

CZECH TECHNICAL UNIVERSITY IN PRAGUE  
FACULTY OF NUCLEAR SCIENCES AND PHYSICAL  
ENGINEERING

Department: Department of Software Engineering  
Study programme: Applications of Informatics in Natural Science



# Modelling laser absorption using machine learning methods

MASTER THESIS

Author: Bc. Samuel Šitina  
Supervisor: doc. Ing. Ondřej Klimo, Ph.D.  
Year: 2024

**vložit naskenované zadání, kde jsou  
všechny podpisy!!!**

## I. OSOBNÍ A STUDIJNÍ

Příjmení:  Jméno:  Osobní číslo:

Fakulta/ústav: **Fakulta jaderná a fyzikálně inženýrská**

Zadávací katedra/ústav: **Katedra matematiky**

Studijní program: **(není nutné vyplňovat, doplní se po přepsání do systému)**

Specializace: **(není nutné vyplňovat, doplní se po přepsání do systému)**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Název bakalářské práce anglicky:

Pokyny pro vypracování:

1. první bod zadání
2. druhý bod zadání
3. atd.

Seznam doporučené literatury:

- [1] reference č. 1
- [2] reference č. 2
- [3] atd.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **bude doplněno** Termín odevzdání bakalářské práce: **bude doplněno**

Platnost zadání bakalářské práce: **bude doplněno**

(jméno vedoucího - bude doplněno)  
podpis vedoucí(ho) práce

(jméno vedoucího - bude doplněno)  
podpis vedoucí(ho) ústavu/katedry


doc. Ing. Václav Čuba, Ph.D.  
podpis děkana

### III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta



V případě jednostránkového zadání zakomentujte (v \*.tex) variantu B1 a odkomentujte např. variantu A1

### **Statement of originality**

Hereby I declare that this thesis is my original authorial work, which I have worked out on my own with the guidance of my supervisor. All sources, references, and literature used or excerpted during the elaboration of this work are properly cited and listed in complete reference to the due source.

In Prague on .....

.....

Bc. Samuel Šitina

## **Acknowledgment**

I would like to thank my supervisor Doc. Ing. Ondřej Klimo, Ph.D. for valuable guidance throughout the entire process. I would like to thank my parents who have supported unconditionally me in unimaginable ways. I would also like to thank my sister and my brother, whose advice helped me find motivation in the times of struggle.

Huge thanks goes to my friends (yes, even those of you who think plasma is not real), because without you life would be boring and it would be hard to find meaning in anything.

Bc. Samuel Šitina

*Title:* Modelling laser absorption using machine learning methods

*Author:* Bc. Samuel Šitina

*Study programme:* Applications of Informatics in Natural Science

*Type of thesis:* Master thesis

*Supervisor:* doc. Ing. Ondřej Klimo, Ph.D. Department of Laser Physics and Photonics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague

*Abstract:* Popis práce anglicky

*Key words:* Klíčová slova

*Title:*

**Modelling laser absorption using machine learning methods**

*Author:* Bc. Samuel Šitina

*Abstract:* Popis práce česky

*Key words:* Key words

# Contents

<b>Introduction</b>	<b>8</b>
<b>1 Plasma</b>	<b>9</b>
1.1 Temperature of plasma . . . . .	10
1.2 Ionization . . . . .	11
1.3 Absorption of ultra-short, ultra-intense lasers . . . . .	13
1.4 Motivation - X-ray $\mathbf{K}_\alpha$ emission . . . . .	16
1.5 PIC simulations . . . . .	17
<b>2 Temperature fitting</b>	<b>21</b>
2.1 Boltzmann vs. Maxwellian distribution . . . . .	22
2.2 Exponential-sum fitting . . . . .	23
<b>3 Surrogate hot electron temperature models</b>	<b>29</b>
3.1 Regression methods . . . . .	29
3.2 Support vector regression . . . . .	30
3.3 Neural networks . . . . .	31
3.4 Gaussian Processes . . . . .	33
<b>4 Dataset</b>	<b>36</b>
4.1 EPOCH Simulations . . . . .	36
4.2 Temperature fitting . . . . .	37
<b>5 Hot electron temperature modelling</b>	<b>39</b>
<b>6 Comparison to contemporary data</b>	<b>40</b>
<b>Conclusion</b>	<b>41</b>
<b>Bibliography</b>	<b>42</b>
<b>Attachments</b>	<b>46</b>
<b>A EPOCH input file</b>	<b>46</b>

# Introduction

Plasma, the fourth state of matter, constitutes 99% of the visible universe [13].



# Chapter 1

## Plasma

A plasma is a quasi-neutral gas of charged and neutral particles which exhibits collective behaviour [9]. In simple terms, quasi-neutrality means that the density of electrons  $n_e$  and density of positively charged ions  $n_i$  locally satisfy:

$$n_e \simeq Zn_i \quad (1.1)$$

where  $Ze$  is the charge of one positively charged ion and  $e$  is elementary charge [13].

The non-neutral particles in plasma are subject to electric and magnetic fields generated either by external sources or by neighbouring particles. The long-range nature of Coulomb potential ensures that macroscopic fields dominate over forces created by microscopic fluctuations [13]. To explain the collective behaviour properly, one can start by writing *Vlasov equation* [21]:

$$\frac{\partial f_j}{\partial t} + \mathbf{v} \cdot \frac{\partial f_j}{\partial \mathbf{x}} + \frac{q_j}{m_j} \left( \mathbf{E} + \frac{\mathbf{v} \times \mathbf{B}}{c} \right) \cdot \frac{\partial f_j}{\partial \mathbf{v}} = 0 \quad (1.2)$$

where  $f_j = f_j(\mathbf{x}, \mathbf{v}, t)$  is the phase space distribution function, which characterizes the location of the particles of species  $j$  (electrons or ions) in phase space  $(\mathbf{x}, \mathbf{v})$  (position, velocity) as a function of time.  $q_j$  and  $m_j$  are charge and mass of the species  $j$  and  $c$  is the speed of light [21].

After calculating the 0th and 1st moment of Vlasov equation (averaging through  $\mathbf{v}$ ), we obtain the equation of continuity and force equations for the density  $n_j = \int f_j(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}$  and mean velocity  $\mathbf{u}_j$  defined by  $n_j \mathbf{u}_j = \int \mathbf{v} f_j(\mathbf{x}, \mathbf{v}, t) d\mathbf{v}$ :

$$\frac{\partial n_j}{\partial t} + \nabla \cdot (n_j \mathbf{u}_j) = 0 \quad (1.3)$$

$$n_j \left( \frac{\partial \mathbf{u}_j}{\partial t} + (\mathbf{u}_j \cdot \nabla) \mathbf{u}_j \right) = \frac{n_j q_j}{m_j} \left( \mathbf{E} + \frac{\mathbf{u}_j \times \mathbf{B}}{c} \right) - \frac{1}{m_j} \nabla p_j \quad (1.4)$$

where  $p_j$  is isotropic pressure and in case of negligible heat flow also the adiabatic state equation:

$$p_j n_j^{-\gamma} = \text{const.} \quad (1.5)$$

where  $\gamma = (2 + N)/N$  and  $N$  is the number of degrees of freedom. Equations 1.3, 1.4 and 1.5 together with the Maxwell equations are often referred to as *two-fluid*

*model of plasma* and describe wide range of plasma (collective) behaviour such as plasma waves or Debye shielding [21].

## 1.1 Temperature of plasma

In this thesis, we are studying so called *hot electrons* produced by the interaction of short laser pulse of high intensity with plasma. All important details of the physical phenomena will be covered in later sections, but let us now look at what is meant by *hot* and how the temperature of plasma is usually understood.

In a gas in the thermal equilibrium, particles velocities are given by Maxwellian distribution (in three dimensions) [9]:

$$f(\mathbf{v}) = n \left( \frac{m}{2\pi k_B} \right)^{3/2} \exp \left( -\frac{\frac{1}{2}mv^2}{k_B T} \right) \quad (1.6)$$

where  $v = |\mathbf{v}|$  is magnitude of velocity,  $m$  is mass.  $k_B$  is the Boltzmann constant and  $T$  is temperature. The average kinetic energy  $E_{av}$  is then [9]:

$$E_{av} = \frac{3}{2} k_B T \quad (1.7)$$

Because of this relation between  $E_{av}$  and  $T$ , it is customary in plasma physics to express the temperature in the same units as energy. If  $k_B T = 1 \text{ eV} = 1.6 \times 10^{-19} \text{ J}$ , then [9]:

$$T = \frac{1.6 \times 10^{-19}}{1.38 \times 10^{-23}} = 11600 \quad (1.8)$$

From this it follows that the factor of the conversion is:

$$1 \text{ eV} = 11600 \text{ K} \quad (1.9)$$

The electrons and the ions can have different temperature [9]. Moreover, there can be multiple groups of electrons with different distributions, but we will describe this more deeply in one of the later sections.

## Critical density

Now consider a high frequency electric field  $\mathbf{E} = \mathbf{E}(\mathbf{x}) \exp(-i\omega t)$ . The frequency  $\omega$  is assumed to be greater than electron plasma frequency  $\omega_{pe}$  defined as  $\omega_{pe}^2 = 4\pi e^2 n_e / m_e$  with  $n_e = Z n_i$  being electron density and  $m_e$  electron mass. Maxwell equations plus equation of motion of electron fluid give us:

$$\nabla \times \mathbf{B} = -\frac{i\omega}{c} \epsilon \mathbf{E}, \quad (1.10)$$

where  $\epsilon = 1 - \omega_{pe}^2 / \omega^2$  defines the dielectric function of the plasma [21]. After further derivation using the other Maxwell equations and vector identities we can get:

$$\nabla^2 \mathbf{B} + \frac{\omega^2}{c^2} \epsilon \mathbf{B} + \frac{1}{\epsilon} \nabla \epsilon \times (\nabla \times \mathbf{B}) = 0 \quad (1.11)$$

Assuming space dependency described by  $\exp(i\mathbf{k} \cdot \mathbf{x})$ , the dispersion relation is then:

$$\omega^2 = \omega_{\text{pe}}^2 + k^2 c^2. \quad (1.12)$$

It is possible to show, that  $k$  becomes imaginary for  $\omega < \omega_{\text{pe}}$ . This can be interpreted the following way: electrons screen the field of a light wave if  $\omega < \omega_{\text{pe}}$ . Because of that,  $\omega_{\text{pe}} = \omega$  defines the maximum plasma density to which a light wave can penetrate - *critical density*:

$$n_{\text{cr}} = \frac{\omega^2 m_e}{4\pi e^2} = 1.1 \times 10^{21} / \lambda_\mu^2 \text{ cm}^{-3}, \quad (1.13)$$

where  $\lambda_\mu$  is the wavelength of the light in microns in vacuum [21].

