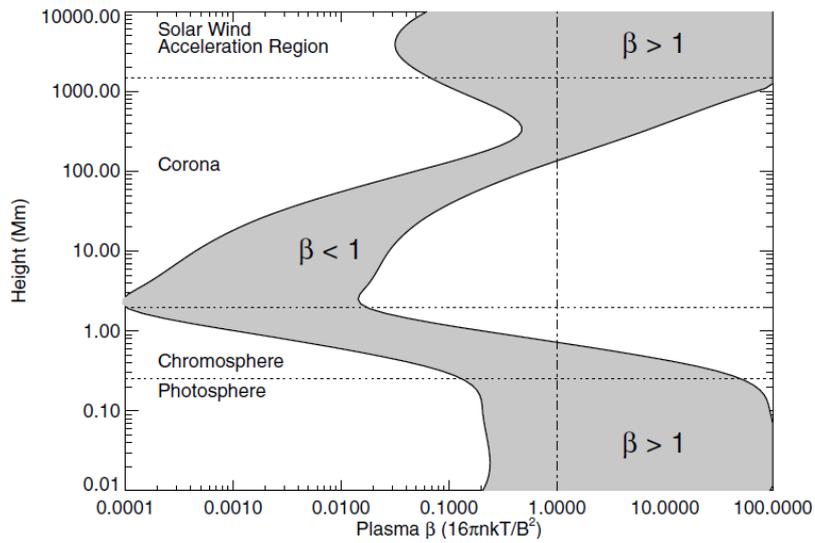


Figure 1.5: *Plasma- $\beta$*  parameter in the solar atmosphere assumed for two magnetic field strengths of 100 and 2500 G. (Gary [14]).

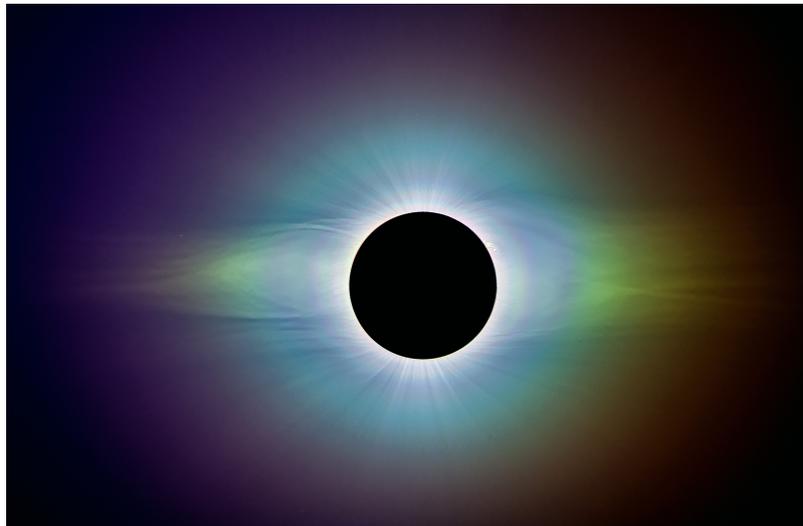


Figure 1.6: Image taken by the ESA-CESAR team at the total solar eclipse, La Silla Observatory, 2 July 2019. It was made by combining multiple polarised images of the solar corona during totality to bring out the details in its structure [44].

### 1.3 Historical Observations of the Solar Corona

Most of the solar optical radiation is emitted by the photosphere. The optical emission produced by Thomson scattering of the atmospherical regions above is in many orders of magnitude less intense and subsequently visible only during the solar eclipse. The first observations of solar corona date back to sources of ancient civilizations such as Indians, Babylonians, and Chinese. Regular observations of solar eclipses started in 1842 when corona was observed by the likes of Airy, Arago, Baily, Littrow, and Struve. The first photographs were taken in 1851 during the solar eclipse in Norway and Sweden by Berkowski. The fact that solar corona contains helium was discovered by Janssen in 1868. The very first coronagraph was built at the Pic-du-Midi Observatory in 1930 which is an instrument that covers the solar disc and thus allows the observation without the need for a solar eclipse.

If we observe the Sun in wavelengths other than optical, such as soft or hard X-rays or radio wavelengths, the photosphere becomes invisible and corona produces the most intense emission. A breakthrough in coronal observations, therefore, came with the start of the space era in the 1950s. This enabled the observations in the above-mentioned wavelength regimes. As the details of the observations exceed the idea of this work, we will not discuss them further. The timeline of observing instruments is displayed in Figure 1.7.

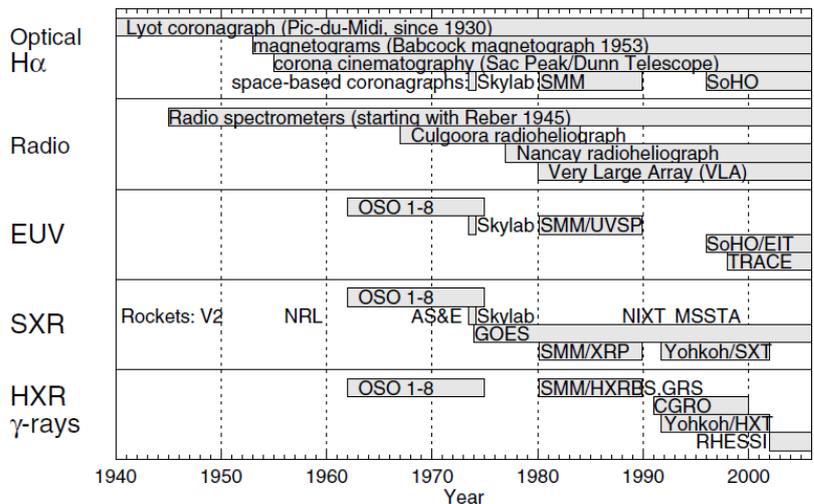


Figure 1.7: Timeline of the operational periods of instruments and space missions that provided unique observations of solar corona distinguished by different wavelength observations [5].

### 1.4 Coronal Heating Phenomena

As we stated many times, the process of coronal heating is one of many unsolved phenomena occurring in the solar atmosphere. In principle, it is suspected to be caused by two mechanisms: the energy release caused by the reconnection of the magnetic field and heating via magnetoacoustic waves.

### 1.4.1 Magnetic Field Reconnection

The boundary conditions of solar corona have a dynamic character. The solar dynamo in the internal regions of Sun generates magnetic flux at the bottom of the tachocline, which rises by buoyancy and emerges through the photosphere into the corona. The dynamic conditions are further supported by differential rotation of the star (this wraps up the coronal field with every rotation) and the fact that the connectivity to the interplanetary field is constantly broken up to avoid excessive magnetic stress. These conditions are the fundamental reason behind the adjustments made by the coronal field, which are called the *magnetic reconnection* processes. As this phenomenon is not trivial and its full explanation is beyond the objectives of this work, we will not address it in detail.

The reconnection process may occur in areas where two separate magnetic fields with antiparallel field lines get near each other. The value of the magnetic field is low in this region. In the location of their closest contact, the diffusion region is created. The field configuration is changed in the diffusion region (the process of reconnection) into a field configuration that is energetically more efficient. During the process, enormous electric currents can flow through the region. This heats the surrounding plasma which subsequently transfers the heat further into the solar corona. The magnetic reconnection not only operates locally in flares, but it also organizes the global corona by restructuring its field. However, only steady 2D reconnection models can be formulated analytically, which provide basic relations for inflow speed, outflow speed, and reconnection rate, but they only represent oversimplifications for observed solar flares. There are 2 commonly used models of magnetic field reconnection – Sweet – Parker and Petschek model (Figure 1.8). The Sweet–Parker model (Sweet [2]) considers the diffusion region as a long thin sheet. The field equations (fully included in [1]) result in half of the inflowing magnetic energy being converted to kinetic energy, while the remaining half is converted to thermal energy. This means that the reconnection generates hot fast streams of plasma. In the Petschek model (Petschek [3]), most of the energy conversion is situated into the standing slow-mode shocks. These shocks accelerate and heat the surrounding plasma, resulting in 2/5 of inflowing magnetic energy being converted to thermal energy and the remaining 3/5 being converted to kinetic energy. Consequently, these shocks could be the way of heating the corona by transforming the energy of a strong electromagnetic field below the corona region into the thermal and kinetic energy of the particles in the corona.

### 1.4.2 Magnetoacoustic Waves

Another very influential process that contributes to the heating of solar corona is due to magnetoacoustic waves. However, the wave propagation is not as trivial, as the plasma environment has anisotropic character thanks to the present magnetic field. The magnetoacoustic wave in the plasma is not propagated in a single spherical equiphase wave surface (the case of waves spread in air environment), instead they are propagated in 3 equiphase surfaces: fast magnetoacoustic waves, slow magnetoacoustic waves, and Alfvén waves. Recent studies show that it could be the Alfvén waves process that mainly contributes to the phenomena of coronal heating (De Pontieu [8], Tomczyk [9]). These waves are capable of transferring energy to upper regions of the solar atmosphere along the magnetic field lines. These waves transfer the energy from lower solar regions into higher regions, where

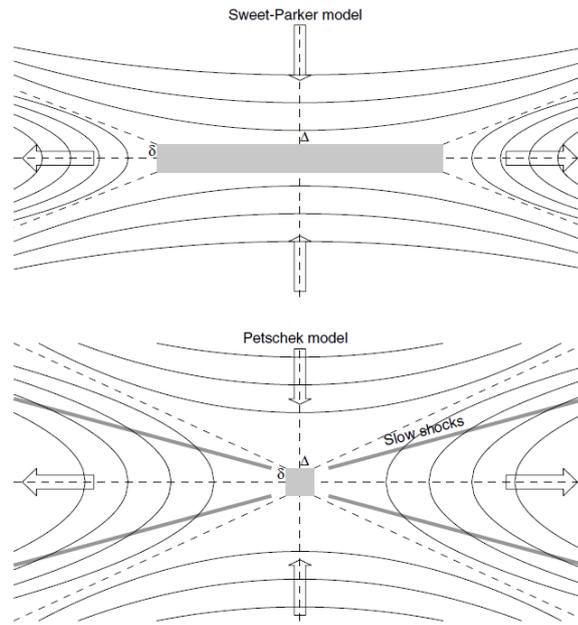


Figure 1.8: The geometry of the Sweet – Parker (top) and Petschek (bottom) reconnection model. The shape of the diffusion region is a long thin sheet ( $\Delta \gg \delta$ ) in the case of Sweet – Parker model, it is much more compact ( $\Delta \approx \delta$ ) in Petschek model, which also considers slow mode MHD shocks in the outflow region [5].

they collapse and transfer their energy to the surrounding plasma environment. Alfvén waves were observed for the first time in 2008 by the SOT instrument on the HINODE probe.

# Chapter 2

## Physics of Plasma

### 2.1 Concept of Plasma

Only 4 % of the mass in the universe consists of observable baryonic mass. However, it is commonly stated in the literature that 99 % of the universe is in the state of plasma [10]. The most common objects consisting of plasma are stars, nebulae and comet tails, but also the solar and stellar winds, interplanetary and intergalactic space. The state of plasma is naturally found on the Earth only in rare cases, such as lightning or flames.

In order to achieve the physical state of plasma, several conditions have to be met. A plasma is a partially or fully ionized gas with macroscopic electrical neutrality, consisting only of electrons, ions and neutral atoms. The aspect of plasma can be expressed by the following conditions (for the electron-proton plasma) [1]:

- a) The mean force of near interactions is very low compared to distant collective interaction of particles

$$\langle F_{near} \rangle \ll \langle F_{distant} \rangle,$$

- b) the number of particles in the so-called Debye sphere is large

$$\frac{1}{n\lambda_D^3} \ll 1,$$

where  $n$  denotes the particle density and  $\lambda_D$  is the Debye length,

- c) the thermal kinetic energy is  $KE$  is much greater than potential energy  $PE$

$$KE \gg PE, \quad \frac{3}{2}k_B T \gg \frac{e^2}{4\pi\epsilon_0\lambda_D},$$

where  $k_B$  is the Boltzmann constant,  $T$  is the plasma temperature,  $e$  is the elementary charge and  $\epsilon_0$  is permittivity of vacuum.

The combination of the above-mentioned conditions means that plasma is a sufficiently diluted and hot gas with a characteristic length of the plasma system being much greater than the Debye length.

## 2.2 Classification and Attributes of Plasma

### 2.2.1 Types of Plasmas

There are various types of plasmas in the universe, ranging from very high density inside a white dwarf stars ( $n \sim 10^{36} \text{ m}^{-3}$ ) to a very low density plasma in interstellar space ( $n \sim 10^6 \text{ m}^{-3}$ ). The various types of plasmas are displayed in Figure 2.1 as a function of temperature and density. In the very high temperature state (*relativistic plasma*), the plasma must be treated relativistically, whereas high-density plasma must be treated as a quantum-mechanical degenerate Fermi gas (*degenerate plasma*). The rest of the plasmas are divided into either *classical plasma* or *neutral gas* depending on whether the b) plasma condition is satisfied. The various parameters of certain types of plasma are shown in Table 2.1.

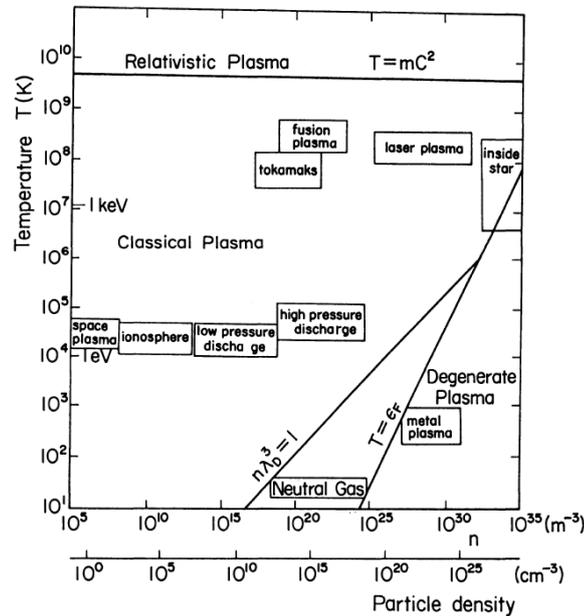


Figure 2.1: Types of plasmas in a logarithmic temperature-density parameter space [10].

$n [\text{m}^{-3}]$	$T_e [\text{eV}]$	$\lambda_D [\text{m}]$	$n\lambda_D^3$	<b>description</b>
$10^6$	1	7.4	$4 \cdot 10^8$	Interstellar gas
$10^{12}$	$10^2$	$7.4 \cdot 10^{-2}$	$4 \cdot 10^8$	Solar corona
$10^{18}$	$10^2$	$7.4 \cdot 10^{-5}$	$4 \cdot 10^5$	Hot plasma
$10^{19}$	$10^3$	$7.4 \cdot 10^{-5}$	$4 \cdot 10^6$	Hot plasma
$10^{20}$	$10^4$	$7.4 \cdot 10^{-5}$	$4 \cdot 10^7$	Thermonuclear plasma
$10^{26}$	$10^4$	$7.4 \cdot 10^{-8}$	$4 \cdot 10^4$	Laser plasma

Table 2.1: Density, temperature, Debye length, and plasma parameter of selected plasma types [10].

### 2.2.2 Debye Shielding

It is well known that Coulomb potential is of long-range character in the sense that it extends over an infinite distance in vacuum or unpolarized medium. However, in plasma, the charge of any charged particle induces an electric polarization because it attracts/repels the opposite/same sign charge. This polarization change tends to cancel the charge of the particle and subsequently restricts the effect of its electric potential to the finite range. This range is characterized by the above mentioned Debye length of  $\lambda_D$ . The restriction of the effective range of the electric potential is called *Debye shielding* [10]. Debye length is defined as

$$\lambda_D \equiv \sqrt{\left(\frac{\epsilon_0 k_B T}{n_e e^2}\right)}, \quad (2.1)$$

where  $n_e$  denotes the electron density. As we see from the Equation 2.1, Debye length has the dimension of length [m].

### 2.2.3 Plasma Oscillations

Let us consider a spatially uniform plasma that is free from magnetic fields. We also assume that an electron density perturbation is produced and it is dependant only on coordinate  $x$  and time  $t$ . Such a perturbation is therefore made by a displacement in the  $x$ -the direction of the electron flow relative to the ion flow. As a result, the electric field is produced and acts to bring the electron fluid back to its original position. Because of the electron inertia, the electron fluid passes the original position and must reverse itself once again. These oscillations are called the *electron plasma frequency* and are a function of the electron mass.

The only force that is applied to the electron fluid is electrostatic force  $F$  which can be written as

$$F = -eE = -\frac{ne^2}{\epsilon_0}x \quad (2.2)$$

where  $E$  the intensity of the electric field. By substituting the relation for force  $F = m_e \frac{d^2x}{dt^2}$  into the equation, we get the equation of harmonic oscillator

$$m_e \frac{d^2x}{dt^2} + \frac{ne^2}{\epsilon_0}x = 0. \quad (2.3)$$

Knowing the solution of this equation, analogically the *plasma frequency*  $\omega_{pe}$  is

$$\omega_{pe} = \sqrt{\frac{ne^2}{m_e \epsilon_0}}. \quad (2.4)$$

It is one of the most important parameters of plasma widely used in plasma diagnostics. Important outcome is that in a scenario, in which we let wave with a certain frequency pass through plasma environment, in order to pass through it, its frequency must higher than the plasma frequency  $\omega > \omega_{pe}$  otherwise it would be blocked. This fact lets us measure the *plasma density*  $n$ .

From equation (2.4) we can see, that plasma frequency depends solely on the concentration of electrons  $n$ , therefore by using the mass of the proton  $m_i$  instead of the electron mass, we get the *ion plasma frequency* (frequency of the ion acoustic wave)

$$\omega_{pi} = \sqrt{\frac{ne^2}{m_i \epsilon_0}}. \quad (2.5)$$

### 2.2.4 Cyclotron Motion

Let us consider a charged particle in the case of no electric field, therefore  $E = 0$ . The only force acting on particle is perpendicular to the electric field, the Lorentz force, therefore it does no work on the particle. This means that the perpendicular component of the kinetic energy is conserved,

$$\frac{1}{2}mv_{\parallel}^2 = \text{const.}, \quad (2.6)$$

where  $v = |\mathbf{v}|$ . If the magnetic field is uniform ( $\nabla \mathbf{B} = \vec{0}$ ), then the acceleration parallel to the magnetic field vanishes,

$$v_{\parallel} = \text{const.}, \quad (2.7)$$

where  $v_{\parallel} = \mathbf{b} \cdot \mathbf{v}$  with  $\mathbf{b}$  is the unit vector along  $\mathbf{B}$ ;  $\mathbf{b} = \mathbf{B}/B$ . Along the magnetic field, the particle moves at constant speed. Combining the Equations 2.6 and 2.7, we get that the perpendicular particle motion is also conserved:

$$\frac{1}{2}v_{\perp}^2 = \text{const.}, \quad (2.8)$$

where  $v_{\perp} = \sqrt{v^2 - v_{\parallel}^2}$ . Subsequently, the particle undergoes a circular motion at a constant speed  $v_{\perp}$  around the magnetic field lines. This motion is called the *cyclotron motion*, or *gyration*. Its angular frequency denoted as  $\omega_c$  is given as

$$\omega_c = \frac{|q|B}{m} \quad (2.9)$$

and is called the *cyclotron*, *gyrocyclotron* or *Larmor frequency*. The radius of the cyclotron motion is called the *Larmor radius*  $\rho_L$  and is given by

$$\rho_L = \frac{v_{\perp}}{\omega_c}. \quad (2.10)$$

The direction of the cyclotron motion depends on the charge of the particle, as shown on Figure 2.2.