The examples of plasmas of different densities and temperatures found in the real world can be seen in the table 1.1.

Type	Electron density	Electron temperature
	$n_e \text{ [}(\text{cm})^{-3}\text{]}$	$T_e \text{ [eV]}$
Stars	$10^{26}$	$2 \times 10^3$
Laser fusion	$10^{25}$	$3 \times 10^3$
Magnetic fusion	$10^{15}$	$10^3$
Laser-produced	$10^{18} - 10^{24}$	$10^2 - 10^3$
Discharges	$10^{12}$	$1 - 10$
Ionosphere	$10^6$	$1.0$
Interstellar medium	$1$	$10^{-2}$

Table 1.1: Densities and temperatures of various plasma types [13].

## 1.2 Ionization

Any substance can become plasma with the sufficient increase of its temperature. The threshold can vary, but usually can be found in the order of 1 eV, because any neutral atom binds the outer electron with a binding energy in order of 1 eV [38].

### Ionization mechanisms

There are several mechanisms which can be used to describe ionization. One can start with directly hitting the atoms with fast particles, but for that one would need a stream of such particles. More common way of ionization is achieved by electromagnetic radiation (photoionization) or even via electrical breakdown in strong electric fields [13]. For this thesis, the most relevant ionization is through electromagnetic radiation - in our case a laser.

Firstly, oscillating electromagnetic field makes free electrons oscillate and they can ionize other atoms via collisions. New free electrons freed by the collisions can then also hit other atoms and an avalanche if ionization events can develop.

There are also non-collisional mechanisms of ionization. Imagine field of hydrogen atom at Bohr radius  $a_B$  - the most probable distance of electron from the atomic nucleus:

$$a_B = \frac{\hbar^2}{m_e e^2} = 5.3 \times 10^{-9} \text{ cm} \quad (1.14)$$

where  $\hbar$  is the reduced Planck constant. The electric field for hydrogen  $E_H$  is then:

$$E_H = \frac{e}{a_B^2} \simeq 5.1 \times 10^9 \text{ V.m}^{-1}. \quad (1.15)$$

The corresponding so called *atomic intensity* for hydrogen  $I_H$  is [13]:

$$I_H = \frac{E_H^2}{8\pi} \simeq 3.51 \times 10^{16} \text{ W.cm}^{-2} \quad (1.16)$$

where  $c$  is the speed of light in vacuum.

It is reasonable to think that to ionize the hydrogen atom one needs  $I_L > I_H$ , where  $I_L$  is the intensity of the laser. In reality, the ionization occurs already for smaller laser intensities due to so called *multiphoton absorption* [13] and *quantum tunnelling* [38]. The first one can occur, because the electron can climb the virtual energy states one after another and it can get hit by next photon before it falls back to lower energy state [38]. The calculation of these transitions is non-trivial, because one has to solve time-dependant Schroedinger equation. The reader can find deeper analysis in [19].

The tunnelling effect is as well a consequence of the external electric field. The superposition of the electric field which binds the electron to the atom and the strong electric field of the laser results in conditions that allow the electron escape the potential well even if the electron energy is not higher than the threshold energy for instant ionization. Let  $V_H(r) = -\frac{C}{r}$  Coulomb potential of hydrogen nucleus, where in CGS unit  $C = e^2$ . The superposition with strong external field gives us:

$$V_F = V_H(r) + eE_{\text{ext}}(r) \quad (1.17)$$

Let  $E_{\text{ext}}(r) = -10^{10}r \text{ V/m}$ . The final potential  $V_F$  together with highlighted region of tunneling can be seen in figure 1.1.

The stronger is the external field, the shorter is the tunnelling distance for the electron to escape and the higher is the probability that this can happen. The field can even be so strong that the potential barrier will have its peak below the ground state energy. In that case, the electron is instantly considered to be free [38].

It is possible to estimate, which mechanism is more dominant cause of ionization by calculating so called *Keldysh parameter*  $\gamma_K = \frac{\omega_0}{\omega_t}$ , where  $\omega_0$  is the frequency of the laser and  $\omega_t = \frac{eE_{\text{ext}}}{\sqrt{2mE_i}}$ , where  $E_i$  represents the energy the electron needs to receive to be ionized [38].

The ionization processes can be explored in greater depth, but the fundamental concepts have already been adequately outlined. Henceforth, we will assume the plasma being targeted by the laser is fully ionized and will focus on how it can absorb additional energy from the laser.

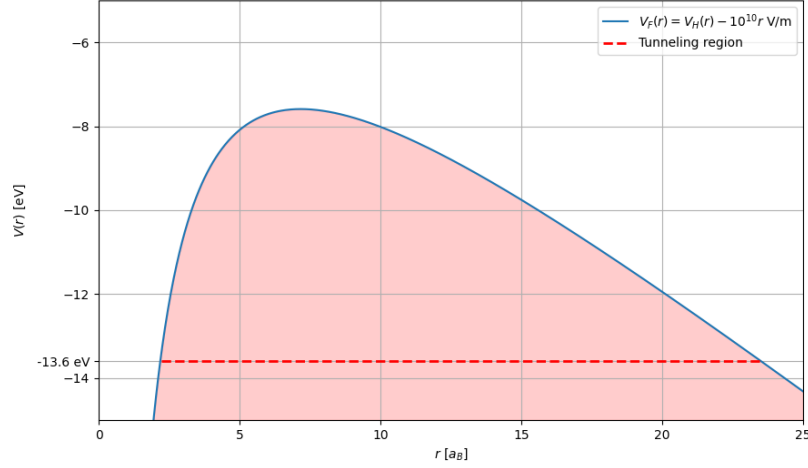


Figure 1.1: The potential of hydrogen atom modified by external field:  $E_{\text{ext}} = -10^{10} r \text{ V/m}$ .  $r$  is shown the radial coordinate normalized to Bohr radius  $a_B$ . Energy of ground state of electron in hydrogen atom  $E_0 = 13.6 \text{ eV}$  is highlighted.

### 1.3 Absorption of ultra-short, ultra-intense lasers

Modern lasers can generate pulses with durations of only few femtoseconds and extremely high intensities (up to  $10^{22} \text{ W.cm}^{-2}$ ) [43, 45]. The interaction of such pulses with dense plasma produces hot electrons [23]. There are several processes of energy transfer from the laser's electromagnetic field to the electrons. Let us examine the most significant of these processes.

#### Collisional absorption

The principles of collisional absorption are similar to those of collisional ionization. In this process, an electron oscillates due to the influence of the laser field and transfers part of its kinetic energy to other ions through collisions. However, since the frequency of ion-electron collisions scales as  $\nu_{ie} \propto E^{-3/2}$ , this absorption mechanism is primarily significant for laser intensities below  $10^{15} \text{ W.cm}^{-2}$  [35]. Given that this thesis focuses on intensities above this threshold, further discussion on collisional absorption is unnecessary.

#### Resonance absorption

The first non-collisional absorption process we will describe is the *resonance absorption*. This phenomenon can occur during the propagation of a p-polarized light wave through a density gradient. By p-polarized, we refer to a wave that is linearly polarized with its polarization vector lying in the plane of incidence. The complete

analytical description is difficult, but after few simplifications it is possible to obtain reasonable idea of the principle [24].

Laser light will reach density  $n_t = n_{cr} \cos^2 \theta$  (from Snell's law [43]), where  $\theta$  is the angle of incidence [24]. At this turning point, some light energy will tunnel through the critical density and the electron plasma will be resonantly excited at frequency of the laser  $\omega_0$ . The resonant wave is then capable of accelerating electrons and is defined by:

$$\frac{E_d}{\epsilon} = \frac{E_L}{\sqrt{2\pi\omega_0 L_n/c}} \phi(\tau) \quad (1.18)$$

where  $\epsilon$  is plasma dielectric function,  $L_n$  is the density length scale and the parameter  $\tau = (\omega_0 L_n/c)^{1/3} \sin \theta$  and  $\phi(\tau) \propto \exp(-2\tau^3/3)$  [24, 43].

The angle of optimum resonance absorption for exponential density profile  $\theta_{opt}$  can then be estimated as a function of  $L$ :

$$\theta_{opt}(L) = \arcsin(0.68(2\pi L)) \quad (1.19)$$

where  $L$  is normalized to the laser wavelength [35].

## Vacuum heating

The second non-collisional absorption process (and no less important than the first one) is called *Vacuum heating* or sometimes *Brunel effect* or even “*not-so-resonant resonance absorption*” [7]. It was proposed by Brunel in 1987 it was later confirmed by many experiments [43].