## 2.3 Equations Describing Plasma Phenomena

Descriptions above were focused on the individual behavior of a particle in plasma. However, we cannot describe whole plasma with this approach, as plasma is a system of an enormous amount of particles that cannot be tracked individually, therefore the statistical approach is used for its description.

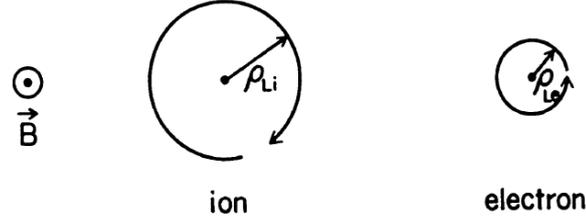


Figure 2.2: Direction of cyclotron motion for an electron and ion [10].

### 2.3.1 Kinetic Description

The kinetic description treats the plasma as a phase space continuum. In general, the plasma can be described as N-particle distribution function  $f_N(z_1, z_2, z_3, \dots, z_N, t)$ , where  $z_1 = (\mathbf{r}_1, \mathbf{p}_1)$ ,  $z_2 = (\mathbf{r}_2, \mathbf{p}_2)$ , ..., and where  $\mathbf{r}$  and  $\mathbf{p}$  are the mean values of corresponding particle position and velocity. Due to Liouville theorem, the function  $f_N$  fulfills the continuity equation in the  $6N$  dimensional space

$$\frac{\partial f_N}{\partial t} + \sum_{l=1}^N \frac{\partial}{\partial z_l} (z_l f_N) = 0, \quad (2.11)$$

where  $z_l$  is the time derivative of  $z_l$ . Using the Hamiltonian equations

$$\dot{\mathbf{r}}_l = \frac{\partial H}{\partial \mathbf{p}_l}, \quad (2.12)$$

$$\dot{\mathbf{p}}_l = -\frac{\partial H}{\partial \mathbf{r}_l}, \quad (2.13)$$

where  $H$  is the Hamiltonian of the plasma, we can write

$$\frac{\partial f_N}{\partial t} + [f_N, H] = 0, \quad (2.14)$$

where  $[f_N, H]$  is the Poisson bracket. Furthermore, the Equation 2.14 can be transformed into a chain of equations by integration over part of the variables. This process is quite difficult and can be found in the book of Achiezer et al. (1974). Subsequently, the plasma can be described by distribution function  $f(\mathbf{r}, \mathbf{v}, t)$ . The distribution gives the number of particles that are present in a unit volume of the 6-dimensional phase space defined by spatial coordinate  $\mathbf{r}$  and velocity coordinate  $\mathbf{v} = \mathbf{p}/m$  (non-relativistic case). This means that the distribution function is the solution of the so called *Boltzmann equation*

$$\frac{\partial f(\mathbf{r}, \mathbf{v}, t)}{\partial t} + \mathbf{v} \cdot \frac{\partial f(\mathbf{r}, \mathbf{v}, t)}{\partial \mathbf{r}} + \frac{\mathbf{F}}{m} \cdot \frac{\partial f(\mathbf{r}, \mathbf{v}, t)}{\partial \mathbf{v}} = \left( \frac{\partial f(\mathbf{r}, \mathbf{v}, t)}{\partial t} \right)_{coll} \quad (2.15)$$

where  $m$  is the mass of particle,  $\mathbf{F}$  is the general Lorentz force in the form of

$$\mathbf{F} \equiv m \frac{d\mathbf{v}}{dt} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \quad (2.16)$$

where  $q$  is the charge of particle and  $\mathbf{E}$  and  $\mathbf{B}$  are the electric field and magnetic induction. The term on the right side of the Equation 2.15 stands for the effects of particle collisions.

Because plasma contains many types of particles (such as electrons, protons, ions, or neutrons), the Boltzmann equation requires individual solution for every component of the plasma system that will correctly address the collisions between these components. However, in order to perform many tasks in our simulations in effective time, some approximations are made and we only solve the Boltzmann equation for electrons and ions. Furthermore, when the collisional term is very low (example being if the plasma frequency  $\omega_{pe}$  is much greater than the collision frequency  $\nu_c$ ), such plasma is called collisionless and for its description the *Vlasov equation* is used

$$\frac{\partial f(\mathbf{r}, \mathbf{v}, t)}{\partial t} + \mathbf{v} \cdot \frac{\partial f(\mathbf{r}, \mathbf{v}, t)}{\partial \mathbf{r}} + \frac{\mathbf{F}}{m} \cdot \frac{\partial f(\mathbf{r}, \mathbf{v}, t)}{\partial \mathbf{v}} = 0. \quad (2.17)$$

Finally, for a full set of equations describing a plasma behavior we need the *Maxwell equations*

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \quad \nabla \cdot \mathbf{E} = \frac{\rho_e}{\epsilon_0}, \quad (2.18)$$

$$\nabla \times \mathbf{B} = \mu_0 \left( \mathbf{j} + \epsilon_0 \frac{\partial \mathbf{E}}{\partial t} \right), \quad \nabla \cdot \mathbf{B} = 0 \quad (2.19)$$

where  $\mathbf{j}$  is the electric current density and  $\rho_e$  is the charge density. These can be expressed as

$$\mathbf{j} = \sum_{\alpha} e_{\alpha} \int \mathbf{v} f_{\alpha} d^3 v, \quad (2.20)$$

$$\rho_e = \sum_{\alpha} e_{\alpha} \int f_{\alpha} d^3 v, \quad (2.21)$$

where the index  $\alpha$  corresponds to individual plasma components [1].

### 2.3.2 Magnetohydrodynamic Description

Many tasks in astronomical plasma are way too complex for the effective usage of the kinetic model. In order to correctly describe the plasma, we do not really need to know the distribution functions of plasma particles – in these cases, the knowledge of macroscopic quantities, such as the mean plasma velocity or pressure, is sufficient. This allows us to integrate the kinetic equations. Thus the equations with the macroscopic quantities (called the *magnetohydrodynamic equations*, abbreviated as *MHD*) can be obtained as the moments of the Boltzmann equation  $\frac{Df}{Dt}$ :

$$\int \left[ \frac{Df}{Dt} \right] d^3 v, \quad (2.22)$$

$$\int m \mathbf{v} \left[ \frac{Df}{Dt} \right] d^3 v, \quad (2.23)$$

$$\int \frac{1}{2} m v^2 \left[ \frac{Df}{Dt} \right] d^3 v. \quad (2.24)$$

Integrating the first moment 2.22 yields us the mass conservation equation

$$\frac{D\rho}{Dt} \equiv \frac{\partial\rho}{\partial t} + \mathbf{v} \cdot \nabla\rho = 0, \quad (2.25)$$

where the density is taken as  $\rho = \int f d^3v$ . By integrating the second moment 2.23, we get the momentum conservation equation

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \mathbf{j} \times \mathbf{B} + \nabla \cdot \mathbf{S} + \mathbf{F}_g, \quad (2.26)$$

where  $p$  is the plasma pressure,  $\mathbf{j}$  is the current density,  $\mathbf{B}$  is the magnetic induction,  $\mathbf{S}$  is the viscous stress tensor and  $\mathbf{F}_g$  is the gravitational force density. Integrating the third moment 2.24 describes the internal energy conservation

$$\rho \frac{dU}{dt} + p\nabla \cdot \mathbf{v} = \nabla \cdot (\kappa \cdot \nabla T) + (\eta_e \cdot \mathbf{j}) \cdot \mathbf{j} + Q_v - Q_T, \quad (2.27)$$

where

$$U = \frac{P}{(\gamma - 1)\rho} \quad (2.28)$$

is the internal energy per unit mass,  $\kappa$  is the thermal conductivity tensor,  $T$  is the temperature,  $Q_v$  is the heating by viscous dissipation,  $\kappa$  is the polytropic coefficient and  $Q_T = \rho^2 Q(T)$  is the radiative energy loss,  $Q(T)$  is a function describing the temperature variation due to radiative loss in the optically thin approximation [1].

### 2.3.3 Collisional and Anomalous Resistivity

The electric conductivity of a plasma depends on the *collisional resistivity*  $\eta_e$ . More frequent the collisions are, the higher the electric resistivity is. This can be expressed as

$$\eta_e = \frac{m_e v_{ei}}{n_e e^2}, \quad (2.29)$$

where  $v_{ei}$  is the electron-ion collision frequency and  $n_e$  is the electron plasma density. However, in many astrophysical cases the plasma is very hot and diluted, therefore the collisions tend to be rare, which makes the value of collisional resistivity nearly zero. Various types of plasma waves are generated due to occurring plasma instabilities of collisionless plasma – some of these possess an electric field and can interact with electrons. These electron-wave interactions are the cause of so called *anomalous resistivity*, which is an analogy of the Equation 2.29 written as

$$\eta_e = \frac{m_e v_{\text{eff}}}{n_e e^2}, \quad (2.30)$$

where  $v_{\text{eff}}$  is the effective collision frequency. This value can be severely exceed the value collisional resistivity [1].

### 2.3.4 Induction Equation

By combining the Maxwell equations and using the vector formulas, the *induction equation* is formulated:

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B}) + \eta \nabla^2 \mathbf{B} \quad (2.31)$$

where  $\eta = \eta_e / \mu_0$  is the magnetic diffusivity. In order to get an appropriate approximation of the equation, we define the so called Reynolds number that compares terms on the right side of Equation 2.31

$$R_m = \frac{\nabla \times (\mathbf{v} \times \mathbf{B})}{\eta \nabla^2 \mathbf{B}} \approx \frac{\frac{v_0 B_0}{L_0}}{\frac{\eta B_0}{L_0^2}} = \frac{L_0 v_0}{\eta} \quad (2.32)$$

where  $v_0$  is the characteristic plasma velocity and  $L_0$  is the characteristic length scale. We consider 2 extreme regimes of the induction equation. If the processes are characterized by small plasma velocities ( $v_0 \rightarrow 0 \implies R_m \rightarrow 0$ ), the induction equation has the form of diffusion equation

$$\frac{\partial \mathbf{B}}{\partial t} = \eta \nabla^2 \mathbf{B}. \quad (2.33)$$

The other possibility is the case of collisionless plasma with  $\eta \approx 0$ , i.e. for  $R_m \gg 1$ , then the equation becomes

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B}). \quad (2.34)$$

This equation leads to the conservation of magnetic field flux as it moves with plasma – it is widely called as the concept of frozen flux [1].

### 2.3.5 Plasma- $\beta$ Parameter

It is defined as a ratio between the kinetic and magnetic pressure, written as [7]

$$\beta = \frac{p_{kin}}{p_{mag}} = \frac{nk_B T}{\frac{B^2}{2\mu_0}} = \frac{2}{\gamma} \left( \frac{c_s}{c_A} \right)^2 \quad (2.35)$$

where  $\gamma$  is adiabatic coefficient,  $c_s$  is the sound speed and  $c_A$  denotes the Alfvén's speed defined as

$$c_A = \frac{B}{\sqrt{\mu_0 \rho}}. \quad (2.36)$$

Another formulation of the plasma- $\beta$  parameter is [5]

$$\beta = \frac{2\xi n_e k_B T}{B^2 / 8\pi} \approx 0.07 \xi n_9 T_6 / B_1^2, \quad (2.37)$$

where  $\xi = 1$  is the ionization fraction for the corona (it is 0.5 for the photosphere),  $B_1 = B/10$  G is the magnetic field strength,  $n = n_e/10^9$  cm<sup>-3</sup> is the electron density, and  $T_6 = T/10^6$  K is the temperature. If the value of  $\beta > 1$ , the plasma dynamics of the chosen environment are governed by thermal kinetic pressure, if  $\beta < 1$ , it is governed by magnetic pressure.

## 2.4 Double Layer Phenomena

Double layers were discovered by Langmuir (who called them "double sheaths") in the 1920s in his experiments with low-pressure discharges. Eventually, they were observed by Tonks (1931), Schönhuber (1958), Crowford and Freeston (1963). The double layer can be described as two thin and close regions of opposite charge excess that give rise to a potential drop, and therefore an electric field, across the layer. They were theoretically predicted to exist in the cosmic environment by Alfvén [12], however, their proof of existence in Earth's Magnetosphere did not come until the devices on satellites made measurements two decades later [13]. Understanding of the phenomena is a vital part in the case of our later mentioned simulations, as we simulate two plasma regions divided by thin region.

### 2.4.1 General Description of Double Layers

Electrons and ions in plasma counterflow with velocities  $v_{e0}$  and  $v_{i0}$ , respectively. As a result of this counterflow, instabilities are set up within the current. The classic double layer (further noted as DL) is therefore an electrostatic structure that is created within the current with the ability to sustain a significant net potential difference. Its thickness can be described in the Debye length scale and tends to be very small. The DL is not neutrally charged as it must contain at least two layers of opposite net charge associated with an internal electric field. The structure is displayed on Figure 2.3. Four populations of particles are required to produce the space-charge distribution that is needed to sustain the double layer potential. These are

- the current-carrying streaming electrons
- the current-carrying streaming ions
- trapped electrons on the downstream side of streaming electrons
- trapped ions on the downstream side of streaming ions.

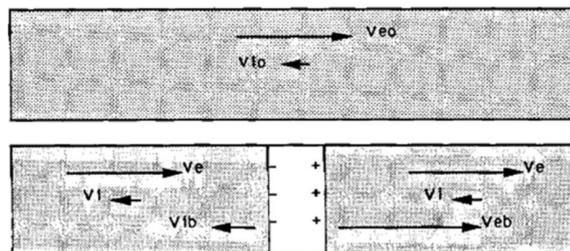


Figure 2.3: Top: condition for the formation of a double layer. Bottom: A double layer [13].

For a better understanding, Figure 2.4 displays the potential profile and the contribution of the four-particle populations to the space-charge distribution to support the double layer potential. Because of the acceleration of the streaming particles by the DL potential,

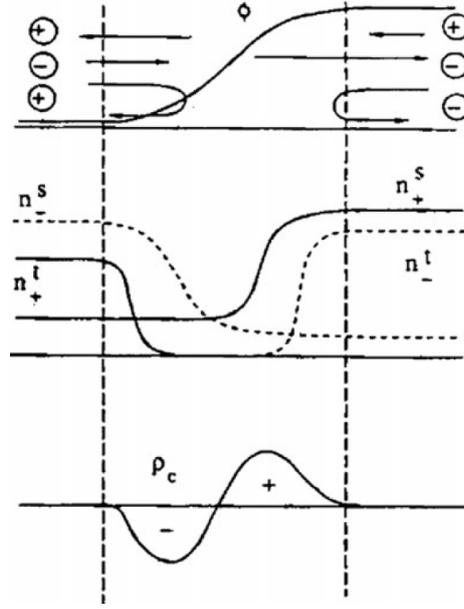


Figure 2.4: A sketch of the double layer potential  $\phi$ , the streaming and trapped particles, the space charge due to each of the four particle populations, and the net space charge required to support the double layer potential [13].

the density of streaming particles decreases on the downstream side – therefore, trapped particles must be produced to maintain the overall charge neutrality.

In an astrophysical environment, electric currents tend to be aligned along the magnetic fields, the internal electric field has a component that is parallel to the one of the magnetic field. This means that the DL is the region where ideal magnetohydrodynamics break down. This is because DL as a whole has no net charge and the surrounding plasma may be perturbed by extended presheats matching the conditions at the edges of the DL to those of undisturbed plasma. A fundamental property is that particles traversing the DL are directly accelerated by the net potential difference  $\phi_{DL}$ . Subsequently, the DL acts as an electrical load that dissipates the energy at a rate  $I\phi_{DL}$  ( $I$  is the total current passing through the DL), which is transformed to the kinetic energy of accelerated particles.

## 2.4.2 One-Dimensional DL Model

For a sense of better understanding of the phenomena, it is important to first consider a one-dimensional time-independent model with a high potential drop (Carlqvist [15]). The setup of model is following: the layer is situated between the cathode boundary at  $x = 0$  and the anode boundary at  $x = d$ ; the potential is set to  $\phi = 0$  at  $x = 0$  and  $\phi = \phi_{DL}$  at  $x = d$  (shown on Figure 2.5). Electrons and ions are emitted with zero velocity from both boundaries. Electrons and ions inside the DL are accelerated in opposite directions along the magnetic field by the electric field. The equations describing the phenomena can be found in [13]. The ratio of the current densities in DL is defined as

$$\frac{j_i}{j_e} = \left( \frac{\phi_{DL} + 2\phi_e}{\phi_{DL} + 2\phi_i} \right)^{1/2} \quad (2.38)$$

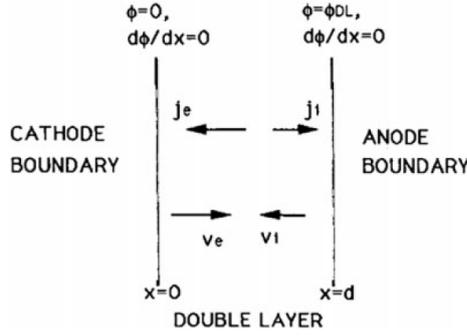


Figure 2.5: Schematic of a one-dimensional DL model [13].

where  $j_i$  and  $j_e$  are the respective current densities of ions and electrons with  $\phi_i$  and  $\phi_e$  being their respective potentials. For the nonrelativistic DL ( $\phi_{DL} \ll \phi_e \leq \phi_i$ ), the Equation 2.38 is reduced to (Carlqvist [16])

$$\frac{j_i}{j_e} \approx Z^{1/2} \left( \frac{m_e}{m_i} \right)^{1/2} \quad (2.39)$$

where  $Z$  denotes the ion charge and  $m_i$ ,  $m_e$  are the particle masses for ions and electrons respectively. This equation is referred to as the Langmuir condition. For the relativistic DL case ( $\phi_{DL} \gg \phi_i \geq \phi_e$ ), the Equation 2.38 reduces to

$$\frac{j_i}{j_e} = 1 - \frac{\phi_i}{\phi_{DL}} + \frac{\phi_e}{\phi_{DL}} = 1. \quad (2.40)$$

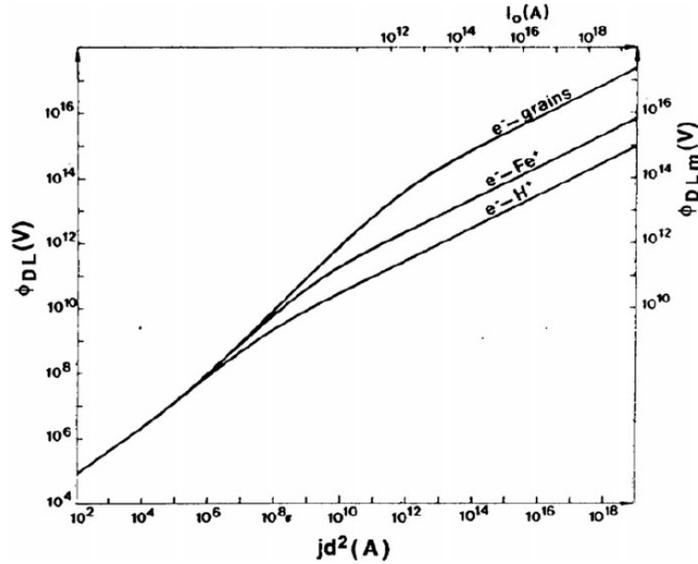


Figure 2.6: Potential drop  $\phi_{DL}$  as a function of  $jd^2$  for double layers consisting of electrons and protons, electrons and singly ionized iron ions, and electrons and dust grains of mass  $m_g = 10^{-20}$  kg and positive charge  $10^2 e$ . The graph also shows the maximum potential drop  $\phi_{DLm}$  as a function of the current  $I_0$  for relativistic double layers [13].

The potential drop of the double layer is defined a function of the total current density  $j = j_i + j_e$  and the thickness of the layer  $d$ . For nonrelativistic double layers, numeric

integration by Langmuir 1929 yields

$$\phi_{\text{DL}} = \left\{ \frac{9}{4\epsilon_0 C_2} \left( \frac{m_e}{2e} \right)^{1/2} \left[ 1 + \left( Z \frac{m_e}{m_i} \right) \right]^{-1} j d^2 \right\}^{2/3} \quad (2.41)$$

where  $C_2$  is a constant having the value 1.865 (Raadu [17]). For relativistic double layers the potential drop is defined as

$$\phi_{\text{DL}} = \left( \frac{\phi}{4\epsilon_0 c} j d^2 \right)^{1/2} \quad (2.42)$$

where  $c$  is the light velocity. As we see, the product of both the nonrelativistic and relativistic double layers depends on the  $j d^2$ . The potential drop  $\phi_{\text{DL}}$  is shown as function of  $j d^2$  on Figure 2.6.

# Chapter 3

## Particle-In-Cell Code

There are two primary approaches to simulate a plasma system. The first approach is represented by MHD models that describe the plasma as a fluid with its parameters described by statistical properties. The second approach is focused around the kinetic properties of plasma, this also includes our particle-in-cell (PIC) model. Plasma has been studied using PIC codes since the 1950s [22]. The code we will use is one of the most advanced PIC codes nowadays. It is called TRISTAN (TRI-dimensional STANford code). As the name suggests, it is three-dimensional, but also fully electromagnetic and relativistic code. It was developed by Buneman & Storey [23] for planet magnetosphere simulations and eventually was published by Matsumoto & Omura [25].

### 3.1 Fundamentals of the PIC Method

The kinetic description of a plasma describes the system via electromagnetic fields and particle distribution functions, as we described in the chapter before. The Equation 2.17 (Boltzmann equation) or Equation 2.26 (Vlasov equation) for the case of collisionless plasma, they are the starting point of a plasma simulation model, in which the fields are calculated using the Maxwell Equations 2.18 and 2.19. Kinetic effects on particles are calculated using Equation 2.16. The fields interpolated on the grid and the particles are the bottom line of the method – their solution is made as plasma superparticles in a four-part established loop (Figure 3.1).

#### 3.1.1 Mathematical Formulation of PIC

The PIC method can be understood as a representation of the distribution function of each particle type by a superposition of moving elements that represent a cloud of physical particles. We refer to this cloud of particles as a *superparticle*. The mathematical formulation of the PIC method is obtained by assuming that the distribution function of each types of particle is given by superposition of superparticles:

$$f_s(\mathbf{x}, \mathbf{v}, t) = \sum_p f_p(\mathbf{x}, \mathbf{v}, t). \quad (3.1)$$

As we stated above, each element represents a large number of physical particles that are near each other in the phase space. Therefore the choice of superparticles is made

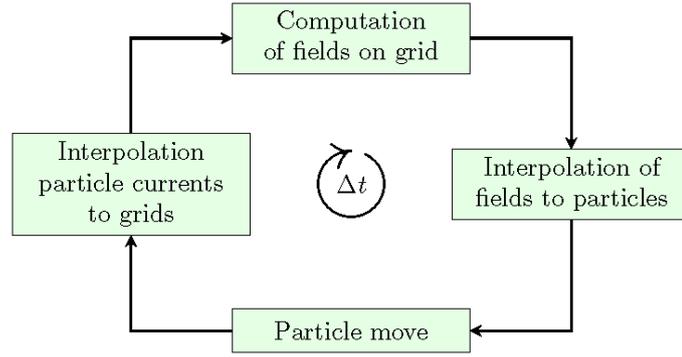


Figure 3.1: The computational steps of the general particle-cell-cell method [20].

to be physically meaningful (i.e. to represent a group of particles that are condensed near each other) and also mathematically convenient (i.e. to allow the derivation of a manageable set of equations) [18]. This is achieved by assigning a specific functional form to each computational particle for its distribution. This functional form consists of several free parameters whose time evolution will determine the numerical solution of the Vlasov equation 2.26. In the standard PIC methods, the choice is made to have two free parameters in the functional space for each spatial dimension.

### 3.1.2 Implementation of the Equations Using the Leap-Frog Algorithm

The PIC method uses the so-called *leap-frog algorithm*. This algorithm is based on staggering the time levels of the velocity and spatial position by half time step and calculating the physical properties by the finite difference method:  $\mathbf{x}_p(t = n\Delta t) \equiv \mathbf{x}_p^n$  and  $\mathbf{v}_p(t = [n + 1/2]\Delta t) \equiv \mathbf{v}_p^{n+1/2}$  (index  $p$  denotes that we are addressing the respective velocity and position of a particle). This describes the update of position from time level  $n$  to time level  $n + 1$  using the velocity at the mid-point  $\mathbf{v}_p^{n+1/2}$ . Similarly, the update of the velocity from time level  $n - 1/2$  to time level  $n + 1/2$  uses the mid-point position  $\mathbf{x}_p^n$ . The leap-frog process is displayed on Figure 3.2. The electric field  $\mathbf{E}$  and magnetic field  $\mathbf{B}$  must also be leap-frogged in time. For each spatial coordinate, this is given by the interpolation of the adjacent cells for the electric field  $\mathbf{E}$  with indexes  $(i', j', k')$  in the same position as the magnetic field  $\mathbf{B}$  with the indexes  $(i, j, k)$  in Yee lattice [21]

$$e_x(i, j, k) = \frac{e_x(i' - 1, j', k') + e_x(i', j', k')}{2}, \quad (3.2)$$

$$e_y(i, j, k) = \frac{e_y(i', j' - 1, k') + e_y(i', j', k')}{2}, \quad (3.3)$$

$$e_z(i, j, k) = \frac{e_z(i', j', k' - 1) + e_z(i', j', k')}{2}. \quad (3.4)$$

The same process applies to the calculation of the magnetic field  $\mathbf{B}$ . We will discuss the field and particle updates specifically in the TRISTAN section.

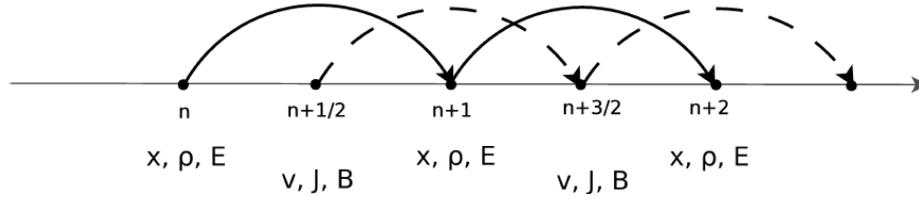


Figure 3.2: Visual representation of the leap-frog algorithm [18].

### 3.1.3 Stability of the Simulation

Explicit PIC is subject to stability conditions. In case they are exceeded, the energy of the simulation diverges, rendering the results meaningless.

- **Stability of the particle mover;**  $\omega_{pe}\Delta t < 2$ ; This is derived from the leap-frog solution of the velocity of the particle. Exceeding this condition leads to exponentially growing numerical frequency, which has no physical equivalence. The solution, therefore, becomes unstable with the particles heating unboundedly and quickly. Simulation usually fails within a few time steps.
- **Stability due to explicit time differencing of the field equations;**  $\Delta t < \Delta x/c$ ; This condition is called the Courant-Friedrich-Levy (CFL) condition [24]. The CFL condition states that the time step must not exceed the time taken by the signal to travel one cell. In the case of the Maxwell equations, the characteristic speed of the signal is the speed of light  $c$ .

However, these are not the only conditions that ensure the stability of the simulation, i.e. the finite grid instability. Addressing all of the conditions would exceed the idea of this paper. They are explained in full detail in Hockney & Eastwood (1988).

## 3.2 TRISTAN

### 3.2.1 Field Update

In order to preserve the space-time symmetries, all finite-difference implementations of differential operators in Equations 2.18 and 2.19 must be space- and time-centered [21]. Therefore, as mentioned in Chapter 3.1.2,  $\mathbf{B}$  data must be leap-frogged over  $\mathbf{E}$  data in time. At this moment, the data is staggered in space. These calculations of averages are made using the Equations 3.2, 3.3 and 3.4. TRISTAN uses model relative scales, in which  $\epsilon_0 = 1$  and hence  $\mu_0 = 1/c^2$ . This means that the model records  $\mathbf{E}$  with components  $e_x, e_y, e_z$  and components  $b_x, b_y, b_z$  of  $c\mathbf{B}$ . This makes the fundamental symmetry  $e \longleftrightarrow b$  of Maxwell Equations. TRISTAN uses a rectangular cubic grid with coordinate units  $\Delta x = \Delta y = \Delta z = 1$  and time discretization  $\Delta t = 1$ . The simulation value of  $c$  is usually 0.5.

For a better sense of understanding the TRISTAN code solutions of Equations 2.18 and 2.19, the time change of magnetic field in one time step is [26]

$$b_x^{\text{new}}(i, j, k) = b_x^{\text{old}}(i, j, k) + c\Delta t \left[ \frac{e_y(i, j, k+1) - e_y(i, j, k)}{\Delta z} - \frac{e_z(i, j+1, k) + e_z(i, j, k)}{\Delta y} \right], \quad (3.5)$$

$$b_y^{\text{new}}(i, j, k) = b_y^{\text{old}}(i, j, k) + c\Delta t \left[ \frac{e_z(i+1, j, k) - e_z(i, j, k)}{\Delta x} - \frac{e_x(i, j, k+1) + e_x(i, j, k)}{\Delta z} \right], \quad (3.6)$$

$$b_z^{\text{new}}(i, j, k) = b_z^{\text{old}}(i, j, k) + c\Delta t \left[ \frac{e_x(i, j+1, k) - e_x(i, j, k)}{\Delta y} - \frac{e_y(i+1, j, k) + e_y(i, j, k)}{\Delta x} \right]. \quad (3.7)$$

For a better numerical stability of the code, the advance of the magnetic field is computed in two sub-time steps  $\Delta t = \frac{1}{2}$ . This is because  $\mathbf{B}$  must be updated in two half steps so that it is available at the same times as  $\mathbf{E}$  for the particle update. The advance of the electric field follows the half advances of the magnetic field in the form of

$$e_x^{\text{new}}(i, j, k) = e_x^{\text{old}}(i, j, k) + c\Delta t \left[ \frac{b_y(i, j, k-1) - b_y(i, j, k)}{\Delta z} - \frac{b_z(i, j+1, k) + b_z(i, j, k)}{\Delta y} \right], \quad (3.8)$$

$$e_y^{\text{new}}(i, j, k) = e_y^{\text{old}}(i, j, k) + c\Delta t \left[ \frac{b_z(i-1, j, k) - b_z(i, j, k)}{\Delta x} - \frac{b_x(i, j, k-1) + b_x(i, j, k)}{\Delta z} \right], \quad (3.9)$$

$$e_z^{\text{new}}(i, j, k) = e_z^{\text{old}}(i, j, k) + c\Delta t \left[ \frac{b_x(i, j-1, k) - b_x(i, j, k)}{\Delta y} - \frac{b_y(i-1, j, k) + b_y(i, j, k)}{\Delta x} \right]. \quad (3.10)$$

### 3.2.2 Particle Update

The time-centered, finite difference modified version of the Lorentz particle update is [19]

$$\mathbf{v}^{\text{new}} - \mathbf{v}^{\text{old}} = \frac{q\Delta t}{m} \left[ \mathbf{E} + \frac{1}{2}(\mathbf{v}^{\text{new}} + \mathbf{v}^{\text{old}}) \times \mathbf{B} \right] \quad (3.11)$$

$$\mathbf{r}^{\text{next}} - \mathbf{r}^{\text{present}} = \Delta t \mathbf{v}^{\text{new}}. \quad (3.12)$$

This way, the positions  $\mathbf{r}$  are leap-frogged over velocities  $\mathbf{v}$ . A three step procedure (Hockney & Eastwood [27]), (Birdsall & Langdon [28]) is used for the calculation of the advance of a particle:

1. The first half of an electric acceleration

$$\mathbf{v}_0 = \mathbf{v}^{\text{old}} + \frac{q\mathbf{E}\Delta t}{2m}, \quad (3.13)$$

2. The magnetic rotation

$$\mathbf{v}_1 = \mathbf{v}_0 + 2 \frac{\mathbf{v}_0 \times \mathbf{v}_0 \times \mathbf{b}_0}{1 + b_0^2} \times \mathbf{b}_0 \quad (3.14)$$

3. The second half of the electric acceleration

$$\mathbf{v}^{\text{new}} = \mathbf{v}_1 + \frac{q\mathbf{E}\Delta t}{2m}. \quad (3.15)$$

### 3.2.3 Current Decomposition

Throughout the code, linear interpolation is applied, so the fields interacting with the particle at the position  $\mathbf{r}$  are linearly weighted in a dependance on their distance  $\delta x, \delta y, \delta z$  from the closest grid point  $i, j, k$ , where

$$i = \text{round}(x), \quad j = \text{round}(y), \quad k = \text{round}(z),$$

are the rounded values and

$$\delta x = x - i, \quad \delta y = y - j, \quad \delta z = z - k,$$

denote the volume weighted values that can be used in the expression of  $\mathbf{F}^{x,y,z}$ . TRISTAN does not employ a charge density array. Instead of that, it uses a direct particle current decomposition into the electric field – this removes the cohesion between the particles and enables the parallelization of the simulation. A charge conservation scheme was proposed by Villasenor & Buneman [29]. The electric field is modified by the current  $\mathbf{J} = (j_x, j_y, j_z)$  of each particle

$$e_x(i, j, k) = e_x(i, j, k) - j_x \cdot cy \cdot cz, \quad (3.16)$$

$$e_x(i, j + 1, k) = e_x(i, j + 1, k) - j_x \cdot \delta y \cdot cz, \quad (3.17)$$

$$e_x(i, j, k + 1) = e_x(i, j, k + 1) - j_x \cdot cy \cdot \delta z, \quad (3.18)$$

$$e_x(i, j + 1, k + 1) = e_x(i, j + 1, k + 1) - j_x \cdot \delta y \cdot \delta z, \quad (3.19)$$

$$e_y(i, j, k) = e_y(i, j, k) - j_y \cdot cx \cdot cz, \quad (3.20)$$

$$e_y(i, j + 1, k) = e_y(i, j + 1, k) - j_y \cdot \delta x \cdot cz, \quad (3.21)$$

$$e_y(i, j, k + 1) = e_y(i, j, k + 1) - j_y \cdot cx \cdot \delta z, \quad (3.22)$$

$$e_y(i, j + 1, k + 1) = e_y(i, j + 1, k + 1) - j_y \cdot \delta x \cdot \delta z, \quad (3.23)$$

$$e_z(i, j, k) = e_z(i, j, k) - j_z \cdot cy \cdot cx, \quad (3.24)$$

$$e_z(i, j + 1, k) = e_z(i, j + 1, k) - j_z \cdot \delta y \cdot cx, \quad (3.25)$$

$$e_z(i, j, k + 1) = e_z(i, j, k + 1) - j_z \cdot cy \cdot \delta x, \quad (3.26)$$

$$e_z(i, j + 1, k + 1) = e_z(i, j + 1, k + 1) - j_z \cdot \delta y \cdot \delta x, \quad (3.27)$$

where

$$cx = 1 - \delta x, \quad cy = 1 - \delta y, \quad cz = 1 - \delta z. \quad (3.28)$$

### 3.2.4 Speed Limits

As we stated in Chapter 3.1.3, speed limits must be applied to maintain the stability of the simulation. If a particle were to move through several cells in one time step, it would miss available field information for each of these cells. With  $c$  set to 0.5, we ensure that particles will not go through more than half a mesh cell per time step, so the particles do not outrun the fields.

Discretizing the Maxwell equations leads to the imposition of the Courant-Friedrich-Levy (CFL) condition. This condition has to be met whenever a hyperbolic equation is discretized explicitly. It states that the time step cannot exceed the time taken by the characteristic signal to travel through one grid cell.

### 3.2.5 Boundary Conditions

For the code to function properly, several boundary conditions for cells must be applied, such as particle absorption at the boundary, periodicity, or reflection. Boundary conditions have to be applied for particles, but also fields. We differentiate the boundaries between the computing grids into two categories:

- Boundaries between the subgrids that belong to unique processors. These boundaries are set in such a way that they effectively sew the neighboring subgrids together in all three dimensions. For the sake of the continuity of the physical quantities, these boundaries cannot be changed throughout the process of simulation.
- Outer boundaries of the whole computing domain – the code uses a periodic boundary condition. These are implemented in a way that they match the studied physical situation.

The boundary data must be sent between the processors. Each processor determines where the data are sent. This is easier for the case of fields, as the data amount always stays the same. In the case of particles, the size of the sending buffers of the particle is prone to change.