Like before, p-polarized laser pulse is needed. At the angle of incidence  $\theta$  the laser is hitting the target with steep density profile (a big gradient). A part of the pulse is reflected. The incoming laser wave  $\mathbf{E}_L$  and reflected wave  $\mathbf{E}_R$  are in superposition perpendicular to the target surface and the resulting field has a perpendicular component with amplitude  $E_0 = 2E_L \sin \theta$ , under approximation that  $E_R = E_L$ . Poisson's equation at the surface gives us [43]:

$$\Delta E = -4\pi e \int_{x=-\Delta x}^{x=0} n_e dx = 4\pi e n_e \Delta x. \quad (1.20)$$

The electrons are pulled out from plasma by the field  $E_0$ . If  $n = N/(A\Delta x)$  with  $N/A$  being number of electrons pulled out into vacuum per unit area  $A$  and if  $\Delta E = E_0$ , we get [43]:

$$\frac{N}{A} = \frac{2E_0 \sin \theta}{4\pi e}. \quad (1.21)$$

The energy absorbed  $E_{abs}$  by the electrons is then [43]:

$$E_{abs} = \frac{1}{2} N m_e v_e^2. \quad (1.22)$$

After calculating the power absorbed by unit area and after substituting  $v_e$  with *quiver velocity*  $v_{osc}$  defined by:  $\frac{v_{osc}}{c} = \frac{eE_0}{m_e c \omega_0}$ , we get the absorbed fraction of the power  $f_{VH} = I_{abs}/I_0$  [43]:

$$f_{VH} = 8 \frac{v_{osc}}{c} \sin^3 \theta \quad (1.23)$$

Note that we made a simplification by letting  $E_R = E_L$ . Another option would be to directly write  $E_0 = (1 + R^{1/2}) E_L \sin \theta$ , where  $R$  is the reflectivity [35]. We also neglected that the electron in vacuum is very fast and therefore relativistic correction has to be made. It is possible to generally follow a more rigorous path found for example in [23] and [35]. Then the formula for  $f_{\text{VH}}$  is expanded to:

$$f_{\text{VH}} = \frac{\eta}{2\pi} \frac{1}{a_0} \frac{\sin \theta}{\cos \theta} (1 + R^{1/2}) \left\{ \left[ 1 + (1 + R^{1/2})^2 a_0^2 \sin^2 \theta \right]^{1/2} - 1 \right\}, \quad (1.24)$$

where  $\eta = 1.74$  and  $a_0 = eE_L/(m_e \omega_0 c)$ .

## **$\mathbf{J} \times \mathbf{B}$ heating**

The last absorption mechanism we want to discuss is usually called  **$\mathbf{J} \times \mathbf{B}$  heating**. In the sections above, we discussed the heating of plasma due to electron motion in the direction of the oscillating  $\mathbf{E}$  component of the laser beam. That is of course caused by the  $e\mathbf{E}$  part of the Lorentz force. The other part -  $\mathbf{j} \times \mathbf{B}$  - can be neglected in non-relativistic cases. However, for laser intensities higher than  $10^{17} \text{ W.cm}^{-2}$  it is not possible to explain all absorption using the classical limit and another consideration has to be made [8].

Let  $\phi$  and  $\mathbf{A}$  be scalar and vector potential ( $\mathbf{E} = -\nabla\phi - \frac{\partial \mathbf{A}}{\partial t}$  and  $\mathbf{B} = \nabla \times \mathbf{A}$ ) satisfying Coulomb gauge  $\nabla \cdot \mathbf{A} = 0$ . Also, we can separate transverse and longitudinal part of electron momentum  $\mathbf{p} = \mathbf{p}_t + \mathbf{p}_l$ . Then the equations of motion for electron can be written as [8]:

$$\frac{\partial \mathbf{p}_t}{\partial t} = \frac{e}{c} \frac{\partial \mathbf{A}}{\partial t} \quad (1.25)$$

$$\frac{\partial \mathbf{p}_l}{\partial t} = e\nabla\phi - m_e c^2 \nabla(\gamma - 1) \quad (1.26)$$

where  $\gamma = \sqrt{1 + \frac{a_0^2}{2}}$  is the relativistic factor for linearly polarized light [43].  $a_0$  is the same as defined in equation 1.24, but here it has meaning of a constant normalizing vector field  $\mathbf{A}$ . To be more precise  $a_0 = |\mathbf{a}|$  and:

$$\mathbf{a} = \frac{e\mathbf{A}}{m_e c^2}. \quad (1.27)$$

The second term of the equation 1.26 is the relativistic ponderomotive force and we can write the ponderomotive potential  $U_p$  as:

$$U_p = (\gamma - 1)m_e c^2. \quad (1.28)$$

There is a force with frequency  $2\omega_0$  which will affect the electrons in longitudinal direction. One of the interpretations of this force can sound like this: Twice every laser period, streams of electrons are pushed into the target [8]. This causes the production of fast electrons.

It is important to note, that at higher intensity, the electron density can rise as a consequence of the zero-frequency pondermotive force. The change in density can cause the  $2\omega_0$  force component to be inefficient for hot electron generation. The density is also related the scale length of the target. For example for exponential density profile, the density decreases with the increase of scale length [8]. This means the  $\mathbf{J} \times \mathbf{B}$  heating is expected to play a greater role for bigger scale lengths. In other words, the initial plasma conditions need to be known to estimate the effects of  $\mathbf{J} \times \mathbf{B}$  heating.

One last note regarding  $\mathbf{J} \times \mathbf{B}$  heating. The relativistic factor  $\gamma$  has a different form in a case of circularly polarized laser. That leads to suppressing this kind of electron heating altogether [8].

The three mentioned mechanisms are by no means exhaustive when it comes to laser absorption. They are the three most relevant in the context of this work. There are other physical processes contributing to heating up the plasma especially when the parameters of the experiment change [35]. We will now move on to describe the particular motivation of the thesis.

## 1.4 Motivation - X-ray $K_\alpha$ emission

Because hot electrons accelerated by ultra-intense ultra-short laser pulse can carry significant (keV-MeV) energy, they can penetrate deeper into the ionized part of the target, where they can generate characteristic X-rays by K-shell ionization [32]. The X-ray spectrum is consisting of spectral lines (e.g.  $K_\alpha$ ) and continuous part X-ray from Bremsstrahlung. The uniqueness of this method of producing X-rays rests in the monochromatic spectrum of high energy X-ray photons within a short pulse synchronized with the laser pulse. The source is typically very small [27].

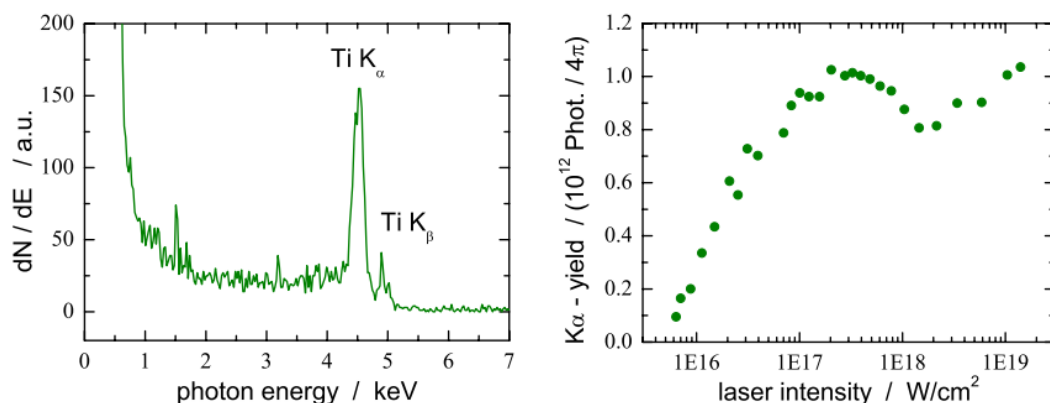


Figure 1.2: *Left:* The spectrum of laser generated  $K_\alpha$  and  $K_\beta$  radiation of titanium. *Right:* The scaling of  $K_\alpha$  - yield in relation to laser intensity. [34]

Let us now briefly look into the origin  $K_\alpha$ -line. In figure 1.2, there is an energy spectrum of titanium x-ray photons and the  $K_\alpha$  - yield scaling with the laser intensity. The generation of the  $K_\alpha$  radiation clearly depends on laser intensity through



the hot electrons temperature. According to [34], the yield drops at the laser intensity around  $10^{18} \text{ W.cm}^{-2}$  *"because the interaction time with the atom decreases with higher electron velocity"*. The following rise in the yield is then attributed to the relativistic effect where the electric fields of the fast hot electrons are contracted and therefore have greater effect [34].

The total yield  $N$  can be expressed analytically as [32]:

$$N_{\text{yield}} = \int n_{\text{hot}} f_{\text{hot}}(E) N_{\text{gen}}(E) f_{\text{em}}(E) dE \quad (1.29)$$

where  $N$  is the number of emitted photons,  $n_{\text{hot}}$  is the total number of hot electrons, and  $f_{\text{hot}}(E)$  is their energy distribution,  $N_{\text{gen}}(E)$  is the number of  $K_{\alpha}$  photons generated by an electron of incidence energy  $E$ , and  $f_{\text{em}}(E)$  is the fraction of these photons that escapes from the solid [32].

Having a reliable numerical model for  $n_{\text{hot}}$  and  $f_{\text{hot}}(E)$  from equation 1.29 based on the parameters of the laser and the plasma could allow us to optimize the  $K_{\alpha}$  yield. Namely, the angle of incidence, the laser intensity and the plasma length scale are the most important parameters for laser absorption. The research conducted by Reich et al. [32] could be followed up by examining a wider range of parameters beyond just laser intensity. This is crucial because, as shown by Cui et al. [10], the temperature function of the electrons has a complex and non-trivial shape. The complexity comes from the complex nature of the physical processes causing the electron heating.

In this thesis, we presents results from hundreds of PIC simulations while scanning through the mentioned parameters. Then we try to present and defend a way of how the hot electron temperature can be modelled and how to make the model more precise. The implications with respect to previous works and to equation 1.29 are discussed.

## 1.5 PIC simulations

As previously mentioned, we are using Particle-In-Cell (PIC) simulations to obtain the data necessary for our model. PIC codes have been under development since the advent of computers in the 1960s, and advancements in computer technology over the past 30 years have enabled us to run simulations on a much larger scale. One significant advantage of simulations is that they allow theoretical predictions to be verified in greater detail than is possible with real plasma experiments [11]. To illustrate the progress made over the decades, let us mention, that in 1962 Dawson and Buneman simulated the motion of 1000 plasma particles. Today, we can simulate the motion of more than  $10^{10}$  particles [40].