Our model uses periodic boundary conditions in the  $y$  and  $z$  axis and mirror boundary conditions in the  $x$  axis. To maintain the stability, all particles in the hot plasma region of our model are removed if they exceed thermal velocity 5 times higher than that of cold plasma.

### 3.2.6 Parallelization

The Tristan model code we used is parallelized in both the particles and the fields (implemented by Benáček [20]). This allows the direct communication of multiple processors throughout the process of simulation, subsequently greatly increasing the potential of using the computational possibilities that are available. This was done using the MPI (*Message Passing Interference*) communication. This technique is based on the principle that each processor has its data memory domain right where it computes. The advantage is that it can run on supercomputers or computer clusters and use an existing Ethernet network special independent Infiniband.

### 3.2.7 Model Workflow

The simulation model is implemented in the programming language Fortran 90/95. It is divided into several files, from which the main part of model in the file `model.f90`, file `savedata.f90` resolves the calculation of various physical data we want to extract and save on the disk, `boundpt1.f90` and `boundfield.f90` address the boundary conditions, `init.f90` is for the initial conditions and `size.inc` is for the computing array configuration.

How the code functions is visually described in Figure 3.3. Firstly, the parameters

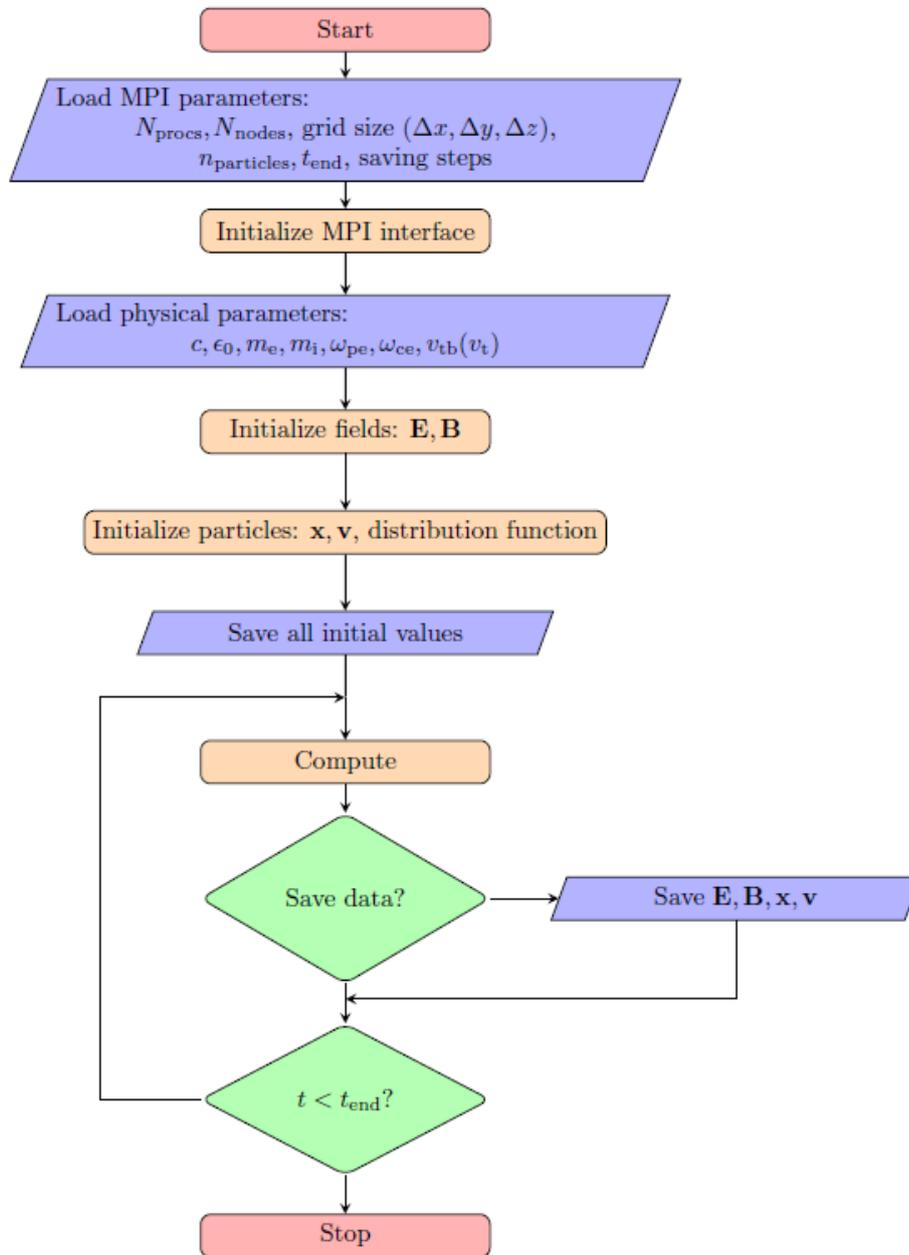


Figure 3.3: Visual representation of computational steps of the TRISTAN particle-in-cell model [20].

connected with the allocation of the particle and the field arrays are loaded in the processors. Each processor allocates some subgrid of the corresponding fields and particles. The number of processors in each of the simulated dimensions is specified in the configuration. All of the processors maintain the grid size for the full computing duration. The number of particles per processor is also given by configuration. The density is usually set to be constant, however, due to the flow of the particles from one processor, it does change.

Right after the array loading, the MPI interface is initialized. MPI automatically assigns the position in the grid of the model. This is determined by dependence on the MPI library and the server MPI configuration. It is common for the processors to be assigned in a way that the close subgrids are assigned to close processors to minimize the communication overhead between them.

When this is done, the physical parameters are loaded. From the independent parameters (that are set as initial parameters), the other dependant parameters are computed. Then, the fields are initialized. Usually, the initial electric field is set to be 0 and the magnetic field is configured according to the physical situation of studied phenomena.

The model includes subroutines for saving fields and particles. These are saved using the high-performance parallelized library Hierarchical Data Format in version 5 (HDF5), which internally uses the MPI interface. The initial field and particle information are stored before the start of the computation process.

The computing is a looped cycle that is incrementing the time until it equals the defined amount of time steps. Each step consists of the computation itself and if defined the storing of data on the disk. We can define the interval of data saving in the `savedata.f90` file.

The computing algorithm is displayed in Figure 3.4. The first half advance of the magnetic field is computed simultaneously with the magnetic field boundary conditions. After that, the particles advance, the code computes the remaining part of the magnetic field half advance and the boundary conditions for the particles. Following this, the particles are being exchanged between the processors and the current decomposition into the electric field is computed (this takes the longest in the computational step). Then, the electric field boundary conditions are computed. This marks the completion of a single computational step.

### 3.2.8 Improvements implemented in the Code

Several calculations of moments of distribution function for ions and electrons ( $s$  indicating the respective particle) were implemented into the `savedata.f90` file. This includes:

- implementation of computation for general vector and tensor fields
- implementation of computation of energy flux density, defined as

$$\mathbf{Q}_s(\mathbf{r}, t) = \int \frac{1}{2} m_s v^2 \mathbf{v} f_s(\mathbf{r}, \mathbf{v}, t) d^3 \mathbf{v} \quad (3.29)$$

- implementation of computation of stress tensor, defined as

$$\mathbf{s}_s(\mathbf{r}, t) = \int m_s \mathbf{v} \mathbf{v} f_s(\mathbf{r}, \mathbf{v}, t) d^3 \mathbf{v} \quad (3.30)$$

- implementation of computation of pressure tensor, defined as

$$\mathbf{p}_s(\mathbf{r}, t) = \int m_s \mathbf{w}_s \mathbf{w}_s f_s(\mathbf{r}, \mathbf{v}, t) d^3 \mathbf{v}, \quad (3.31)$$

where  $\mathbf{w}_s$  is the relative velocity

$$\mathbf{w}_s \equiv \mathbf{v} - \mathbf{V}_s, \quad (3.32)$$

with  $\mathbf{V}_s$  being the flow velocity of respective particles. Equations are taken from [30].

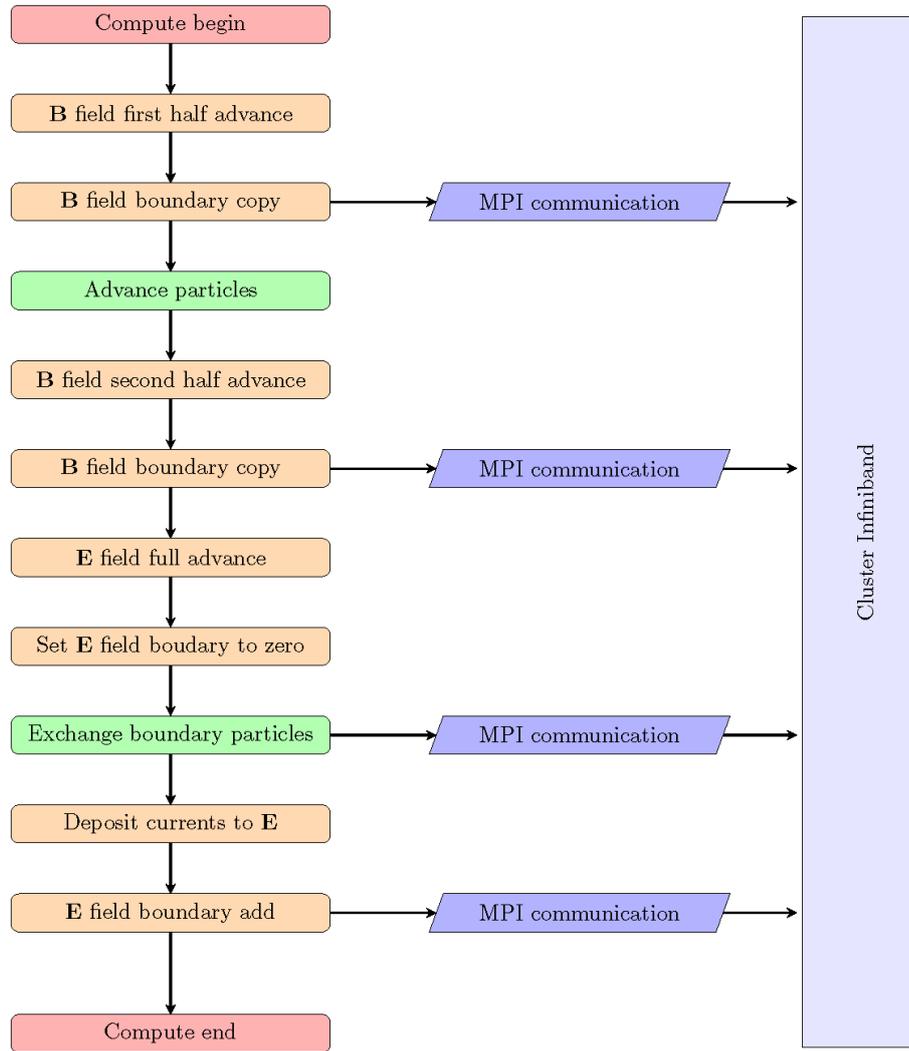


Figure 3.4: Visual representation of a detailed computational step of the model [20].

# Chapter 4

## Hot-Cold Plasma Transition

The problem of hot-cold plasma interaction is one of the most important problems. We will focus on the physical phenomena occurring in the hot-cold region, the so-called thermal fronts. The thermal front is described as an interface between the hot and cold plasmas. The temperature transition of the thermal front is usually accompanied by density transition. We will discuss its past research and compare it to our results. Its simulation is the direct aim of this thesis. We conducted two simulation cycles: at first, we focused on three simulations with different length of initial transition and observed its impact on the development of physical quantities. Then, we conducted two simulations, one with unequilibrated and one with equilibrated pressure conditions, and observed the differences between these two models.

### 4.1 Thermal Fronts

It is commonly accepted that electrons are impulsively heated in certain regions of solar flares due to magnetic field reconnection. The flares start mainly in the low corona region. The expansion of erupting hot plasma into surrounding cold plasma leads to formation of thermal fronts. This occurs along the magnetic field lines of the eruptive coronal loops. Due to the narrowness and highly dynamic state of the hot-cold plasma transition, the processes that lead to its formation are not well understood. There is also a question whether the solar transition region is not some kind of thermal front or cascade of thermal fronts [33].

### 4.2 Previous Theoretical Research

Paper *Thermal Fronts in Solar Flares* [32] is the focal point of our work, as we used a similar TRISTAN code with some improvements. The paper studied the formation of a thermal front during the expansion of hot plasma into a colder plasma that occurs in some localized regions of the flare loop.

The numerical model used for this research is configured with the grid size of  $L_x = 8\Delta$ ,  $L_y = 8\Delta$ ,  $L_z = 19000\Delta$  ( $\Delta$  indicates the grid size). It assumes a sufficiently strong magnetic field, therefore the energy transfer is considered only in the loop direction. The model

initiates a spatially homogeneous electron-proton plasma with proton-electron mass ratio of  $m_p/m_e = 100$ . The value is chosen to accelerate the generation process of the ion-sound waves. The initial number density for the particles is set 60 per grid cube for both electrons and protons. The thermal velocity of electrons in the cold region is set to be  $v_{Tec} = 0.025 c$  (with  $c$  being the speed of light). This corresponds with temperature  $T = 1.85$  MK. In the hot plasma region ( $z = 0-3500\Delta$ ), the thermal velocity is increased 10 times to a value  $v_{Teh} = 0.25 c$ , corresponding with temperature of  $T_h = 185$  MK. The plasma frequency is  $\omega_{pe} = 2\pi/t_p = 0.05$ , where  $t_p$  is configured plasma period. The time step of the simulation is set to  $\Delta t = 1$ . The magnetic field is oriented in the  $z$ -direction and its value corresponds to the electron cyclotron frequency,  $\omega_{ce} = 0.1 \omega_{pe}$ . The electron Debye length in the hot plasma region is  $\lambda_D = 2.5 \Delta$ , in the cold plasma region, it is  $\lambda_D = 0.25 \Delta$ . All of the above-mentioned input parameters are set to close approximations of real physical circumstances occurring in the thermal front. The length of the numerical system in the  $z$ -direction is 319 meters.

The temperature evolution of the simulation (Karlický [32]) is shown on Figure 4.1. The thermal front with the largest temperature difference (the main thermal front) was formed at  $\approx 4400\Delta$  within a time interval of  $\omega_{pe} = 1000 - 3500$  (corresponding with 398 plasma periods). The detailed look at the front is shown in Figure 4.2. We see, that the thermal front is spatially shifting towards the cold plasma region. The calculated velocity in the numerical model is slightly smaller than the estimated ion-sound velocity. After  $\omega_{pet} = 3500$ , the largest thermal front disappears due to the limited amount of thermal energy in the region with the hot plasma. At this point, most of the energetic electrons escaped from this region and they are further in front of the thermal front.

The results (further followed by [34–39]) can be more-less generalized in the following points:

- There are strong ion-sound waves and therefore also anomalous resistivity. They are considered to be the cause of the anomalous resistivity.
- At the location of the thermal front plasma double-layer forms, which limits the expansion of the hot plasma into the cold plasma region.
- Thermal fronts are located on the edge of plasma density depressions that are connected with ion-sound waves propagating with a velocity comparable to the ion-sound velocity.
- The thermal front in the simulation disappeared due to the limited number of particles on the left side of the front. To observe further evolution of the thermal front, a more extended numerical system is required.

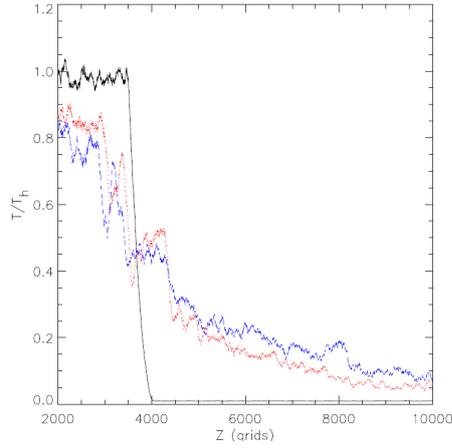


Figure 4.1: Ratio of the electron temperature  $T$  to that of the hot plasma  $T_h$  along the  $z$ -coordinate at three different states:  $\omega_{pe}t = 0$  (black solid line),  $\omega_{pe}t = 2250$  (red dashed line), and the  $\omega_{pe}t = 3500$  (blue dashed line). The main thermal front is formed at  $4400 \Delta$  [32].

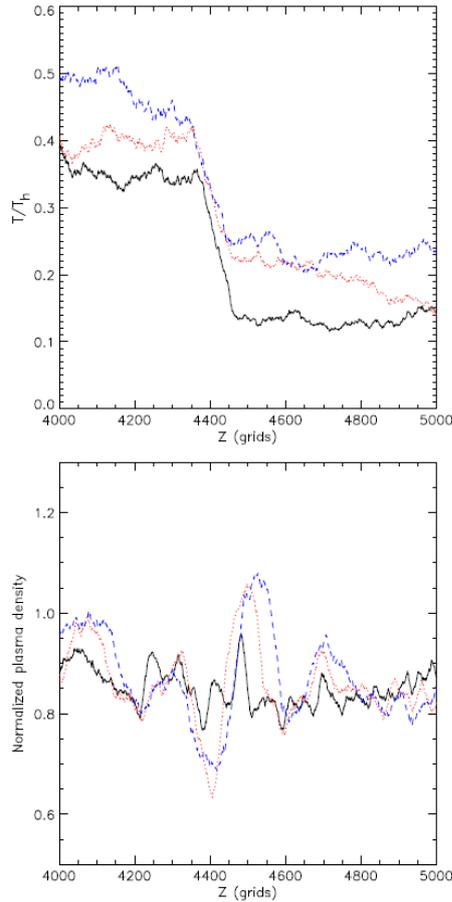


Figure 4.2: Ratio of the electron thermal energy  $T$  to that of the hot plasma  $T_h$  (upper part) and plasma density (bottom part) along the  $z$ -coordinate at three different states:  $\omega_{pe}t = 1000$  (black solid line),  $\omega_{pe}t = 1500$  (red dashed line), and the  $\omega_{pe}t = 2000$  (blue dashed line). The main thermal front is formed at  $4400 \Delta$  [32].

### 4.3 Thermal Front Formation Simulations

As a core part of our work, we conducted 5 different simulations with different configuration setups. We used modified TRISTAN code [20] including our implementations. The first 3 simulations were aimed to observe the thermal front formation with different transition region setups, the latter two aimed to show the evolution for such systems with equilibrated and unequilibrated pressure conditions. For our simulations, we used the computing power of the department computer *Crab* and its 24 processors. The scale of our model is  $L_x = 9216\Delta$ ,  $L_y = L_z = 8\Delta$ .