We are not developing our own simulation code. Instead, we are using code freely available for academic purposes, specifically the 2D version of simulation code EPOCH [2]. EPOCH has been widely used in numerous publications within the laser-plasma field and adequately meets our requirements. Below, we will present a

brief overview of the key principles of the PIC method, upon which EPOCH is also based.

## Macro-particles

It is not possible to have as many particles in a simulation as in a real plasma, even in very small scales, because of the computational cost. Because of that, the simulations usually work with macro-particles which represent clouds of many real particles. These particles have finite sizes (as opposed to infinitesimal), so that there are no divergent forces in case of collisions. The forces in simulations go to zero for small distances between macro-particles. For large distances, they comply with the Coulombic behaviour. In plasma simulations, it is possible to do this without sacrificing accuracy, because of the collective behaviour of plasma[33].

## Computational cycle

The simulation runs in a cycle. In each step, we solve for electromagnetic fields created by the charged particles. Then we evaluate the equations of motion for the particles, which are influenced by the Lorentz force [5]. The laser pulse is included as an external source of electromagnetic radiation at the boundary.

Typically, the finite-difference time-domain method (FDTD) is used for numerically solving Maxwell's equations, which fully describe the electromagnetic field. *Finite-difference* means that the electric field  $\mathbf{E}$  and magnetic field  $\mathbf{B}$  are specified in the points of a grid - usually a *Yee grid*. A comprehensive description of a Yee grid can be found in the original article by Yee [46]. The critical concept of a Yee grid is illustrated in figure 1.3. - the magnetic field components are calculated in the center of the faces of an imaginary cube (cell) while the electric field components are calculated in the center of the edges. The cube represents one cell of the 3-dimensional grid. We can approximate derivatives of electric field with centered finite differences [2]:

$$\left(\frac{\partial E_y}{\partial x}\right)_{i+\frac{1}{2},j,k} = \frac{E_{y,i+1,j,k} - E_{y,i,j,k}}{\Delta x} \quad (1.30)$$

Note, that this approximation is second order accurate at the cube point where we calculate  $B_{z,i,j,k}$ , because the formula is centered. Moreover, because of one of the Maxwell's equations,  $\nabla \times \mathbf{E} = \frac{\partial \mathbf{B}}{\partial t}$ , this derivative is exclusively used to calculate time-derivative of  $B_{z,i,j,k}$ . A similar relationship can be found when calculating all components of  $\mathbf{B}$  from  $\mathbf{E}$  and vice versa. Therefore, all used numerical derivatives are second order accurate [2].

In EPOCH, as in other PIC codes, fields are updated at both the half time-step and full time-step. The first part - time-step from  $n$  to  $n+1/2$  - uses currents calculated at  $n$ :

$$\mathbf{E}^{n+1/2} = \mathbf{E}^n + \frac{\Delta t}{2} \left( c^2 \nabla \times \mathbf{B}^n - \frac{\mathbf{J}^n}{\epsilon_0} \right) \quad (1.31)$$

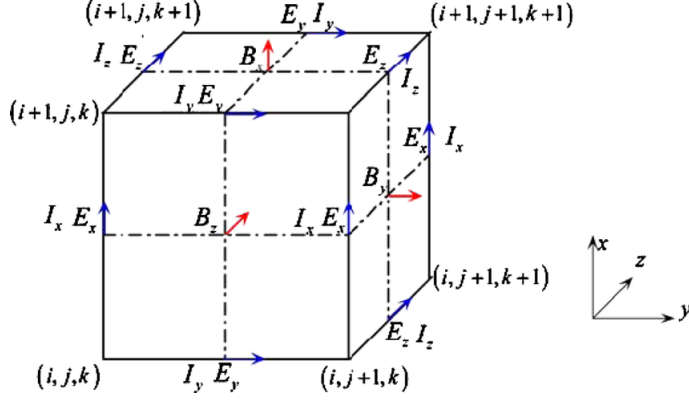


Figure 1.3: An illustration of a Yee grid [42]

$$\mathbf{B}^{n+1/2} = \mathbf{B}^n - \frac{\Delta t}{2} (\nabla \times \mathbf{E}^{n+1/2}) \quad (1.32)$$

where  $\mathbf{J}$  is the current density and  $\Delta t$  is a size of a full time-step [2].

In the second step, we use the updated currents at  $n + 1$ :

$$\mathbf{B}^{n+1} = \mathbf{B}^{n+1/2} - \frac{\Delta t}{2} (\nabla \times \mathbf{E}^{n+1/2}) \quad (1.33)$$

$$\mathbf{E}^{n+1} = \mathbf{E}^{n+1/2} + \frac{\Delta t}{2} \left( c^2 \nabla \times \mathbf{B}^{n+1} - \frac{\mathbf{J}^{n+1}}{\epsilon_0} \right) \quad (1.34)$$

The motion of particles and the resulting currents are consequences of the fields. Generally, the equations of motion are:

$$\frac{d\mathbf{x}_l}{dt} = \mathbf{v}_l \text{ and } \frac{d\mathbf{p}_l}{dt} = \mathbf{F}_l \quad (1.35)$$

where vectors  $\mathbf{x}_l$ ,  $\mathbf{v}_l$  and  $\mathbf{p}_l$  represent the position, velocity and momentum of the  $l$ -th macro-particle.  $\mathbf{F}_l = \mathbf{F}_l(t, \mathbf{x}_l, \mathbf{v}_l, \mathbf{E}, \mathbf{B})$  is the force. Fields  $\mathbf{E}$  and  $\mathbf{B}$  are functions of the positions and time [40]. In this context, the right-hand side of the second equation represents the Lorentz force, and the time-step formula for the momentum is: [2]:

$$\mathbf{p}_l^{n+1} = \mathbf{p}_l^n + q_l \Delta t \left[ \mathbf{E}^{n+1/2}(\mathbf{x}_l^{n+1/2}) + \mathbf{v}_l^{n+1/2} \times \mathbf{B}^{n+1/2}(\mathbf{x}_l^{n+1/2}) \right] \quad (1.36)$$

where  $q_l$  is the charge of the  $l$ -th particle. The velocity can be calculated from the momentum using:

$$\mathbf{v}_l = \frac{\mathbf{p}_l}{\gamma_l m_l} \quad (1.37)$$

where  $m_l$  is the mass of the particle and  $\gamma_l = [p_l^2/(m_l^2 c^2) + 1]^{1/2}$  is the corresponding gamma-factor [2].

The particle position update is calculated from the velocity, but this is also done in multiple steps. First, we calculate movement of half time-step from the old velocity as we need it to update the momentum in equation 1.36 [2]:

$$\mathbf{x}_l^{n+1/2} = \mathbf{x}_l^n + \frac{\Delta t}{2} \mathbf{v}_l^n \quad (1.38)$$

In similar way, we can then calculate  $\mathbf{x}_l^{n+1}$  and  $\mathbf{x}_l^{n+3/2}$  [2], which are needed for calculating the currents.

The currents necessary for updating the fields can be calculated using methods such as the one presented by Esirkepov in [12]. Modern approaches to calculating currents are generally based on solving the discrete form of the continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0 \quad (1.39)$$

where  $\rho$  is the charge density and  $\mathbf{J}$  is the electric current. This can be discretized as:

$$\frac{\rho_{i+1/2,j+1/2,k+1/2}^{n+1} - \rho_{i+1/2,j+1/2,k+1/2}^n}{\Delta t} + \frac{J_{x,i+1,j+1/2,k+1/2}^{n+1/2} - J_{x,i,j+1/2,k+1/2}^{n+1/2}}{\Delta x} + \dots = 0. \quad (1.40)$$

The charge density is calculated using *form-factors* of macro-particles:

$$\rho_{i,j,k} = \sum_l Q_l S_{i,j,k}(x_l, y_l, z_l) \quad (1.41)$$

where  $Q_l$  is the charge and  $S_{i,j,k}(x_l, y_l, z_l)$  is the form-factor. During macro-particle motion, the total charge should remain constant, necessitating that the form-factors satisfy the condition :

$$\sum_{i,j,k} S_{i,j,k}(x_l, y_l, z_l) = 1 \quad (1.42)$$

Since macro-particles in the simulation represent numerous real particles, a *weight* is assigned to each macro-particle. Although the exact particle distribution within a macro-particle is unknown, a representative function, known as a *shape function*, is chosen.[2].

Any function with a unit integral and compact support can be used as a shape function. An even distribution of particle in volume  $\Delta x \times \Delta y \times \Delta z$  is referred to as the *top hat* shape function. Using higher order shape functions are one of the improvements programmers were able to make thanks to more powerful computers. A higher order shape functions are for example triangular shape functions with volume of  $2\Delta x \times 2\Delta y \times 2\Delta z$  . The weight is then calculated as a convolution of shape function with the 'top hat' function [2].

Working with macro-particles instead of individual particles neglects some effects, particularly those effective over distances shorter than  $\Delta x$ . EPOCH uses a fully relativistic, energy-conserving binary collision model, which favors small-angle scattering to improve simulation behavior with a limited number of particles per cell. For our study involving laser intensities above  $10^{10}$ , collisions are not necessary. Other effects, such as ionization and quantum phenomena like photon emission and pair production, are often included when relevant, with detailed descriptions available in [2].

# Chapter 2

## Temperature fitting

The results of the EPOCH simulations provide weighted momenta for a subset electrons  $\mathbf{p}_e$ , where the weight represents the number of electrons with that momentum. They are not histograms yet, because more than one macro-particle can have the same value of  $\mathbf{p}_e$ . The energies of electrons can then be calculated from  $\mathbf{p}$  using the relativistic formula:

$$E_e = m_e \cdot c^2 \left( \sqrt{1 + \left( \frac{p}{m_e \cdot c} \right)^2} - 1 \right), \quad (2.1)$$

where  $p = \sqrt{\mathbf{p} \cdot \mathbf{p}}$  is the size of  $\mathbf{p}$ ,  $m_e = 9.109 \times 10^{-31}$  kg is the electron rest-mass and  $c = 3 \times 10^8$  m.s<sup>-1</sup> is the speed of light [**cite-the-constants**].

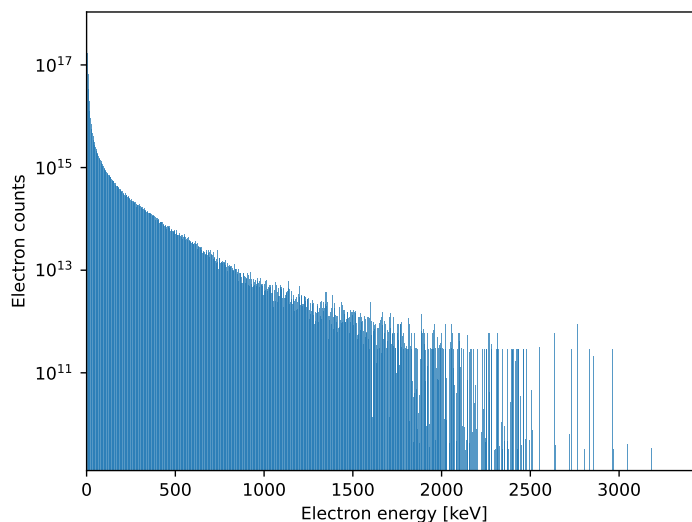


Figure 2.1: An example of electron energy distribution of 2D EPOCH simulation with intensity of laser  $I = 10^{19}$  W.cm<sup>-2</sup>, characteristic scale of the exponential preplasma profile  $L = 0.1 \mu\text{m}$  and angle of incidence with respect to target normal direction  $\alpha = 10^\circ$ .

One can easily create the histogram from electron energies and their counts. An example of such histogram can be seen in the figure 2.1. There are several things that need to be discussed.

Firstly, it is important to note that the  $y$ -axis is presented on a logarithmic scale to enhance readability. The extensive range of electron energies necessitates the use of relativistic formula. However, around the energies  $E_k \approx 1900$  keV and more, the histogram exhibits irregularities - specifically, there are empty bins and several bins contain identical electron counts. These anomalies are attributed to the resolution of the simulation. Consequently, this portion of the spectrum is rendered questionable. This issue will be further addressed in the section dedicated to temperature fitting.

Last but not least, note the apparent exponential relationship between  $N_i$  and  $E_k$  in the energy spectrum between  $E_k \approx 500$  keV and  $E_k \approx 1500$  keV:

$$N_i = N_0 \cdot \exp\left(-\frac{E_i}{T}\right), \quad (2.2)$$

where  $N_i$  is count in  $i$ -th bin and  $E_i$  is the corresponding energy and  $T$  is temperature in units discussed in section 1.1. The relationship is, of course, linear after the logarithmic transformation. The equation of 2.2 resembles exponential distribution with  $1/T$  missing on the right-hand side. It is also not normalized, because it represents counts that have to add up to total number of electrons with that temperature. For the purpose of this thesis, we will call it the *Boltzmann distribution*, even though it is not completely accurate.

Fitting the correct part of the histogram using equation 2.2 and estimating  $T$  and  $N_0$  is the first bigger part of this thesis. There are several obstacles, all of which are discussed in following sections.

## 2.1 Boltzmann vs. Maxwellian distribution

First of all, in section 1.1, where we introduced temperature as a quantity describing plasma, we said that the distribution of energies is usually considered to be Maxwellian. The question, whether we can fit our histogram using Boltzmann distribution, is therefore valid and needs to be addressed.

The equations for the Maxwellian  $f_M(E)$  and Boltzmann  $f_B(E)$  distributions can be after few simplifications regarding units defined as:

$$f_M(E)dE = \frac{\sqrt{E}}{(T)^{3/2}} \exp\left(-\frac{E}{T}\right) dE \quad (2.3)$$

and

$$f_B(E)dE = \frac{1}{T} \exp\left(-\frac{E}{T}\right) dE. \quad (2.4)$$

By ignoring that both are defined by the differential, taking logarithm of both  $f_M(E)$  and  $f_B(E)$  we get:

$$\bar{f}_M(E) = \ln(f_M(E)) = \ln\left(\frac{\sqrt{E}}{(T)^{3/2}}\right) + \left(-\frac{E}{T}\right) \quad (2.5)$$

and

$$\bar{f}_B(E) = \ln(f_B(E)) = \ln\left(\frac{1}{T}\right) + \left(-\frac{E}{T}\right). \quad (2.6)$$

These two equations describe the the distributions in a logarithmic scale. The first derivatives are then:

$$\frac{d\bar{f}_M}{dE} = -\frac{1}{2} \frac{T^{3/2}}{\sqrt{E}} \frac{1}{T^{3/2}\sqrt{E}} - \frac{1}{T} = -\frac{1}{2E} - \frac{1}{T} \quad (2.7)$$

and

$$\frac{d\bar{f}_B}{dE} = -\frac{1}{T}. \quad (2.8)$$

The equations 2.7 and 2.8 suggest that in the logarithmic scale the slopes of these distributions differ by  $-\frac{1}{2E}$ , which decreases with increasing  $E$ . In other words, the fit of  $T$  can be replaced by fitting slope  $s = -1/T$  of the histogram after logarithmic transformation. For large energies, the slopes of both discussed distributions are approximately equal.

Moreover, the assumption of Boltzmann distribution allows us to fit not only the temperature, but also  $N_0$ , which is the total number of electrons in that distribution of hot electrons. If we keep the form of the distribution as in equation 2.2, the total energy absorbed by the group of hot electrons  $E_{\text{tothot}}$  with temperature  $T_{\text{hot}}$  can be calculated as:

$$E_{\text{tothot}} = N_0 \cdot T_{\text{hot}}. \quad (2.9)$$

## 2.2 Exponential-sum fitting

Exponential-sum fitting has been a recognized challenge for many years. Forman S. Acton, a professor at Princeton University, addressed this issue in his 1970 book Numerical Methods That Work. Among various topics, he included an essay titled "What Not to Compute," wherein he described the fitting of sums of exponential functions as "extremely ill-conditioned" [1]. Despite the inherent difficulty of this problem, this thesis endeavors to determine electron temperatures, necessitating the development of an effective solution to this challenging computational task.

The task is to find the parameters  $m, a_0, a_1, \dots, a_m, b_1, \dots, b_m$  so that the function

$$f(x) = a_0 + \sum_{i=1}^m a_i e^{b_i x} \quad (2.10)$$

generalizes the data as good as possible. Formally, for a set of data ( $n$  data points)  $((x_1, y_1), \dots, (x_n, y_n))$  we are minimizing error function

$$e = \sum_{i=1}^n w_i (y_i - f(x_i))^2 \quad (2.11)$$

where  $w_i$  are the weights.

Modern programming libraries provide access to well-known and generally efficient methods, such as non-linear least squares. However, due to the previously mentioned ill-conditioning, these methods do not yield satisfactory results in this context. The convergence of these methods is highly dependent on the quality of the initial guess, which poses a significant challenge. To date, significant efforts have been made to develop reliable methods for solving this problem.

Notably, Prony’s method [29] and its variations, as discussed in sources such as [28], have been proposed. It was originally intended for signal approximation - specifically, the determination of complex parameters in exponential functions, similar to Fourier transformation.

Another, much simpler method is known as *successive subtraction* [44]. This method is frequently employed for exponential decays (negative parameters  $b_i$ ), which closely resemble our problem of exponential distributions. The core concept involves fitting the end of the decay with a single exponential function, subtracting this result from the data, and iteratively identifying all exponential components. This simplicity makes it a promising candidate for our problem, especially since our primary interest lies in determining the temperature of the hottest electrons—the tail. However, automating this method can be challenging, as it is not straightforward to select an interval that can be fitted as a tail in each iteration. What is more, the tail of our histograms are not reliable as was discussed with regard of the example histogram in figure 2.1.

Another method worth considering is presented in [44]. This method, called exponential sum fitting of transmissions (ESFT), was developed for fitting radiative transmission functions in the context of atmospheric research. ESFT too is an iterative method and might serve as an interesting alternative for future works.

We only mentioned a few methods. A summary with a deeper explanation of these and other methods can be found in [44], [16] or in [15].

If the goal is to fit only one of the exponentials, such as the parameters of the hot electrons distribution, the standard approach involves manually selecting the “correct” segment of the distribution and fitting it with a single exponential function. For instance, in [10], where researchers study a problem similar to ours, they appear to use this technique to determine the temperature of hot electrons. The term “correct” is somewhat vague because it can be subjective and we do not have any metric that would quantify how well the segment was chosen. We can only work with the metrics describing the fit itself.

The last mentioned option currently provides the most reliable approach to fitting the temperature of hot electrons. Even for hundreds of simulations, with the right tool, it is possible to fit manually every histogram within few hours. Having such dataset of histograms and temperatures is valuable for evaluating any method that is



trying to automate the fitting process. A tool with an UI was developed specifically for this purpose as a part of this work.

That being said, we still attempted to automate the fitting. The method we used is explicit an non-iterative.