The thermal velocity of cold electrons is set to  $v_{\text{cold}} = 0.01297c$ , therefore we can determine its temperature  $T_{\text{cold}} = v_{\text{cold}}^2 m_e / k_B \approx 1$  MK. The corresponding Debye length of cold electrons can also be determined as  $\lambda_D = v_{\text{cold}} / \omega_{\text{pe}} \approx 0.26\Delta$ . As the hot electrons are set to have 10 times higher temperature, their temperature is  $T_{\text{hot}} \approx 10$  MK, with corresponding thermal velocity  $v_{\text{hot}} = 0.041c$  and Debye length  $\lambda_D = 0.82\Delta$ . Using the Equation 2.1 with characteristic density  $n = 10^{16} \text{ m}^{-3}$ , the estimate of Debye length for cold plasma is  $\lambda_D = 0.07$  cm. Consequently the grid length is  $\Delta = 0.27$  cm long. The simulated region of  $9216\Delta$  is estimated to be approximately 25 metres. Using the Equation 2.4, we estimate the plasma frequency of the system to be  $\omega_{\text{pe}} \approx 5.65 \cdot 10^9 \text{ s}^{-1}$ . The time step of the simulation is configured to cover 0.025 of plasma period in each time step, therefore the time step of our simulation is approximately  $\Delta t = 0.028$  ns. The simulation of the system throughout 60 000 time steps, which is equivalent to 1500 plasma periods covers approximately  $1.67 \mu\text{s}$ . The simulations were configured in a way that the left half of the grid consists of hot plasma and the right part consists of cold plasma with temperature ratio  $T_{\text{hot}}/T_{\text{cold}} = 10$ , divided by the initial transition (IT) of specified grid length. We computed three separate simulations with the IT set up to be  $0\Delta$ ,  $50\Delta$ , and  $200\Delta$  long respectively. Simulations were set up to simulate 80 000 time steps. The IT was configured to be in an approximate middle of the grid, therefore it starts at  $x = 4600\Delta$ . Independent parameters of the computed simulations are shown in Table 4.1.

The results of the simulations are shown in Figure 4.3. We interpreted the values as ratios to hot electron temperature  $T_h$ , hot electron kinetic energy  $E_{K,h}$  and initial particle density  $n_0$ . We see that the thermal front forms in all of our simulations. The temperature transition formed instantly in the simulation with IT =  $0\Delta$ , in the simulation with the IT length being  $200\Delta$ , it started forming after around 500 plasma periods. The wider the region is set up, the longer it takes for the temperature transitions to form (4.3a,b,c). Similarities in the evolution of the temperature, electric field, and electron density are apparent. A steep relative difference is created in the electric field (4.3d,e,f) and electron density (4.3g,h,i) propagation that is spread into multiple thermal fronts the longer the IT is. A very important result is that with a wider IT, fluctuations in temperature, electric field, and electron density are increasing in number and decreasing in intensity. This suggests that real physical phenomena consist of many thermal fronts instead of a single major one. Our results correspond with previously done research by Karlický [32]. For more precise results, one would need a system with more processors, allowing the configuration of lower value  $\omega_{\text{pe}}$  and also enlarging the computational grid.

<b>parameter</b>	<b>value</b>
$c$	0.5
$\epsilon_0$	1
$m_i/m_e$	1836
$\omega_{pe}$	0.025
$\omega_{ce}/\omega_{pe}$	0.19
$v_{cold}$	$0.01297 c$
$v_{hot}/v_{cold}$	$\sqrt{10}$
$n$	100
$n_{hot}/n_{cold}$	1
plasma- $\beta$ parameter (cold)	$3.03 \cdot 10^{-2}$
Initial transition [ $\Delta$ ]	0, 50, 200

Table 4.1: Independent parameters of thermal front formation model with three different lengths of transition region.

	<b>Model 1</b>	<b>Model 2</b>
<b>parameter</b>	<b>value</b>	
$c$	0.5	0.5
$\epsilon_0$	1	1
$m_i/m_e$	100	100
$\omega_{pe}$	0.025	0.025
$\omega_{ce}/\omega_{pe}$	0.10	0.10
$v_{cold}$	$0.01297 c$	$0.01297 c$
$v_{hot}/v_{cold}$	$\sqrt{10}$	$\sqrt{10}$
$n$	10	100
$n_{hot}/n_{cold}$	1	0.1
plasma- $\beta$ parameter (cold)	$8.41 \cdot 10^{-3}$	$8.41 \cdot 10^{-3}$
Initial transition [ $\Delta$ ]	20	20

Table 4.2: Independent parameters of computed models with unequilibrated pressure (Model 1) and equilibrated pressure (Model 2).

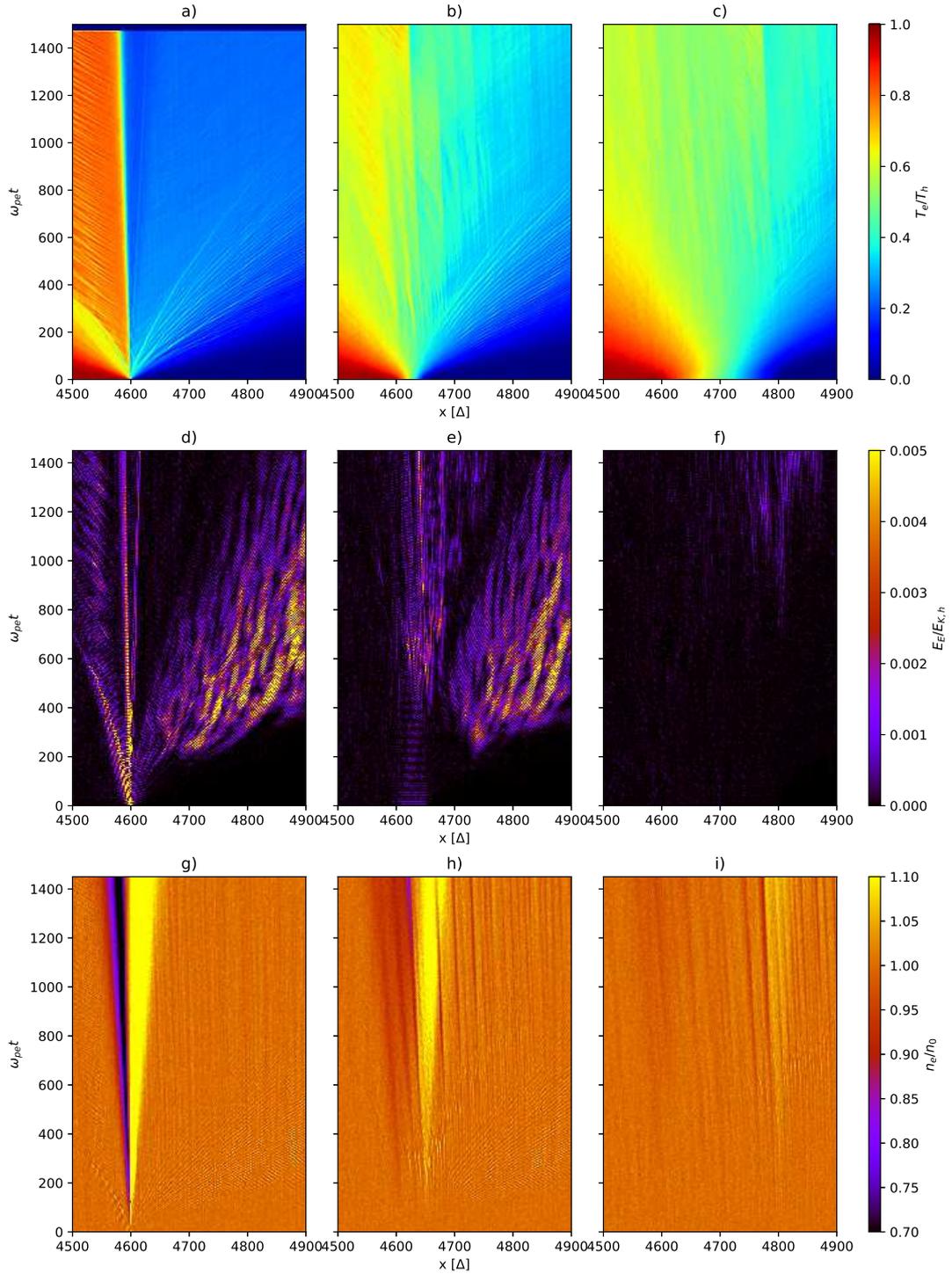


Figure 4.3: Time development of models with different initial transition length. Graphs show the development in the region of  $x = 4500 - 4900 \Delta$  with the start of IT located at  $x = 4600 \Delta$ . Left column: Simulation with  $IT = 0 \Delta$ . Middle column: Simulation with  $IT = 50 \Delta$ . Right column: Simulation with  $IT = 200 \Delta$ . a)-b)-c) Time evolution of electron temperature as ratio to hot electron temperature  $T/T_h$ . d)-e)-f) Time evolution of electric field energy as a ratio to hot electron thermal energy  $E_E/E_{K,h}$ . g)-h)-i) Time development of electron density as a ratio to the initial grid density  $n_e/n_0$ .

## 4.4 Unequilibrated and Equilibrated Pressure Models

We simulated two separate simulations marked as Model 1 and Model 2. Model 1 was configured in a way that resembles plasma with unequilibrated pressure. Model 2 was configured to demonstrate plasma with equilibrated pressure. This was achieved by setting the hot plasma region to have lower particle density than the cold plasma region. The differences in initial configuration are shown in Table 4.2. They differ in particle density per grid  $n$ , where we chose a smaller number for Model 1 to make the simulation run faster. What mattered the most and what ensured the difference in pressure equilibrium was the setting of density ratios for hot and cold particles  $n_{\text{hot}}/n_{\text{cold}}$ . The ratio is set to 1 in Model 1 and 0.1 in Model 2. The mass ratio of ions and electrons  $m_i/m_e$  is set to 100 to accelerate the effects. This also results in lower ion velocities. The temperature of hot/cold ions is the same as the temperature of hot/cold electrons. We interpreted the results of both simulations in time evolution graphs with Model 1 and Model 2 side by side. Graphs are interpreted in a way, that the initial transition is located at  $x = 0\Delta$ .

The time evolution of the electron temperature is shown on Figure 4.4a,b. Thermal front formed in the near of the initial transition in Model 1. The temperature gradient in Model 2 is much sharper with hot plasma getting colder due to having 10 times lower density than that of cold plasma. The equilibrated pressure state supports the formation of multiple temperature transitions in the hot plasma region.

The time evolution of electric field is shown on Figure 4.4c,d. The locations of electric field fluctuations are identical with the locations of formed thermal fronts, as can be also seen in the results of previous simulation 4.3a,b,c. The evolution of the electric field in Model 1 had its fluctuations centered mainly around the border of hot and cold plasma. On the other hand, electric field fluctuations of Model 2 are located in the hot plasma region with their intensity being higher than that of Model 1.

The time development of electron density is displayed in Figure 4.4e,f. This mainly shows the difference in the initial configuration of the models. In Model 1, density decreases are maintained throughout the course of simulation in the thermal front regions. In Model 2 a sharp border between low and high density is kept.

A better view can be provided by looking at the profiles of temperature and particle density of both models at  $\omega_{pe}t = 1000$  shown on Figure 4.5. The temperature profile of temperature and density of Model 1 corresponds with the results of Karlický [32] displayed in Figure 4.2. We see that in Model 2, a higher difference is kept between the temperature and density of the hot and cold plasma region, maintaining a notable border. There is no sharp density transition in Model 2, therefore we assume that the temperature transitions seen in Figure 4.4b are caused by double layers.

Figure 4.6 provides a look at the time evolution of chosen components of moments of distribution function discussed in Chapter 3.2.8. The computed tensors  $\mathbf{s}_e$  (4.6a,b) and  $\mathbf{p}_e$  have similar development in their diagonal components. For this reason, we will only show the development in the  $x$  axis, the major axis of our simulations. The time development of tensor components in Model 1 is more coherent than its counterparts in Model 2. A better look is provided in Figure 4.6c,d that shows the difference of the  $xx$ -component between the stress  $\mathbf{s}_e$  tensor and pressure tensor  $\mathbf{p}_e$ . We can see that it is similar to the temperature evolution of electrons, which corresponds with its physical description.

Figure 4.6e,f shows the time evolution of the  $x$ -component of the energy flux density  $\mathbf{Q}_e$ . We can see that a lot of energy flows at the beginning of the simulation in both models. The general energy flow is higher in Model 1. Due to the equilibrated pressure of Model 2, there is a notable energy flow at the beginning of simulation from the hot plasma region traversing to the cold plasma region.

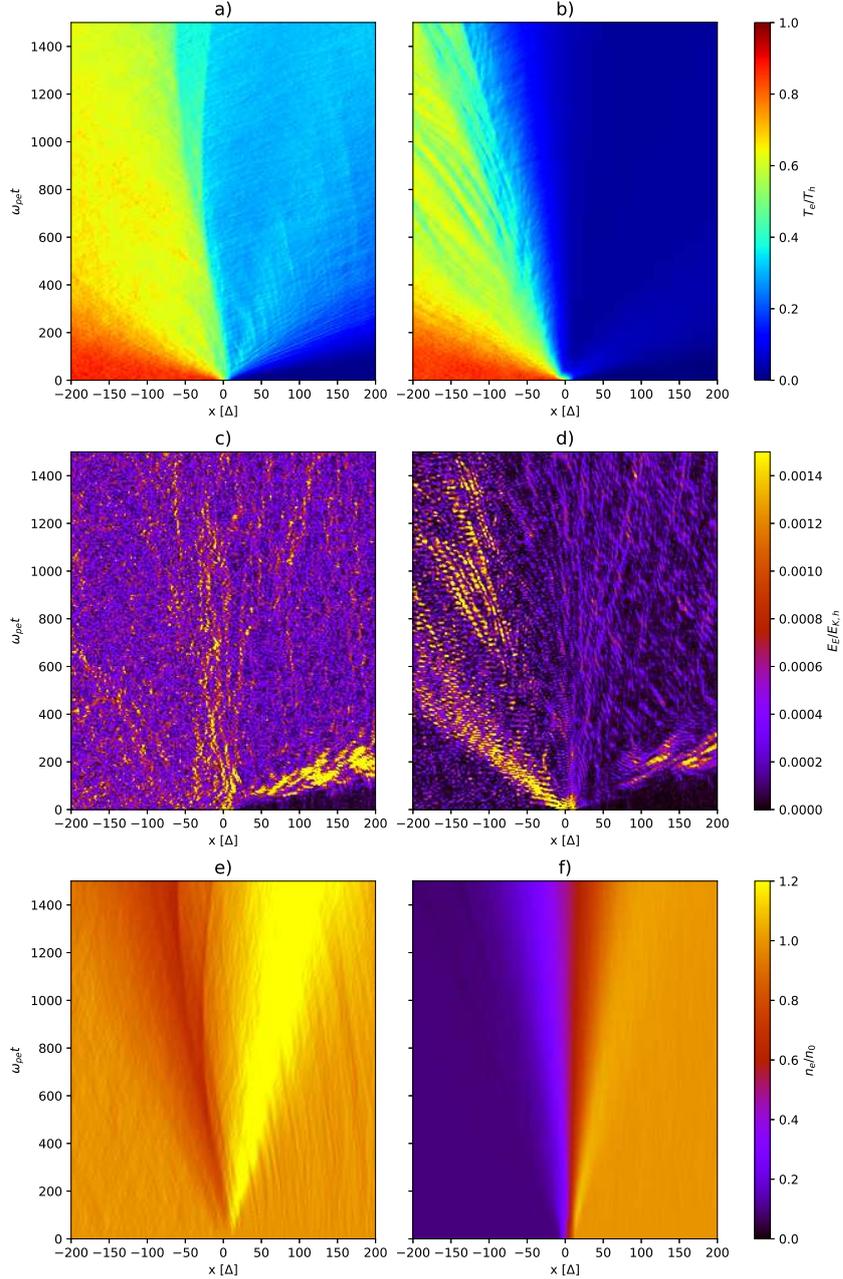


Figure 4.4: Left column: Simulation with unequilibrated pressure (Model 1). Right column: Simulation with equilibrated pressure (Model 2). a)-b): Time evolution of electron temperature as ratio to hot electron temperature  $T/T_h$ . c)-d) Time evolution of electric field energy as a ratio to hot electron thermal energy  $E_E/E_{K,h}$ . e)-f) Time development of electron density as a ratio to the initial grid density  $n_e/n_0$ .

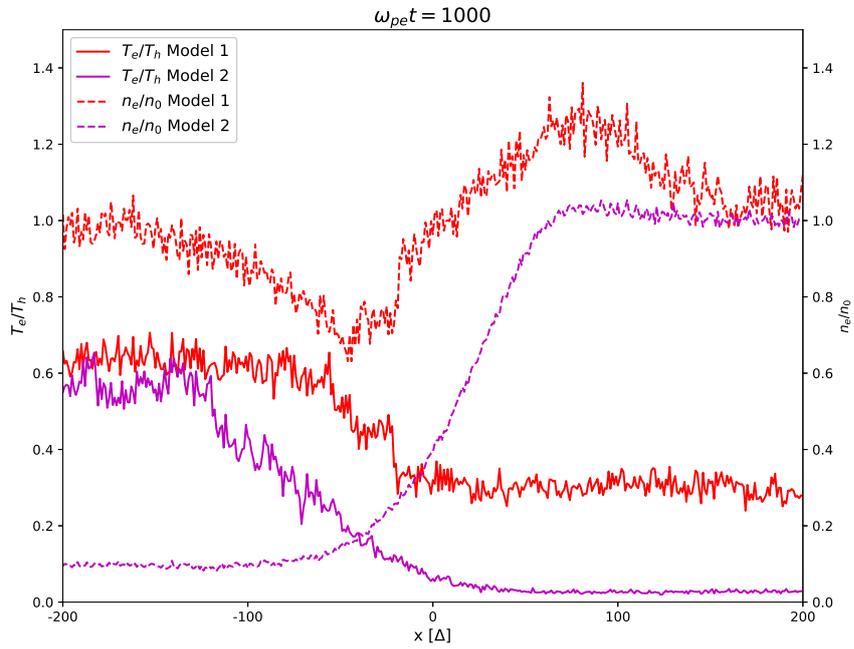


Figure 4.5: Solid lines: profile of electron temperature  $T_e/T_h$  in Model 1 and Model 2 at  $\omega_{pe}t = 1000$ . Dashed lines: profile of the electron density  $n_e/n_0$  in Model 1 and Model 2 at  $\omega_{pe}t = 1000$ . We can see that the thermal front forms at  $x \approx -20\Delta$  in Model 1 (unequilibrated pressure).