## The 'Jacquelin' method

We will call the method we chose *Jacquelin method*, because we will be referencing an article by J.Jacquelin who developed it for his own purpose [18]. This method is explicit and non-iterative which is its main advantage. The original derivation presents a universal strategy how to explicitly approximate any function that is a solution to some integral or differential equation. Let us start with derivation of the numerical algorithm for a function:

$$y(x) = a_0 + a_1 e^{b_1 x} + a_2 e^{b_2 x} \quad (2.12)$$

We start by calculating these integrals:

$$\begin{aligned} S(x) &= \int_{x_0}^x y(t) dt \\ &= \int_{x_0}^x a_0 + a_1 e^{b_1 t} + a_2 e^{b_2 t} dt \\ &= a_0(x - x_0) + \frac{a_1}{b_1} (e^{b_1 x} - e^{b_1 x_0}) + \frac{a_2}{b_2} (e^{b_2 x} - e^{b_2 x_0}) \\ \\ SS(x) &= \int_{x_0}^x S(t) dt \\ &= \int_{x_0}^x \int_{x_0}^t y(u) du dt \\ &= \int_{x_0}^x a_0 t + \frac{a_1}{b_1} e^{b_1 t} + \frac{a_2}{b_2} e^{b_2 t} - \left( \frac{a_1}{b_1} e^{a_1 x_0} + \frac{a_2}{b_2} e^{a_2 x_0} + a_0 x_0 \right) dt \\ &= \frac{1}{2} a_0 x^2 - \left( \frac{a_1}{b_1} e^{a_1 x_0} + \frac{a_2}{b_2} e^{a_2 x_0} + a_0 x_0 \right) (x - x_0) - \frac{1}{2} a_0 x_0^2 \\ &\quad + \frac{a_1}{b_1^2} (e^{b_1 x} - e^{b_1 x_0}) + \frac{a_2}{b_2^2} (e^{b_2 x} - e^{b_2 x_0}) \end{aligned}$$

Now, if we reorganize the terms it can be shown that there are constants  $A$ ,  $B$ ,  $C$ ,  $D$  and  $E$  so that  $y(x)$  can be written as:

$$y(x) = A \cdot SS(x) + B \cdot S(x) + Cx^2 + Dx + E. \quad (2.13)$$

By comparing coefficients before the exponential terms  $e^{b_1 x}$  and  $e^{b_2 x}$  we get:

$$a_1 = A \frac{a_1}{b_1^2} + B \frac{a_1}{b_1} \quad (2.14)$$

$$a_2 = A \frac{a_2}{b_2^2} + B \frac{a_2}{b_2} \quad (2.15)$$

From that it is possible to see that  $b_1$  and  $b_2$  are roots of the quadratic equation:

$$b^2 - bB - A = 0. \quad (2.16)$$

$b_1$  and  $b_2$  are then:

$$b_1 = \frac{1}{2} \left( B + \sqrt{B^2 + 4A} \right) \quad (2.17)$$

$$b_2 = \frac{1}{2} \left( B - \sqrt{B^2 + 4A} \right) \quad (2.18)$$

The equation 2.13 is linear in the unknown parameters  $A, \dots, E$  and after discretization it can be rewritten as:

$$\mathbf{y} = \mathbf{X} \cdot \mathbf{b} \quad (2.19)$$

where  $\mathbf{y}$  is vector:

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad (2.20)$$

after we set  $S_k = S(x_k)$  and  $SS_k = SS(x_k)$  we can write:

$$\mathbf{X} = \begin{pmatrix} SS_1 & S_1 & x_1^2 & x_1 & 1 \\ SS_2 & S_2 & x_2^2 & x_2 & 1 \\ SS_3 & S_3 & x_3^2 & x_3 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ SS_n & S_n & x_n^2 & x_n & 1 \end{pmatrix} \quad (2.21)$$

and

$$\mathbf{b} = \begin{pmatrix} A \\ B \\ C \\ D \\ E \end{pmatrix} \quad (2.22)$$

and we can use the well-known least squares method to calculate the parameters  $A, \dots, E$ . We sort the data so that if  $i < j$ , then  $x_i < x_j$  for  $\forall i, j \in \{1, \dots, n\}$ . The integrals are computed numerically as:

$$S_i = \begin{cases} 0 & i = 0 \\ S_{i-1} + \frac{1}{2}(y_i + y_{i-1})(x_i - x_{i-1}) & i \in \{2, \dots, n\} \end{cases} \quad (2.23)$$

and

$$SS_i = \begin{cases} 0 & i = 0 \\ SS_{i-1} + \frac{1}{2}(S_i + S_{i-1})(x_i - x_{i-1}) & i \in \{2, \dots, n\} \end{cases} \quad (2.24)$$

Now, the solution to the linear equation 2.19 can be written as **[lin-reg]**:

$$\mathbf{b} = (\mathbf{X}^T \mathbf{X})^{-1} \cdot (\mathbf{X}^T \mathbf{y}) \quad (2.25)$$

which in our case can be expanded to:

$$\begin{pmatrix} A \\ B \\ C \\ D \\ E \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n SS_i^2 & \sum_{i=1}^n SS_i S_i & \sum_{i=1}^n SS_i x_i^2 & \sum_{i=1}^n SS_i x_i & \sum_{i=1}^n SS_i \\ \sum_{i=1}^n SS_i S_i & \sum_{i=1}^n S_i^2 & \sum_{i=1}^n S_i x_i^2 & \sum_{i=1}^n S_i x_i & \sum_{i=1}^n S_i \\ \sum_{i=1}^n SS_i x_i^2 & \sum_{i=1}^n S_i x_i^2 & \sum_{i=1}^n x_i^4 & \sum_{i=1}^n x_i^3 & \sum_{i=1}^n x_i^2 \\ \sum_{i=1}^n SS_i x_i & \sum_{i=1}^n S_i x_i & \sum_{i=1}^n x_i^3 & \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \\ \sum_{i=1}^n SS_i & \sum_{i=1}^n S_i & \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i & n \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^n SS_i y_i \\ \sum_{i=1}^n S_i y_i \\ \sum_{i=1}^n x_i^2 y_i \\ \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n y_i \end{pmatrix} \quad (2.26)$$

After we compute  $b_1$  and  $b_2$  from  $\mathbf{b}$ , we still have 3 unknown parameters  $a_0$ ,  $a_1$  and  $a_2$ . For those, we will take the original equation 2.12. As it is already linear in the parameters  $a_0$ ,  $a_1$  and  $a_2$ , we only need to pre-compute the values of  $\alpha_k = e^{b_1 x_k}$  and  $\beta_k = e^{b_2 x_k}$ . One can see that we get a very similar equation as 2.26, that can be written in this form:

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} n & \sum_{i=1}^n \alpha_i & \sum_{i=1}^n \beta_i \\ \sum_{i=1}^n \alpha_i & \sum_{i=1}^n \alpha_i^2 & \sum_{i=1}^n \alpha_i \beta_i \\ \sum_{i=1}^n \beta_i & \sum_{i=1}^n \alpha_i \beta_i & \sum_{i=1}^n \beta_i^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n \alpha_i y_i \\ \sum_{i=1}^n \beta_i y_i \end{pmatrix} \quad (2.27)$$

In [18], only the case without the constant  $a_0$  is presented, but as we have shown the derivation of the generalized version with constant is not difficult. An extension of Jacquelin method for more than two exponential terms is also not very complicated. Note that adding one term adds two parameters, which leads to matrix of size  $7 \times 7$ . However, it always leads to solving for the roots of the polynomial created by the first  $N$  elements of the vector  $\mathbf{b}$ , where  $N$  is the number of exponential terms. In case of two exponential terms, equations 2.17 solve for the roots of the polynomial:

$$b^2 - Bb - A = 0 \quad (2.28)$$

In case of three, we would need to calculate  $SSS(x)$  from  $SS(x)$ . There would also be a term  $x^3$ . If we set  $\mathbf{X}$  as  $[SSS(x), SS(x), S(x), x^3, x^2, x^1, 1]$ , we would solve for the roots of:

$$b^3 - Cb^2 - Bb - A = 0. \quad (2.29)$$

It is necessary to note, that this algorithm is prone numerical defects coming from the fact that there is a cumulative error related to the calculation of the partial sums. For that reason the resulting that one of matrices can be singular or the polynomial can have imaginary roots.

In [26], Mokhomomo et al. did numerical experiments where they have shown that similar method (derived from differential instead of integral equations) can be used

for fitting, but does not yield precise estimation results even for noiseless data. Nevertheless, it is an explicit algorithm for approximation of parameters which then can be used as initial guess for more advanced iterative techniques.

When it comes to uncertainty of the coefficients, there are two possibilities. First, it is possible to propagate the experimental error from  $y_i$  all the way to the final coefficients. The derivation can be found in [22]. However, we do not have the standard error of the histogram bins. The second option is to take the formula for linear regression calculate the error estimation from that.

# Chapter 3

## Surrogate hot electron temperature models

In this chapter, we will talk about the models suitable for hot electron temperature modelling. It was already explained in the first chapter that there are many physical processes contributing to the absorption of laser. As a consequence, there is a non-linear relationship between the initial parameters and the hot electron temperature. It would be very difficult to propose an analytical model where the temperature is explicitly dependant on all the initial parameters. There have been works studying how the temperature scales with respect to intensity [4, 10, 14, 20, 25], angle [10] or length scale and even pulse durations [25], but none of them models the dependency on all parameters at once. Needless to say, these works are valuable and they will be discussed with respect to predictions of our final model in chapter 6.

Given the complexity and multi-dimensionality of our dataset, few machine learning regression methods offer a promising alternative to analytical models. We will now discuss general principle of regression and several particular regression methods as well as their suitability for this thesis.

### 3.1 Regression methods

Regression is a fundamental technique in machine learning used for predicting continuous outcomes [6]. Formally, for the set of data  $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\} \in \mathcal{X} \times \mathbb{R}$ , where  $\mathcal{X}$  is  $d$ -dimensional space of inputs, we want to find a function  $f(\mathbf{x}, \mathbf{w})$  that represents the relationship between  $\mathbf{x}_i$  and  $y_i$ .  $\mathbf{w} = (w_0, \dots, w_m)$  is the vector of parameters of  $f$ . In the observations, we assume a Gaussian noise with zero mean and variance  $\sigma^2$  so we can write:

$$y_i = f(\mathbf{x}_i, \mathbf{w}) + \epsilon_i, \quad (3.1)$$

where  $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$  is the noise.

Moreover, we want the function  $f(\mathbf{x}, \mathbf{w})$  to be able to generalize - to give accurate predictions for inputs not present in the training dataset. A good regression process

balances the trade-off between fitting the training data accurately and maintaining predictive performance on unseen data. This condition transforms most regression problems to an optimization problem, where one tries to find the best model under certain conditions.

A good example is generalized linear regression where the model is given by:

$$y = w_0 + \sum_{i=1}^m \phi_i(\mathbf{x})w_i = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) \quad (3.2)$$

where  $\phi_i(\mathbf{x})$ ,  $i = 1, \dots, m$  are transformation functions of the inputs also called *basis functions* and  $\phi_0(\mathbf{x}) = 1$  [6]. A reasonable metric of the goodness of this model (sometimes also called *loss*) can be the sum of square of errors (*SSE*) calculated on dataset:

$$SSE(\mathbf{w}) = \frac{1}{2} \sum_{i=0}^n (\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) - y_i)^2 \quad (3.3)$$

Minimizing *SSE* with respect to weights in simple problems with correctly chosen  $\boldsymbol{\phi}$  is often enough to get a reliable predictor. This usually requires a good knowledge about the data. If the model is over-fitted - *SSE* is small for training data but large for new data - modifying the error function might help. For example, one can add *regularization* term  $\frac{\lambda}{2} \mathbf{w}^T \mathbf{w}$  to the *SSE*. This introduces *hyper-parameter*  $\lambda$ , which controls the importance of weights being small. The loss function to minimize is:

$$SSE_{\text{reg}}(\mathbf{w}) = \frac{1}{2} \sum_{i=0}^n (\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) - y_i)^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \quad (3.4)$$

Minimizing  $SSE_{\text{reg}}$  instead of *SSE* sometimes helps complex models to improve their predictive performance [6].

The optimization problem can be reversed to so called maximum likelihood estimation (MLE), where instead of minimizing an error function, one tries to find model maximizing the probability that we observe data given the set of parameters. In case of Gaussian noise, this approach is equivalent to minimizing *SSE* [6].

The appropriate selection of the feature mapping  $\boldsymbol{\phi}$  enables one to effectively capture non-linear relationships. However, in situations where determining the exact form of  $\boldsymbol{\phi}$  is ambiguous, various methodologies have been developed to accommodate modelling without explicit knowledge of the underlying model structure. In the following sections, we look into advanced techniques for non-linear regression. We begin with Support Vector Regression (SVR), followed by a brief overview of Neural Networks, and finally, a more detailed exploration of Gaussian Processes.

## 3.2 Support vector regression

Support vector regression (SVR) is powerful regression technique capable of modelling non-linear relationship without any previous assumption about the model. From its formulation it is directly set up to balance complexity and prediction error [47]. In several following paragraphs, we will present the most important principles of SVR.

## Linear $\epsilon$ -SVR model

Assuming simple linear model we can simplify equation 3.2 to  $y = \bar{\mathbf{w}}^T \mathbf{x} + w_0$  where  $\bar{\mathbf{w}} = (w_1, \dots, w_d)$ . In  $\epsilon$ -SVR, there are constraints put on the predictions in such way that the prediction  $\bar{\mathbf{w}}^T \mathbf{x}_i + w_0$  cannot be further from  $y_i$  than  $\epsilon$ . We also want the *tube* defined by  $\epsilon$  to be as *flat* possible, which can be done by minimizing  $\frac{1}{2} \mathbf{w}^T \mathbf{w}$  [47].

In reality, having rigid boundary at distance  $\epsilon$  from  $y_i$  is not practical because of outliers. It is therefore natural to introduce two *slack* variables  $\xi_i$  and  $\xi_i^*$ , which are used to widen the tube at the points of observation. The optimization problem can be formulated as [36]:

$$\text{minimize} \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n (\xi_i + \xi_i^*) \quad (3.5)$$

$$\text{subject to} \quad \begin{cases} y_i - (\bar{\mathbf{w}}^T \mathbf{x}_i + w_0) & \leq \epsilon + \xi_i \\ (\bar{\mathbf{w}}^T \mathbf{x}_i + w_0) - y_i & \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq 0 \end{cases} \quad (3.6)$$

## Kernel SVR

By so-called *kernel trick* - transforming features  $\mathbf{x}_i$  to a higher-dimensional kernel space  $\mathcal{F}$ , we can leave the assumption that the relationship between  $\mathbf{x}$  and  $y$  is linear. Let us note the mapping  $\phi(\cdot) : \mathcal{X} \rightarrow \mathcal{F}$ .  $\mathbf{x}_i$  in 3.5 and 3.6 is replaced by  $\phi(\mathbf{x})$ . Moreover, the new optimization problem can be written in dual form [36]:

$$\max_{\alpha_i, \alpha_i^*} \quad -\frac{1}{2} \sum_{i,j} (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) k(\mathbf{x}_i, \mathbf{x}_j) - \epsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i (\alpha_i - \alpha_i^*) \quad (3.7)$$

$$\text{subject to} \quad \sum_i (\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C] \quad (3.8)$$

where  $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \phi(\mathbf{x}_j)$  is the kernel function. There are multiple conditions which the kernel function must satisfy and they can be found in [36]. Popular kernels include linear, polynomial, radial basis function and others [47]. Note that  $\mathbf{w}$  is no longer explicitly present. Another important fact is that in this case, the flatness is important in the new space  $\mathcal{F}$  and not the original input space  $\mathcal{X}$  [36].

To determine the hyper-parameters  $\epsilon$  and  $C$  (or other parameters associated with the kernel itself), it is customary to perform a *cross-validation* where we randomly divide the dataset to two parts - training set and test set - and evaluate the predictive performance only on the latter which is not used during the optimization [47].

## 3.3 Neural networks

Neural networks have a different approach. They still assume almost nothing about the modelled relationship, but as opposed to SVR, where the goal is to minimize the

confidence interval by searching for a flat solution, neural networks usually search directly for minimum error on the training data [41].

The simplest neural network is probably the Multilayer Perceptron (MLP). A MLP is a type of neural network that consists of an input layer, one or more *hidden* layers, and an output layer. Each layer is composed of neurons, and each neuron in a layer is connected to every neuron in the subsequent layer. Neuron linearly combines all the outputs from the previous layer using its own set of weights and then transforms the result to its output using some function. The mathematical formulation of an MLP can be described as follows:

1. **Input Layer:** Let the input vector be  $\mathbf{x}$ .

2. **Hidden Layers:** Consider a single hidden layer with  $m$  neurons. The output of the  $j$ -th neuron in the hidden layer, denoted as  $h_j$ , is given by:

$$h_j = \sigma \left( \sum_{i=1}^n w_{ij}^{(1)} x_i + b_j^{(1)} \right) \quad (3.9)$$

where  $w_{ij}^{(1)}$  is the weight connecting the  $i$ -th input to the  $j$ -th hidden neuron,  $b_j^{(1)}$  is the bias term for the  $j$ -th hidden neuron, and  $\sigma$  is the *activation function*, that introduces non-linearity.

3. **Output Layer:** Let the output layer have 1 neuron. The output of the  $l$ -th neuron in the output layer, denoted as  $y_l$ , is given by:

$$y_l = \phi \left( \sum_{j=1}^m w_{jl}^{(2)} h_j + b_l^{(2)} \right) \quad (3.10)$$

where  $w_{jl}^{(2)}$  is the weight connecting the  $j$ -th hidden neuron to the  $l$ -th output neuron,  $b_l^{(2)}$  is the bias term for the  $l$ -th output neuron, and  $\phi$  is the activation function for the output layer (for regression  $\phi$  is the identity) [6].

Note that for each layer weights are well represented as matrix. Number of hidden layers and the number of neurons has to be chosen. However, even with only two layers with linear outputs, any continuous function on a compact domain can be approximated to arbitrary precision if the total number of neurons is large enough [6].

## Training the MLP

The training of neural network includes usually calculating gradients of the loss function with respect to the weights, propagating the gradients back through the network and updating the weights - also known as *gradient descent*. It is usually the case that if the loss and the activations are differentiable functions (w.r. to the weights), because their derivative is needed for the error back-propagation [6].

Gradient descent alone can find local minimum of the error function for the training data but it says nothing about the ability to recognize patterns. There are multiple *regularization* strategies that help neural network generalize but in contrast to SVR



they have to be fine-tuned. Common ones are early stopping (not letting the weights to come to the local minimum completely), weight decay (e.g. using  $SSE_{\text{reg}}$ ), batch normalization (the gradients are averaged for multiple training points), or dropout (in each learning process iteration we exclude few neurons from the network) [6, 37].

When training complex non-linear relationship on small datasets (up to 1000 samples), it can be the case that the complexity cannot be captured by network with less weights than the number of training points. This is called *over-parametrization*. However, it is sometimes still valid to train such network, because the effective number of degrees of freedom can be lowered by regularization [3, 17]. Tuning over-parametrized NNs is more difficult, but seeing its predictions side by side with other models can give us a better insight into the properties of our data.

### 3.4 Gaussian Processes

Another viable regression method is Gaussian Process regression. Its approach is completely different compared to both previous methods. Following paragraphs are explaining the basics of Gaussian Processes.

Gaussian Process (GP) can be understood as an extension of multivariate Gaussian distribution over vectors to distribution over functions. Informally, we replace  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ , where  $\boldsymbol{\mu}$  is mean and  $\boldsymbol{\Sigma}$  is the covariance matrix, with Gaussian Process  $\mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ , where  $m(\mathbf{x})$  is mean function and  $k(\mathbf{x}, \mathbf{x}')$  is covariance function.  $m(\mathbf{x})$  and  $k(\mathbf{x}, \mathbf{x}')$  fully specify a Gaussian process. For every input  $\mathbf{x}$  there is now a random variable  $f(\mathbf{x})$  for which we can write [30]:

$$f \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \quad (3.11)$$

Function  $k$  is commonly called *kernel* function as in SVR. Sampling from  $f$  can then be done easily, if we know  $k$  and  $m$  and if we realize that we can usually represent sample function by finite amount of points.