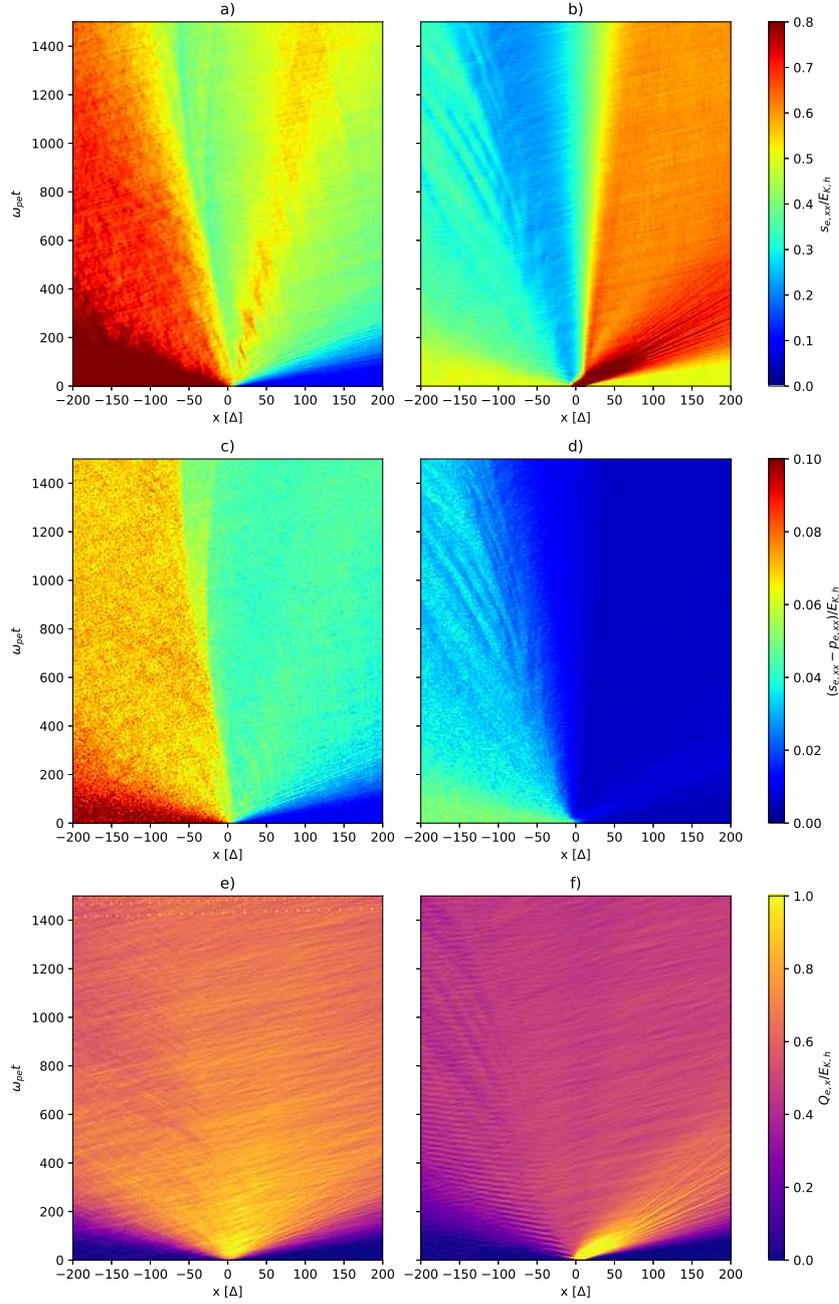


Figure 4.6: Left column: Model 1. Right column: Model 2. a)-b) Time evolution of the  $xx$ -component of the electron stress tensor normalized as a ratio to hot electron kinetic energy  $s_{e,xx}/E_{K,h}$ . c)-d) Time evolution of the difference between the  $xx$ -components of the ion stress and pressure tensor normalized to hot electron kinetic energy  $(s_{e,xx} - p_{e,xx})/E_{K,h}$ . e)-f) Time evolution of  $x$ -component of energy flux density as a ratio to hot electron kinetic energy  $Q_{e,x}/E_{K,h}$ .

In a similar fashion to the case of electrons, we also created graphs showing the time development of ion physical quantities. Figure 4.7a,b shows the time evolution of the ion temperature. We can see that the pressure equilibrium of Model 2 supports a less sharp temperature border between hot and cold plasma. In Model 1, the temperature transition is moved towards the hot plasma due to the lower ion density of this region. Even though the field is the same for both electrons and ions, we included its time evolution in Figure 4.7c,d to compare the symmetries and fluctuations with temperature and density time development. The time evolution of ion density (4.7e,f) is the same as the case of electrons in both Model 1 and Model 2. A more detailed view at temperature and particle density is provided in Figure 4.8. The graph shows the profile of ion temperature and density for both Model 1 and Model 2 at  $\omega_{pe}t = 1000$ . We can see that the temperature profile of both Model 1 and Model 2 differs at distribution towards the hot plasma region without a transparent border between hot and cold plasma, more like a smooth transition towards temperature equilibrium.

Figure 4.9 shows the time evolution of chosen components of moments of the distribution function for ions (discussed in Chapter 3.2.8). Time development of the  $xx$ -component of stress  $\mathbf{s}_i$  tensor is shown on 4.9a,b. There is a notable difference between both models. In the case of Model 2, the higher tensor value is transitioning towards the cold plasma. A better look is provided in Figure 4.9c,d, that displays the difference of the  $xx$ -component between the stress  $\mathbf{s}_i$  tensor and pressure tensor  $\mathbf{p}_i$ . We see that the equilibrated pressure leads to development along the edges of the double layer. There is an interesting development at  $\omega_{pe}t = 200$  that shows connection to the development of the below discussed electric potential (Figure 4.10).

Figure 4.9e,f shows the time evolution of the  $x$ -component of the energy flux density  $\mathbf{Q}_i$ . We see that the energy flux of ions is a lot lower than that of the electrons. The energy flux density of Model 1 is much larger than that of Model 2, but also more symmetrical in its spatial distribution due to not being in the state of pressure equilibrium.

Figure 4.10 shows the time development of electric potential as a ratio of electric energy  $e\Phi$  to the kinetic energy of hot electrons  $E_{K,h}$  of both models. In Model 1, there is a potential increase in the hot plasma region bordering the cold plasma region. This behavior is mirrored in Model 2, where we can see a sharp potential fall at the location of the hot-cold plasma border, forming a double layer potential. Better view at this phenomena is provided with Figures 4.11 and 4.12 that show the spatial distribution of the electron velocity overlaid with profile of electric potential at  $\omega_{pe}t = 1000$  for Model 1 and Model 2 respectively. The overlay of electric potential profile through these graphs yields us a major result: At the location of the steep fall of electric potential, we see that the electrons have higher density in the right of the potential fall in both models. We see that the velocity is distributed more homogeneously in the unequilibrated pressure model with the potential fall forming at the location of the formed thermal front. In the case of the equilibrated pressure model, the decrease in electric potential is larger followed by very low potential values in the cold plasma region. However, the profile of electric potential shows that the location of its drop occurs at the location of distinct border in particle distribution. The relative potential difference is similar for both models.

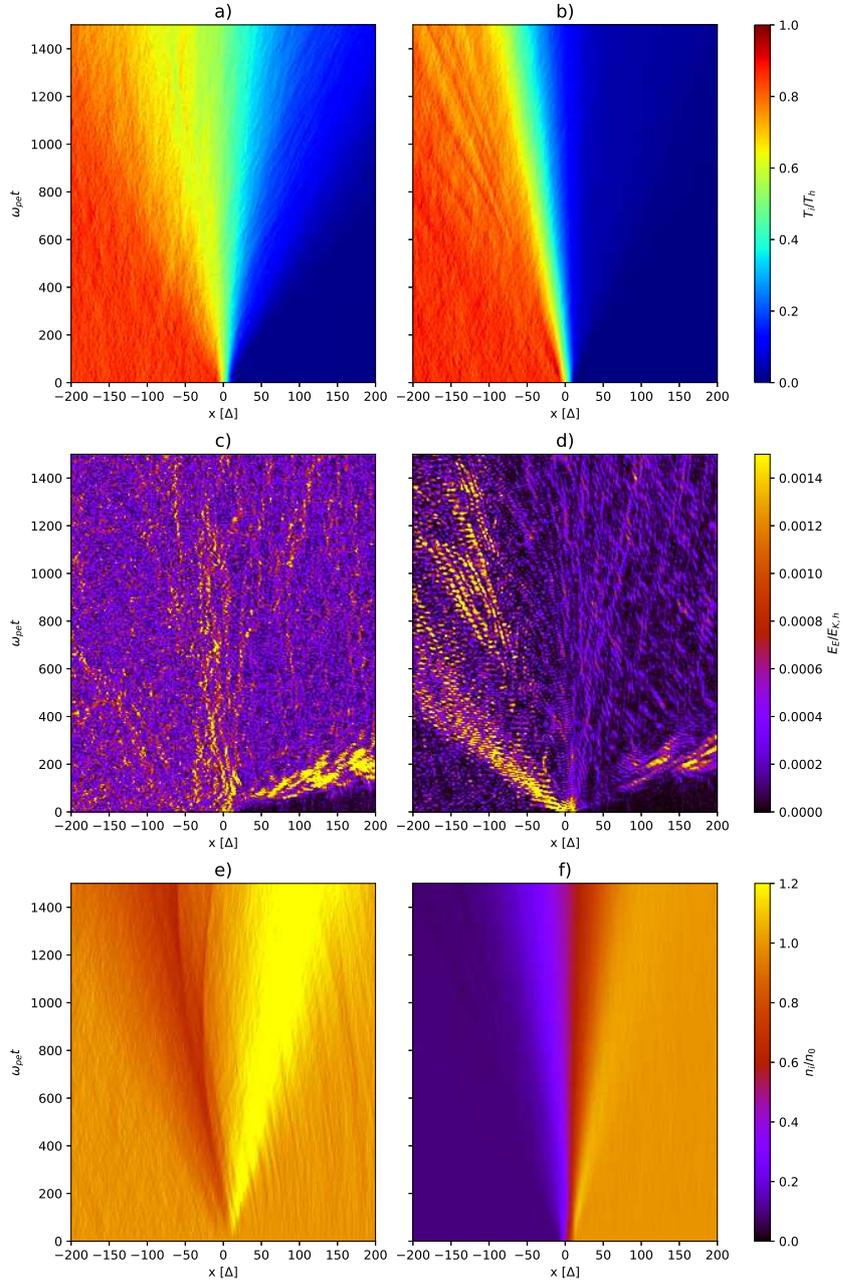


Figure 4.7: Left column: Simulation with unequilibrated pressure (Model 1). Right column: Simulation with equilibrated pressure (Model 2). a)-b) Time evolution of ion temperature as ratio to hot electron temperature  $T/T_h$ . c)-d) Time evolution of electric field energy as a ratio to hot electron thermal energy  $E_E/E_{K,h}$ . e)-f) Time development of ion density as a ratio to the initial grid density  $n_e/n_0$ .

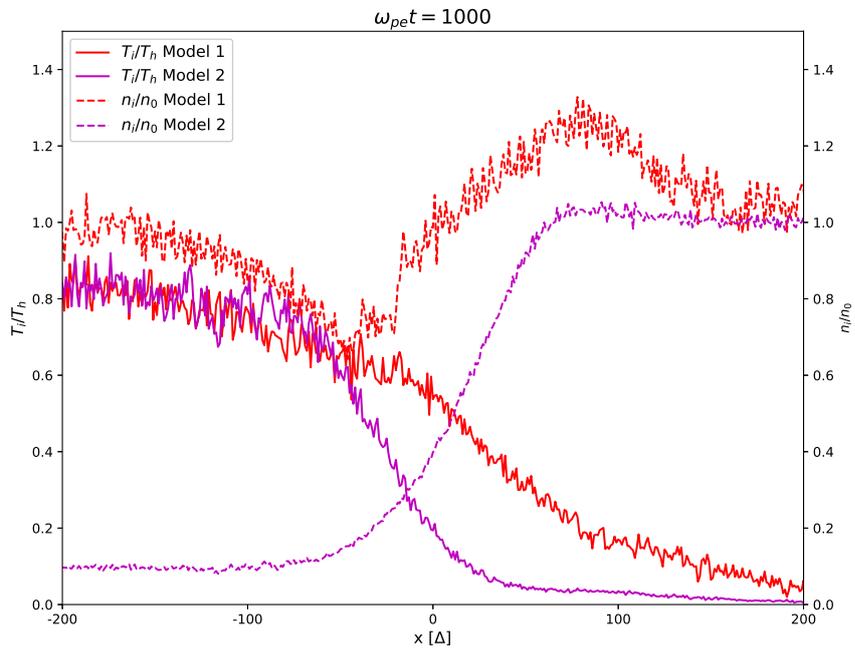


Figure 4.8: Solid lines: profile of ion temperature  $T_i/T_h$  in Model 1 and Model 2 at  $\omega_{pe}t = 1000$ . Dashed lines: profile of the ion density  $n_i/n_0$  in Model 1 and Model 2 at  $\omega_{pe}t = 1000$ . We can see that the thermal front forms at  $x = -20\Delta$  in Model 1 (unequilibrated pressure).

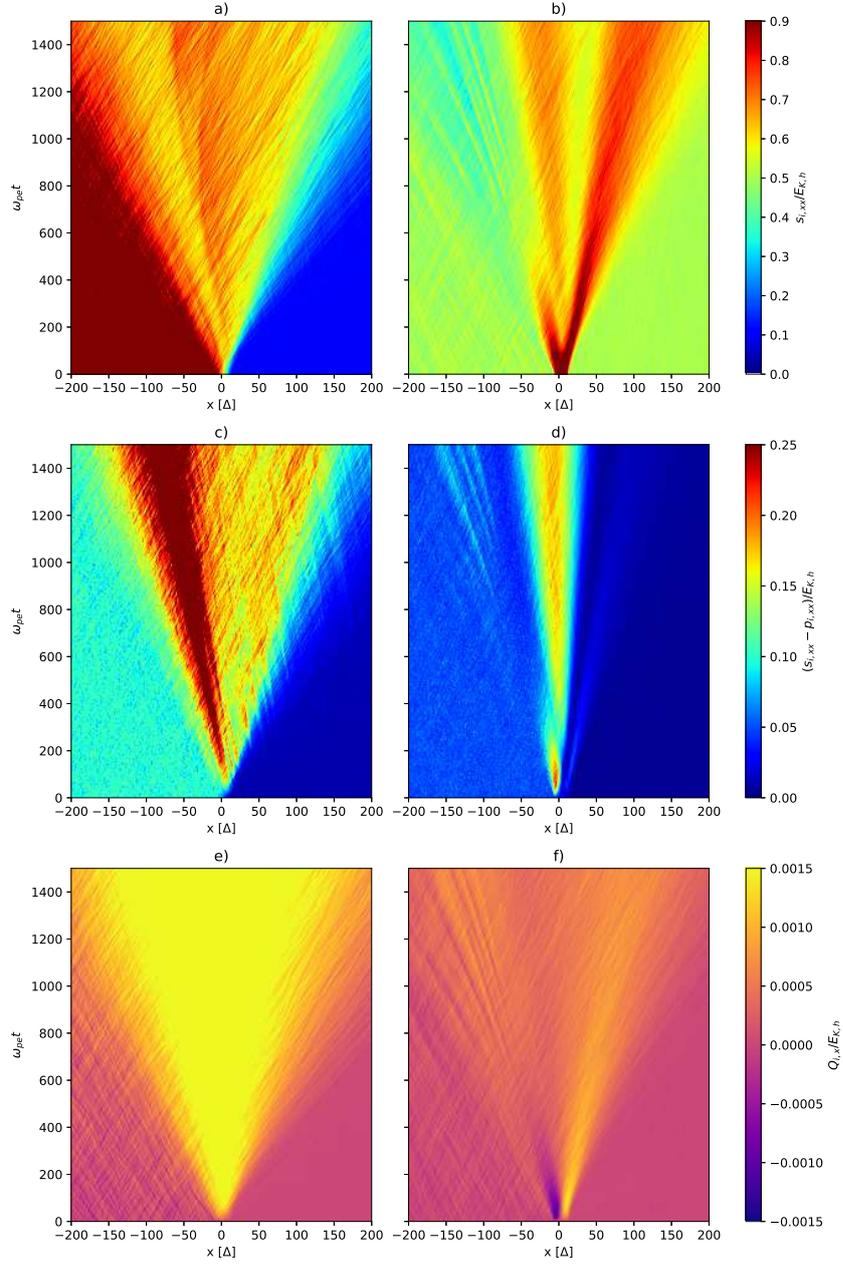


Figure 4.9: Left column: Model 1. Right column: Model 2. a)-b) Time evolution of the  $xx$ -component of the ion stress tensor normalized as a ratio to hot ion kinetic energy  $s_{i,xx}/E_{K,h}$ . c)-d) Time evolution of the difference between the  $xx$ -components of the ion stress and pressure tensor normalized to hot ion kinetic energy  $(s_{i,xx} - p_{i,xx})/E_{K,h}$ . e)-f) Time evolution of  $x$ -component of energy flux density vector as a ratio to hot ion kinetic energy  $Q_{i,x}/E_{K,h}$  (positive value indicates flow along the  $x$  axis, negative the opposite).

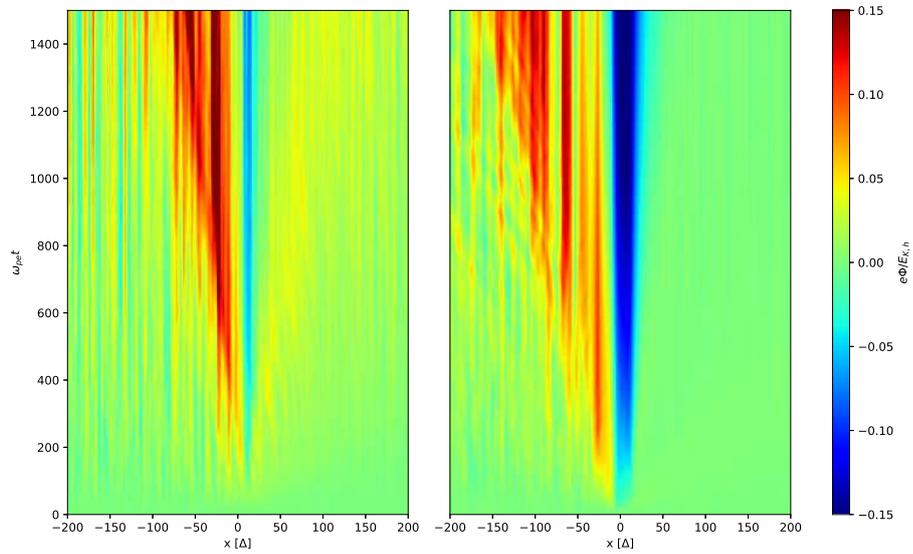


Figure 4.10: Left: evolution of the electric potential  $\Phi$  in Model 1. Right: evolution of the electric potential  $\Phi$  in the Model 2.

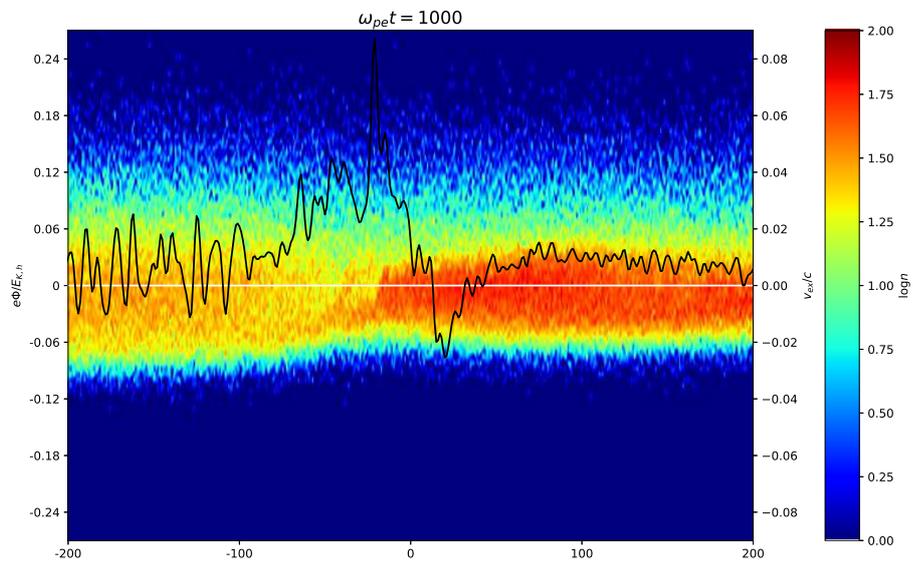


Figure 4.11: Spatial distribution of electron velocity overlaid with profile of the electric potential  $\Phi$  at  $\omega_{pe}t = 1000$  in Model 1.

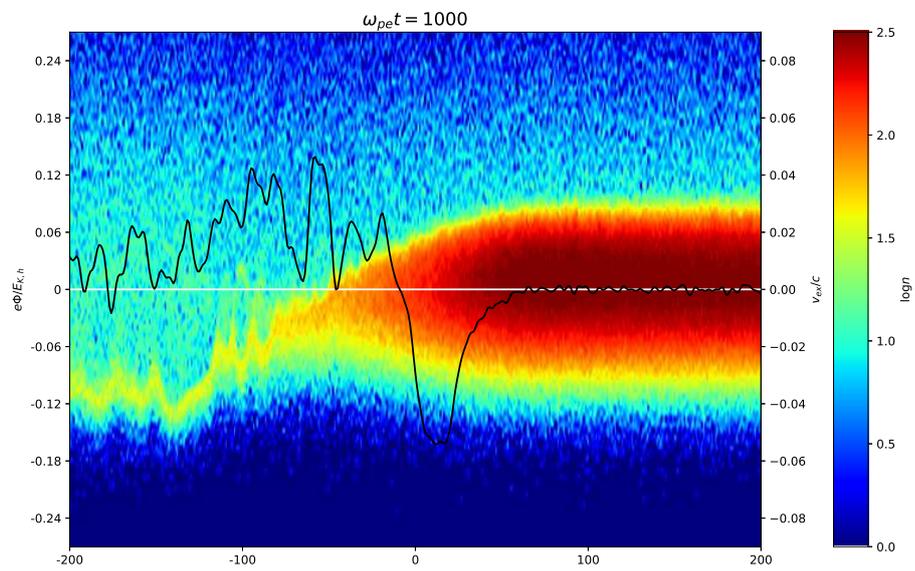


Figure 4.12: Spatial distribution of electron velocity overlaid with profile of the electric potential  $\Phi$  at  $\omega_{pe}t = 1000$  in Model 2.

# Discussion and Future Insights

In our work, we aimed to simulate the hot-cold plasma transition with accurate physical parameters as an application on the thermal front problem. We used a 1D approximation that accurately simulates the effects along the magnetic field lines because the plasma- $\beta$  parameter  $\approx 10^{-3}$  is low. We did not take the effects perpendicular to the magnetic field lines and heating processes into account. We also approximated the simulation region to consist only of electron-proton plasma.

We simulated the simulations in two cycles. First, we computed three simulations and observed the impact on the time development of the simulation with different configurations of initial transition length. By interpreting the computed data (Figure 4.3), we observed that the wider this region is, the longer it takes to form a notable temperature transition, which is a sign of a thermal front. We also noticed that with increasing initial transition length, stand-alone major fronts that formed with no initial transition tend to divide into multiple, less intense ones. In the case of the electric field and density fluctuations, they are fairly similar to the temperature development with their intensity being spread into several, less intense fluctuations.

In the second simulation cycle, we conducted two simulations aiming to demonstrate the impact of having unequilibrated and equilibrated pressure at the time of the simulation. We noticed different temperature evolution of electrons and ions, affected by different mass between them. The equilibrated pressure allows more turbulent electric field fluctuations and creates a distinct particle border (double layer) between the plasmas with different particle densities. However, the thermal fronts do not form in the equilibrated model, as opposed in the unequilibrated model, which can be seen in Figures 4.5, 4.8 for both the electrons and protons respectively. The stress and pressure tensors shown different time development between the two models – in the unequilibrated model, most of its high relative values are centered in the hot plasma, in the equilibrated one they were centered in the cold plasma region due to higher density. Electron tensor difference development (Figure 4.6c,d) shows similarities to the development of the electron temperature (Figure 4.4), matching its physical description. We then demonstrated the impact of the sharp fall of electric potential at the hot-cold plasma border resulting in an increased number of hot electrons at the cold plasma region, which yielded interesting results for the case of equilibrated pressure (Figure 4.12). In the case of ions, we see an interesting tensor difference transition at  $\omega_{pe}t \approx 200$  in the equilibrated model. At this time of the simulation, we see that the double layer potential starts to form (Figure 4.10), which may be worthy of further research by conducting larger scale simulations.

During the time of this work, we implemented several improvements to code. They include the storage for vector and tensor quantities, which allowed us to compute the moments of the distribution function, namely stress tensor, pressure tensor, and energy flux density vector.

Our simulations of the unequilibrated pressure model yielded similar results to the work of Karlický [32], even though it is worth mentioning that our model was much less precise in its size and time step as the mentioned work. To generate more accurate results, one would need a lot more computing power and data storage, which we did not possess at the moment of simulations computing. Future simulations could focus more on the pressure instability aspect of the phenomena. Observed transition in the ion stress and pressure tensor difference (Figure 4.9d) accompanied by the creation of double layer potential and simulation with no electric field to measure the free electron expansion are also worth further simulations. As our model simulated a very small region in a very small time frame (25 meters and  $1,67 \mu\text{s}$ ), having more allocated resources can lead to a simulation of a much larger size and time frame.

# Appendix

A showcase of the implementations made in TRISTAN code modified by Benáček [20]. Please note that full code is not enclosed, only examples of modified implementations, namely the subroutines for writing computed parameters into datasets and computation of parameters described in Chapter 3.2.8.

```

SUBROUTINE WriteDataSetParallelVec(idrank,dimsfi,dims,plist_id,file_id,&
  dataname,offset,count,stride,block,data, mxx,myy,mzz)
  use hdf5
  INCLUDE 'mpif.h'
  integer :: mxx,myy,mzz
  integer error
  INTEGER :: idrank ! Dataset rank
  INTEGER(HID_T) :: filespace, memspace
  INTEGER(HID_T) :: dset_id ! Dataset identifier
  character(len=*) :: dataname

  INTEGER(HSIZE_T), DIMENSION(4) :: count, dimsfi ! All grid dimensions
  INTEGER(HSIZE_T), DIMENSION(4) :: offset, stride, block, dims
  INTEGER(HID_T) :: file_id, plist_id ! File and plist identifier for HDF5

  REAL(4) :: data(3,mxx-5,myy-5,mzz-5)

  ! Create the data space for the dataset.

  CALL h5screate_simple_f(idrank, dimsfi, filespace, error)
  CALL h5screate_simple_f(idrank, dims, memspace, error)

  ! Create chunked dataset.

  CALL h5pcreate_f(H5P_DATASET_CREATE_F, plist_id, error)

  ! For more stable
  CALL h5pset_chunk_f(plist_id, idrank, dimsfi, error)

  ! Recommended by tutorial
  CALL h5pset_chunk_f(plist_id, idrank, dims, error)

  CALL h5dcreate_f(file_id, dataname, H5T_NATIVE_REAL, filespace, &
    dset_id, error, plist_id)
  CALL h5sclose_f(filespace, error)

  ! Each process defines dataset in memory and writes it to the hyperslab
  ! in the file.

  ! Select hyperslab in the file.

  CALL h5dget_space_f(dset_id, filespace, error)
  CALL h5sselect_hyperslab_f (filespace, H5S_SELECT_SET_F, offset, count, error,
    stride, block)

  ! Create property list for collective dataset write

  CALL h5pcreate_f(H5P_DATASET_XFER_F, plist_id, error)
  CALL h5pset_dxpl_mpio_f(plist_id, H5FD_MPIO_COLLECTIVE_F, error)

  ! Write the dataset collectively.

  CALL h5dwrite_f(dset_id, H5T_NATIVE_REAL, data, dimsfi, error, &
    file_space_id = filespace, mem_space_id = memspace, xfer_prp = plist_id)

  ! Close dataspace.

```

```

CALL h5sclose_f(filespace, error)
CALL h5sclose_f(memspace, error)

! Close the dataset.

CALL h5dclose_f(dset_id, error)

END

SUBROUTINE WriteDataSetParallelTen(idrank,dimsfi,dims,plist_id,file_id,&
  dataname,offset,count,stride,block,data, mxx,myy,mzz)
  use hdf5
  INCLUDE 'mpif.h'
  integer :: mxx,myy,mzz
  integer error
  INTEGER :: idrank ! Dataset rank
  INTEGER(HID_T) :: filespace, memspace
  INTEGER(HID_T) :: dset_id ! Dataset identifier
  character(len=*) :: dataname

  INTEGER(HSIZE_T), DIMENSION(5) :: count, dimsfi ! All grid dimensions
  INTEGER(HSIZE_T), DIMENSION(5) :: offset, stride, block, dims
  INTEGER(HID_T) :: file_id, plist_id ! File and plist identifier for HDF5
  REAL(4) :: data(3,3,mxx-5,myy-5,mzz-5)

  ! Create the data space for the dataset.

  CALL h5screate_simple_f(idrank, dimsfi, filespace, error)
  CALL h5screate_simple_f(idrank, dims, memspace, error)

  ! Create chunked dataset.

  CALL h5pcreate_f(H5P_DATASET_CREATE_F, plist_id, error)

  ! For more stable

  CALL h5pset_chunk_f(plist_id, idrank, dimsfi, error)
  CALL h5pset_chunk_f(plist_id, idrank, dims, error)

  CALL h5dcreate_f(file_id, dataname, H5T_NATIVE_REAL, filespace, &
    dset_id, error, plist_id)
  CALL h5sclose_f(filespace, error)

  ! Each process defines dataset in memory and writes it to the hyperslab
  ! in the file.

  ! Select hyperslab in the file.

  CALL h5dget_space_f(dset_id, filespace, error)
  CALL h5sselect_hyperslab_f (filespace, H5S_SELECT_SET_F, offset, count,&
    error, stride, block)

  ! Create property list for collective dataset write

  CALL h5pcreate_f(H5P_DATASET_XFER_F, plist_id, error)
  CALL h5pset_dxpl_mpio_f(plist_id, H5FD_MPIO_COLLECTIVE_F, error)

```

```

! Write the dataset collectively.

CALL h5dwrite_f(dset_id, H5T_NATIVE_REAL, data, dimsfi, error, &
file_space_id = filespace, mem_space_id = memspace, xfer_prp = plist_id)

! Close dataspace.

CALL h5sclose_f(filespace, error)
CALL h5sclose_f(memspace, error)

! Close the dataset.

CALL h5dclose_f(dset_id, error)

END

subroutine write_parameters(x,y,z,u,v,w,ions,lecs,maxhalf, &
                           cartd, rank, MPIposition, MPIdims, mxx, myy, mzz, nstep)
! Saves selected parameters along line in x direction into file "dataline".
! The direction is strictly along x.
! Saved parameters:
! - particle density
! - mean velocity
! - temperature - two types

use hdf5
use config, only: me,mi
implicit none
INCLUDE 'mpif.h'
real(8), dimension(:) :: x,y,z, u,v,w
integer :: ions, lecs, maxhalf
integer :: MPIposition(3), MPIdims(3), mxx,myy,mzz, rank, cartd, nstep

real(4), dimension(mxx-5,myy-5,mzz-5) :: vele_x, vele_y, vele_z
real(4), dimension(mxx-5,myy-5,mzz-5) :: veli_x, veli_y, veli_z
real(4), dimension(mxx-5,myy-5,mzz-5) :: tempe, tempi
integer, dimension(mxx-5,myy-5,mzz-5) :: vele_num, veli_num, tempe_num, tempi_num
real(4), dimension(3, mxx-5,myy-5,mzz-5) :: vele, veli
! energy flux density, vector quadrates
real(4), dimension(3, mxx-5,myy-5,mzz-5) :: efe, efi
real(4) :: v2e, v2i
real(4), dimension(3) :: we, wi
real(8) :: v2
integer :: x0,y0,z0, n, error
integer :: count_rate, count_t1, count_t2
real(4) :: count_delta
! stress and pressure tensor
real(4), dimension(3,3,mxx-5,myy-5,mzz-5) :: stresse, stressi, presse, pressi

INTEGER(HID_T) :: file_id, plist_id      ! File and plist identifier for HDF5
INTEGER(HID_T) :: dspace_id             ! Dataspace identifier for HDF5
INTEGER(HID_T) :: dset_id               ! Dataset identifier
INTEGER(HSIZE_T), DIMENSION(3) :: dims ! Dataset dimensions (chunck dimension)
INTEGER(HSIZE_T), DIMENSION(4) :: dimvec
INTEGER(HSIZE_T), DIMENSION(5) :: dimten
INTEGER(HSIZE_T), DIMENSION(3) :: count, dimsfi ! All grid dimensions
INTEGER(HSIZE_T), DIMENSION(4) :: countvec, dimsfivec
INTEGER(HSIZE_T), DIMENSION(5) :: counttten, dimsfiten
INTEGER(HSIZE_T), DIMENSION(3) :: offset, stride, block
INTEGER(HSIZE_T), DIMENSION(4) :: offsetvec, stridevec, blockvec

```

```

INTEGER(HSIZE_T), DIMENSION(5) :: offsetten, strideten, blockten
character(len=50) :: name
INTEGER :: idrank = 3 ! Dataset rank
INTEGER :: idrankvec = 4
INTEGER :: idrankten = 5

CALL SYSTEM_CLOCK(count_rate=count_rate)
CALL SYSTEM_CLOCK(count_t1)

dims(1) = mxx-5
dims(2) = myy-5
dims(3) = mzz-5

dimvec(1) = 3
dimvec(2) = mxx-5
dimvec(3) = myy-5
dimvec(4) = mzz-5

dimten(1) = 3
dimten(2) = 3
dimten(3) = mxx-5
dimten(4) = myy-5
dimten(5) = mzz-5

! Ions
veli_x = 0
veli_y = 0
veli_z = 0
veli_num = 0
do n=1,ions
  x0 = int(x(n)-2.)
  y0 = int(y(n)-2.)
  z0 = int(z(n)-2.)
  veli_x(x0,y0,z0) = veli_x(x0,y0,z0) + real(u(n),4)
  veli_y(x0,y0,z0) = veli_y(x0,y0,z0) + real(v(n),4)
  veli_z(x0,y0,z0) = veli_z(x0,y0,z0) + real(w(n),4)
  veli(:,x0,y0,z0) = veli(:,x0,y0,z0) + real((/u(n),v(n),w(n)/),4)
  veli_num(x0,y0,z0) = veli_num(x0,y0,z0) + 1

  v2i = real(0.5*mi*(u(n)*u(n) + v(n)*v(n) + w(n)*w(n)),4)
  efi(:,x0,y0,z0) = efi(:,x0,y0,z0) + real((/v2i*u(n),v2i*v(n),v2i*w(n)/),4)

  stressi(1,1,x0,y0,z0) = stressi(1,1,x0,y0,z0) + mi*real(u(n)*u(n),4)
  stressi(1,2,x0,y0,z0) = stressi(1,2,x0,y0,z0) + mi*real(u(n)*v(n),4)
  stressi(1,3,x0,y0,z0) = stressi(1,3,x0,y0,z0) + mi*real(u(n)*w(n),4)
  stressi(2,1,x0,y0,z0) = stressi(2,1,x0,y0,z0) + mi*real(v(n)*u(n),4)
  stressi(2,2,x0,y0,z0) = stressi(2,2,x0,y0,z0) + mi*real(v(n)*v(n),4)
  stressi(2,3,x0,y0,z0) = stressi(2,3,x0,y0,z0) + mi*real(v(n)*w(n),4)
  stressi(3,1,x0,y0,z0) = stressi(3,1,x0,y0,z0) + mi*real(w(n)*u(n),4)
  stressi(3,2,x0,y0,z0) = stressi(3,2,x0,y0,z0) + mi*real(w(n)*v(n),4)
  stressi(3,3,x0,y0,z0) = stressi(3,3,x0,y0,z0) + mi*real(w(n)*w(n),4)
end do
where (veli_num.eq.0) veli_num = 1
veli_x = veli_x / veli_num
veli_y = veli_y / veli_num
veli_z = veli_z / veli_num
veli(1,::,:) = veli(1,::,:) / veli_num
veli(2,::,:) = veli(2,::,:) / veli_num
veli(3,::,:) = veli(3,::,:) / veli_num

```

```

! Electrons
vele_x = 0
vele_y = 0
vele_z = 0
vele_num = 0
do n=(maxhalf+1),(maxhalf+lecs)
  x0 = int(x(n)-2.)
  y0 = int(y(n)-2.)
  z0 = int(z(n)-2.)
  vele_x(x0,y0,z0) = vele_x(x0,y0,z0) + real(u(n),4)
  vele_y(x0,y0,z0) = vele_y(x0,y0,z0) + real(v(n),4)
  vele_z(x0,y0,z0) = vele_z(x0,y0,z0) + real(w(n),4)
  vele(:,x0,y0,z0) = vele(:,x0,y0,z0) + real((/u(n),v(n),w(n)/),4)
  vele_num(x0,y0,z0) = vele_num(x0,y0,z0) + 1
  v2e = real(0.5*mi*(u(n)*u(n) + v(n)*v(n) + w(n)*w(n)),4)
  efe(:,x0,y0,z0) = efe(:,x0,y0,z0) + real((/v2e*u(n),v2e*v(n),v2e*w(n)/),4)
  stresse(1,1,x0,y0,z0) = stresse(1,1,x0,y0,z0) + me*real(u(n)*u(n),4)
  stresse(1,2,x0,y0,z0) = stresse(1,2,x0,y0,z0) + me*real(u(n)*v(n),4)
  stresse(1,3,x0,y0,z0) = stresse(1,3,x0,y0,z0) + me*real(u(n)*w(n),4)
  stresse(2,1,x0,y0,z0) = stresse(2,1,x0,y0,z0) + me*real(v(n)*u(n),4)
  stresse(2,2,x0,y0,z0) = stresse(2,2,x0,y0,z0) + me*real(v(n)*v(n),4)
  stresse(2,3,x0,y0,z0) = stresse(2,3,x0,y0,z0) + me*real(v(n)*w(n),4)
  stresse(3,1,x0,y0,z0) = stresse(3,1,x0,y0,z0) + me*real(w(n)*u(n),4)
  stresse(3,2,x0,y0,z0) = stresse(3,2,x0,y0,z0) + me*real(w(n)*v(n),4)
  stresse(3,3,x0,y0,z0) = stresse(3,3,x0,y0,z0) + me*real(w(n)*w(n),4)
end do
where (vele_num.eq.0) vele_num = 1
vele_x = vele_x / vele_num
vele_y = vele_y / vele_num
vele_z = vele_z / vele_num
vele(1,::,:) = vele(1,::,:) / vele_num
vele(2,::,:) = vele(2,::,:) / vele_num
vele(3,::,:) = vele(3,::,:) / vele_num

! Ion temperature and pressure tensor
tempi = 0
tempi_num = 0
do n=1,ions
  x0 = int(x(n)-2.)
  y0 = int(y(n)-2.)
  z0 = int(z(n)-2.)
  v2 = (u(n)-veli_x(x0,y0,z0))**2 + (v(n)-veli_y(x0,y0,z0))**2 + &
        (w(n)-veli_z(x0,y0,z0))**2
  tempi(x0,y0,z0) = tempi(x0,y0,z0) + real(v2,4)
  tempi_num(x0,y0,z0) = tempi_num(x0,y0,z0) + 1
  wi(:) = real((/u(n),v(n),w(n)/) - veli(:,x0,y0,z0),4)
  pressi(1,1,x0,y0,z0) = pressi(1,1,x0,y0,z0) + mi*real(wi(1)*wi(1),4)
  pressi(1,2,x0,y0,z0) = pressi(1,2,x0,y0,z0) + mi*real(wi(1)*wi(2),4)
  pressi(1,3,x0,y0,z0) = pressi(1,3,x0,y0,z0) + mi*real(wi(1)*wi(3),4)
  pressi(2,1,x0,y0,z0) = pressi(2,1,x0,y0,z0) + mi*real(wi(2)*wi(1),4)
  pressi(2,2,x0,y0,z0) = pressi(2,2,x0,y0,z0) + mi*real(wi(2)*wi(2),4)
  pressi(2,3,x0,y0,z0) = pressi(2,3,x0,y0,z0) + mi*real(wi(2)*wi(3),4)
  pressi(3,1,x0,y0,z0) = pressi(3,1,x0,y0,z0) + mi*real(wi(3)*wi(1),4)
  pressi(3,2,x0,y0,z0) = pressi(3,2,x0,y0,z0) + mi*real(wi(3)*wi(2),4)
  pressi(3,3,x0,y0,z0) = pressi(3,3,x0,y0,z0) + mi*real(wi(3)*wi(3),4)
end do
where (tempi_num.eq.0) tempi_num = 1
tempi = tempi / tempi_num * mi

```

```

! Electron temperature and pressure tensor
tempe = 0
tempe_num = 0
do n=(maxhalf+1),(maxhalf+lecs)
  x0 = int(x(n)-2.)
  y0 = int(y(n)-2.)
  z0 = int(z(n)-2.)
  v2 = (u(n)-vele_x(x0,y0,z0))**2 + (v(n)-vele_y(x0,y0,z0))**2 + &
        (w(n)-vele_z(x0,y0,z0))**2
  tempe(x0,y0,z0) = tempe(x0,y0,z0) + real(v2,4)
  tempe_num(x0,y0,z0) = tempe_num(x0,y0,z0) + 1
  we(:) = real((/u(n),v(n),w(n)/) - vele(:,x0,y0,z0),4)
  presse(1,1,x0,y0,z0) = presse(1,1,x0,y0,z0) + me*real(we(1)*we(1),4)
  presse(1,2,x0,y0,z0) = presse(1,2,x0,y0,z0) + me*real(we(1)*we(2),4)
  presse(1,3,x0,y0,z0) = presse(1,3,x0,y0,z0) + me*real(we(1)*we(3),4)
  presse(2,1,x0,y0,z0) = presse(2,1,x0,y0,z0) + me*real(we(2)*we(1),4)
  presse(2,2,x0,y0,z0) = presse(2,2,x0,y0,z0) + me*real(we(2)*we(2),4)
  presse(2,3,x0,y0,z0) = presse(2,3,x0,y0,z0) + me*real(we(2)*we(3),4)
  presse(3,1,x0,y0,z0) = presse(3,1,x0,y0,z0) + me*real(we(3)*we(1),4)
  presse(3,2,x0,y0,z0) = presse(3,2,x0,y0,z0) + me*real(we(3)*we(2),4)
  presse(3,3,x0,y0,z0) = presse(3,3,x0,y0,z0) + me*real(we(3)*we(3),4)
end do

dimsfi(1) = MPIdims(1) * dims(1)
dimsfi(2) = MPIdims(2) * dims(2)
dimsfi(3) = MPIdims(3) * dims(3)

dimsfivec(1) = 3
dimsfivec(2) = MPIdims(1) * dims(1)
dimsfivec(3) = MPIdims(2) * dims(2)
dimsfivec(4) = MPIdims(3) * dims(3)

dimsfiten(1) = 3
dimsfiten(2) = 3
dimsfiten(3) = MPIdims(1) * dims(1)
dimsfiten(4) = MPIdims(2) * dims(2)
dimsfiten(5) = MPIdims(3) * dims(3)

offset(1) = MPIposition(1) * dims(1)
offset(2) = MPIposition(2) * dims(2)
offset(3) = MPIposition(3) * dims(3)

offsetvec(1) = 0
offsetvec(2) = MPIposition(1) * dims(1)
offsetvec(3) = MPIposition(2) * dims(2)
offsetvec(4) = MPIposition(3) * dims(3)

offsetten(1) = 0
offsetten(2) = 0
offsetten(3) = MPIposition(1) * dims(1)
offsetten(4) = MPIposition(2) * dims(2)
offsetten(5) = MPIposition(3) * dims(3)

stride(1) = 1
stride(2) = 1
stride(3) = 1
stridevec(:) = 1
strideten(:) = 1

```

```

count(1) = 1
count(2) = 1
count(3) = 1
countvec(1) = 1
countvec(2) = 1
countvec(3) = 1
countvec(4) = 1
countten(:) = 1

block(1) = dims(1)
block(2) = dims(2)
block(3) = dims(3)

blockvec(1) = dimvec(1)
blockvec(2) = dimvec(2)
blockvec(3) = dimvec(3)
blockvec(4) = dimvec(4)

blockten(1) = dimten(1)
blockten(2) = dimten(2)
blockten(3) = dimten(3)
blockten(4) = dimten(4)
blockten(5) = dimten(5)

CALL h5pcreate_f(H5P_FILE_ACCESS_F, plist_id, error)
CALL h5pset_fapl_mpio_f(plist_id, MPI_COMM_WORLD, MPI_INFO_NULL, error)

CALL h5fcreate_f(name, H5F_ACC_TRUNC_F, file_id, error, &
access_prp = plist_id)
CALL h5pclose_f(plist_id, error)

call WriteDataSetParallel(idrank,dimsfi,dims,plist_id,file_id,"dense",&
offset,count,stride,block,real(vele_num,4),mxx,myy,mzz)

call WriteDataSetParallel(idrank,dimsfi,dims,plist_id,file_id,"densi",&
offset,count,stride,block,real(veli_num,4),mxx,myy,mzz)

call WriteDataSetParallel(idrank,dimsfi,dims,plist_id,file_id,"velex",&
offset,count,stride,block,vele_x,mxx,myy,mzz)

call WriteDataSetParallel(idrank,dimsfi,dims,plist_id,file_id,"veley",&
offset,count,stride,block,vele_y,mxx,myy,mzz)

call WriteDataSetParallel(idrank,dimsfi,dims,plist_id,file_id,"velez",&
offset,count,stride,block,vele_z,mxx,myy,mzz)

call WriteDataSetParallelVec(idrankvec,dimsfivec,dimvec,plist_id,file_id,&
"vele",offsetvec,countvec,stridevec,blockvec,vele,mxx,myy,mzz)

call WriteDataSetParallelVec(idrankvec,dimsfivec,dimvec,plist_id,file_id,&
"efe",offsetvec,countvec,stridevec,blockvec,efe,mxx,myy,mzz)

call WriteDataSetParallelTen(idrankten,dimsfiten,dimten,plist_id,file_id,&
"stresse",offsetten,countten,strideten,blockten,stresse,mxx,myy,mzz)

call WriteDataSetParallelTen(idrankten,dimsfiten,dimten,plist_id,file_id,&
"presse",offsetten,countten,strideten,blockten,presse,mxx,myy,mzz)

call WriteDataSetParallelVec(idrankvec,dimsfivec,dimvec,plist_id,file_id,&
"veli",offsetvec,countvec,stridevec,blockvec,veli,mxx,myy,mzz)

```

```

call WriteDataSetParallelVec(idrankvec,dimsfivec,dimvec,plist_id,file_id,&
"efi",offsetvec,countvec,stridevec,blockvec,efi,mxx,myy,mzz)

call WriteDataSetParallelTen(idrankten,dimsfiten,dimten,plist_id,file_id,&
"stressi",offsetten,countten,strideten,blockten,stressi,mxx,myy,mzz)

call WriteDataSetParallelTen(idrankten,dimsfiten,dimten,plist_id,file_id,&
"pressi",offsetten,countten,strideten,blockten,pressi,mxx,myy,mzz)

call WriteDataSetParallel(idrank,dimsfi,dims,plist_id,file_id,"tempe",&
offset,count,stride,block,tempe,mxx,myy,mzz)

call WriteDataSetParallel(idrank,dimsfi,dims,plist_id,file_id,"tempi",&
offset,count,stride,block,tempi,mxx,myy,mzz)

CALL h5pclose_f(plist_id, error)
CALL h5fclose_f(file_id, error)

if(rank.eq.0) then
    call AddAttribute(name, (/ "dense", "densi", "tempe", "tempi", "velex", &
    "veley", "velez", "velix", "veliy", "veliz"/),MPIposition, time_step=nstep,
    rank=rank)
end if
end if

CALL SYSTEM_CLOCK(count_t2)
count_delta = real(count_t2-count_t1)
WRITE(*,*) "Wall time for saving parameters:", count_delta/count_rate

end subroutine

```

# Bibliography

- [1] M. Karlický, *Plasma Astrophysics*, MATFYZPRESS, vydavatelství Matematicko-fyzikální fakulty Univerzity Karlovy Praha, 2014, 161, 978-80-7378-281-8.
- [2] P. A. Sweet, *The Neutral Point Theory of Solar Flares*, IAU Symposium 6, Electromagnetic Phenomena in Cosmical Physics, ed. B. Lehnert (Dordrecht: Kluwer), 123, 1958.
- [3] H. E. Petschek, *Magnetic Field Annihilation*, The Physics of Solar Flares, Proceedings of the AAS-NASA Symposium held 28–30 October 1963 at the Goddard Space Flight Center, Greenbelt, MD, p. 425, 1964.
- [4] E. R. Priest, *Solar magneto-hydrodynamics*, Dordrech, Holland, 1982 , 82-5225 (BKS3).
- [5] M. Aschwanden, *Physics of the Solar Corona: An Introduction with Problems and Solutions*, Second Edition, Springer, 2006, 924, 3-540-30765-6.
- [6] O. Engvold, J.-C. Vial, A. Skumanich, *Sun as a guide to stellar physics*, Amsterdam: Elsevier, 2018, 493, 9780128143353.
- [7] P. Jelínek, *Magnetohydrodynamické vlny a oscilace ve sluneční koróně*, Jihočeská univerzita v Českých Budějovicích, Přírodovědecká fakulta, 2017, 150, 978-80-7394-656-2.
- [8] B. De Pontieu, *Chromospheric Alfvénic Waves Strong Enough to Power the Solar Wind*, Science, Volume 318, Issue 5856, pp. 1574, December 2007.
- [9] S. Tomczyk, *Alfvén Waves in the Solar Corona*, Science, Volume 317, Issue 5842, pp. 1192, August 2007.
- [10] M. Wakatani, Kyoji Nishikawa, *Plasma Physics: Basic Theory with Fusion Applications*, Springer, 1994, 352, 354065285X.
- [11] A. Achiezer, *Elektrodinamika plazmy*, Nauka, Moskva, 720, 1974.
- [12] H. Alfvén, *On the Theory of Magnetic Storms and Aurorae*, Tellus, Vol. 10, Issue 1, p. 104, February 1958.
- [13] A. L. Peratt, *Physics of the Plasma Universe*, Second edition, Springer, 2015, 426, ISBN 978-1461478188.

- [14] G. A. Gary, *Plasma Beta above a Solar Active Region: Rethinking the Paradigm*, Solar Physics, v. 203, Issue 1, p. 71, October 2001.
- [15] P. Carlqvist, *On the acceleration of energetic cosmic particles by electrostatic double layers*, IEEE Transactions on Plasma Science, vol. PS-14, p. 794, December 1986.
- [16] P. Carlqvist, *On the physics of relativistic double layers*, Astrophysics and Space Science, Volume 87, Issue 1-2, p. 21, October 1982.
- [17] M. A. Raadu, *Relativistic effects in electrostatic double layers*, Symposium on Plasma Double Layers, Risoe, Denmark, p. 16, June 1982.
- [18] G. Lapenta, *Particle in Cell (PIC) Method*, 2018.
- [19] O. Buneman, *TRISTAN, The 3-D Electromagnetic Particle Code*
- [20] J. Benáček, *Particle-in-cell simulations and analytical study of the processes generating radio solar bursts*, Ph.D. thesis, Masarykova Univerzita, 2019.
- [21] K. Yee, *Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media*, IEEE Transactions on Antennas and Propagation, p. 302, 1966.
- [22] J. M. Dawson, *Particle simulations of plasmas*, Rev. Mod. Phys. 55, p. 403-447, April 1983.
- [23] O. Buneman, L. R. O. Storey, *Simulations of fusion plasmas by A 3-D, E-M particle code*, Final Report, Stanford University, CA. Space, Telecommunications and Radioscience Lab, March 1985.
- [24] R. Courant, K. Friedrichs, H. Lewy, *On the Partial Difference Equations of Mathematical Physics*, Mathematische Annalen, Vol. 100, p. 32, 1928.
- [25] H. Matsumoto, Y. Omura, *Computer Space Plasma Physics: Simulation Techniques and Software*, Terra Scientific Pub. Co., Tokyo, 1993, 487, 488704111X.
- [26] J. P. Boris, *The acceleration calculation from a scalar potential*, Plasma Physics Laboratory, Princeton University, MATT-152, March 1970.
- [27] R. W. Hockney, J. W. Eastwood, *Computer Simulation Using Particles*, New York: McGraw-Hill, 1981.
- [28] C. K. Birdsall, A. B. Langdon, *Plasma Physics via Computer Simulation*, The Adam Hilger Series on Plasma Physics, Adam Hilger, Bristol, England, 1991, 0-07-005371-5.
- [29] J. Villanese, O. Buneman, *Rigorous charge conservation for local electromagnetic field solvers*, Computer Physics Communications, Volume 69, Issue 2-3, p. 306, March 1992.
- [30] R. Fitzpatrick, *Plasma Physics*, The University of Texas, Austin, 2016.

- [31] C. J. Schrijver, A. M. Title, T. E. Berger, *A new view of the solar outer atmosphere by the Transition Region and Coronal Explorer*, *Solar Physics*, v. 187, Issue 2, p. 261-302, July 1999.
- [32] M. Karlický, *Thermal Fronts in Solar Flares*, *Astrophysical Journal*, v. 814, Issue 2, article id. 153, 7 pp., December 2015.
- [33] M. Karlický, František Karlický, *Hot-cold plasma transition region: collisionless case*, arXiv, September 2017.
- [34] T. C. Li, J. F. Drake, M. Swisdak, *Suppression of energetic electron transport in flares by double layers*, *Astrophysical Journal*, 757, p. 20, 2012.
- [35] T. C. Li, J. F. Drake, M. Swisdak, *Coronal electron confinement by double layers*, *Astrophysical Journal*, 778, p. 144, 2013.
- [36] T. C. Li, J. F. Drake, M. Swisdak, *Dynamics of double layers, ion acceleration, and heat flux suppression during solar flares*, *Astrophysical Journal*, 793, p. 7, 2014.
- [37] G. T. Roberg-Clark, J. F. Drake, M. Swisdak, C. S. Reynolds, *Wave Generation and Heat Flux Suppression in Astrophysical Plasma Systems*, *Astrophysical Journal*, 867, p. 154, 2018.
- [38] J. Guo, *The role of electrostatic waves in the formation of thermal front in solar flares: 1-D PIC simulation*, *Physics of Plasmas*, 26, 2019.
- [39] J. Sun, X. Gao, Y. Ke, Q. Lu, X. Wang, S. Wang, *Expansion of Solar Coronal Hot Electrons in an Inhomogeneous Magnetic Field: 1D PIC Simulation*, *Astrophysical Journal*, 887, p. 96, 2019.
- [40] ESA, "Anatomy of our Sun", *ESA.int*, ESA, October 2019 .
- [41] NSF's National Solar Observatory, "Rotation Profiles", *Stanford.edu*, Stanford, March 2000.
- [42] National Solar Observatory (NSO), AURA, NSF, "NSF's newest solar telescope produces first images", *NSO.edu*, NSO, January 2020.
- [43] Hinode Solar Observatory, "Another view of the Sun's 'chromosphere'", *ESA.int*, ESA, March 2007.
- [44] ESA-CESAR, "Polarised solar corona during 2019 La Silla Total Solar Eclipse", *ESO.org*, ESO, July 2019.
- [45] I. Tresman, *Double Layer Formation Summary*, based on: Singh, Nagendra; Thiemann, H.; Schunk, R. W., *Electric Fields and Double Layers in Plasmas* (1987), NASA Conference Publication, 2006.