For sampling, we first calculate covariance matrix  $\Sigma_{i,j} = k(\bar{x}_i, \bar{x}_j)$  for points  $\bar{\mathbf{x}} = (\bar{x}_1, \dots, \bar{x}_s)$ , at which we want to plot the sampled functions. Each sample is then a vector of size  $s$  given by multivariate Gaussian distribution with covariance matrix  $\boldsymbol{\Sigma}$ . As an example, 5 samples from Gaussian Process with  $m(x) = 0$  and  $k(x, x') = \exp(-\frac{(x-x')^2}{2})$  can be seen in figure 3.1. Note that the smoothness is a consequence of the choice of the kernel function.

The next step is to update the  $\mathcal{GP}$  with the measured data. In other words, we want to condition  $f$  from 3.11 on dataset  $D$ . We can write [30]:

$$\begin{aligned} f|D &\sim \mathcal{GP}(m_D, k_D) \\ m_D(\mathbf{x}) &= m(\mathbf{x}) + \boldsymbol{\Sigma}_*(\mathbf{X}, \mathbf{x})^T \boldsymbol{\Sigma}^{-1}(\mathbf{f} - \mathbf{m}) \\ k_D(\mathbf{x}, \mathbf{x}') &= k(\mathbf{x}, \mathbf{x}') - \boldsymbol{\Sigma}_*(\mathbf{X}, \mathbf{x})^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Sigma}_*(\mathbf{X}, \mathbf{x}'), \end{aligned} \quad (3.12)$$

where  $\boldsymbol{\Sigma}_*(\mathbf{X}, \mathbf{x})$  is a vector of covariances between every training case  $\mathbf{X}$  and  $\mathbf{x}$ . Applying equation 3.12 to our previous example, we can generate new samples

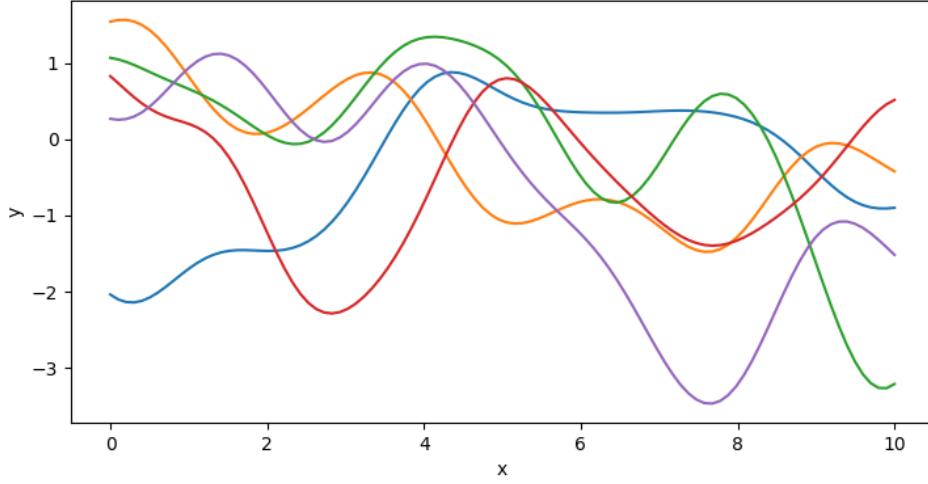


Figure 3.1: 5 sample functions from  $\mathcal{GP}(0, \exp(-\frac{(x-x')^2}{2}))$ . It was sampled at 100 points between  $x = 0$  and  $x = 10$ .

from the posterior distribution of functions. The term *posterior* comes from the Bayesian inference. If we let for example the training dataset consist of five points  $D = \{(1, 2), (3, 2), (5, 3), (7, 4), (9, 2)\}$ , the 5 samples from the updated distribution can be seen in figure 3.2.

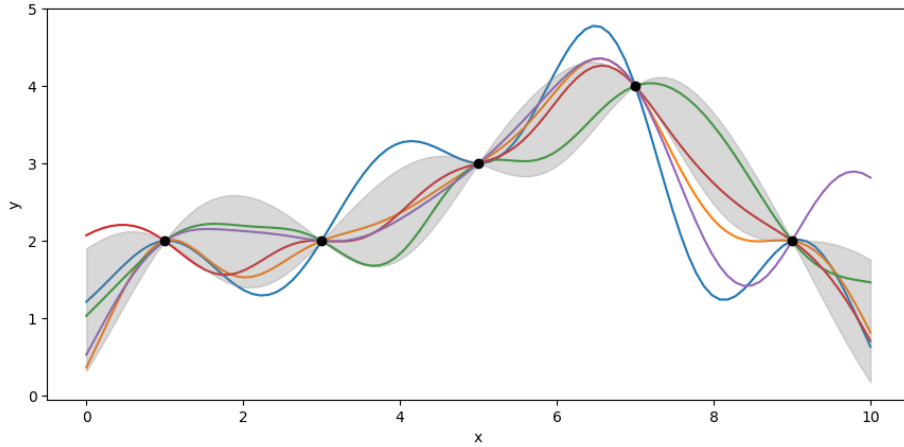


Figure 3.2: 5 sample functions from  $\mathcal{GP}(m_D, k_D)$ . It was sampled at 100 points between  $x = 0$  and  $x = 10$ . The grey area depicts one one standard deviation from the mean.

One can see that the figure 3.2 has sample sample functions that go through the points in  $D$ . This is because we set  $D$  as the condition. A less strict condition would be to set a non-zero noise to the dataset. This allows the samples not to go strictly through all points in the dataset. Mathematically speaking, adding noise is represented by adding a diagonal matrix to the covariance matrix or even more

general [30]:

$$\begin{aligned} y &= f + \epsilon, & \epsilon &\sim \mathcal{N}(0, \sigma_n^2) \\ f &\sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), & y &\sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}') + \delta_{xx'}\sigma_n^2), \end{aligned} \quad (3.13)$$

where  $\delta_{xx'} = 1$  for  $x = x'$  and  $\delta_{xx'} = 0$  for  $x \neq x'$ .

Apart from RBF kernel  $k_{\text{RBF}}(x - x') = \sigma^2 \exp(-\frac{(x-x')^2}{2l})$  another common choice are kernels from the Matern class [31]:

$$k_{\text{Matern}}(r) = \frac{2^{1-\nu}}{\gamma(\nu)} \left( \frac{\sqrt{2\nu}r}{l} \right)^\nu K_\nu \left( \frac{\sqrt{2\nu}r}{l} \right), \quad (3.14)$$

where  $\nu$  and  $l$  are positive paremeters,  $r = |\mathbf{x} - \mathbf{x}'|$ ,  $\Gamma(\nu)$  is a gamma function and  $K_\nu$  is a modified Bessel function. Common choices of  $\nu$  are  $\frac{1}{2}$ ,  $\frac{3}{2}$  and  $\frac{5}{2}$ . For  $n \rightarrow \infty$ ,  $k_{\text{Matern}}(|\mathbf{x} - \mathbf{x}'|)$  degenerates to  $k_{\text{RBF}}(\mathbf{x}, \mathbf{x}')$  [31].

Gaussian process regression is a non-parametric model, meaning the training data cannot be discarded after the training, because they are needed for the computation of  $\Sigma_*$ . The memory complexity of the algorithm is  $O(n^2)$  and the time complexity is  $O(n^3)$ , because of the costly matrix inversion needed for computation of the posterior distribution [30]. For small datasets, however, this is manageable by standard computers.

Apart from that, the tuning of the hyper-parameters such as  $\sigma$ , the choice of the covariance function or other parameters is usually done using maximum-likelihood methods [30].

To the contrast of previous two models, which assume nothing about the data, in GP it is helpful to specify the prior mean function. It can be specified parametrically and the parameters can be tuned during hyper-parameter tuning. Having prior mean  $m(\mathbf{x}) = 0$  makes the model more simple but can introduce bias in the predictions if it is not handled. Common technique of dealing with this, implemented for example in the GPy library, is to normalize  $y$  values so that they have zero mean [39]. The transformation is reversed when the model returns non-biased predictions even if prior mean functions was zero.

For us, it is particularly interesting to have the possibility of calculating the uncertainty of the prediction as can be seen in the figure 3.2. Of course, it is mainly related to the density of the dataset in the studied domain. However, knowing the model uncertainty as function of input parameters can be leveraged for selection of parameters for new simulations with the goal of making the model more precise by expanding the training data.

# Chapter 4

## Dataset

In this chapter we will describe in detail, what simulations did we run and how we proceeded with the automated temperature fitting to obtain the dataset for surrogate temperature modelling.

### 4.1 EPOCH Simulations

The EPOCH simulation was run in following setting:

- The intensities of the laser:  $I = 10^{17} \text{ W.cm}^{-2}$ ,  $I = 10^{18} \text{ W.cm}^{-2}$  and  $I = 10^{19} \text{ W.cm}^{-2}$ .
- The angle of incidence with respect to target normal direction:  
 $\alpha \in \{0^\circ, 1^\circ, 2^\circ, 3^\circ, 4^\circ, 5^\circ, 10^\circ, 20^\circ, 30^\circ, 40^\circ, 45^\circ, 50^\circ, 60^\circ\}$ .
- The characteristic scale length of the preplasma:  
 $L \in \{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5\}$  in microns.
- The laser wavelength is 1 micron.
- The laser is p-polarized and focused to a Gaussian spot of size 3.2 microns.
- The density in the target ranges from about  $0.01n_c$  to  $3.\gamma_{osc}n_c$ , where  $n_c$  is critical density of laser radiation [10] and  $\gamma_{osc}$  is defined in [10].
- The initial temperature of plasma is 100 eV.
- The target is composed of electrons and protons. They are represented by 30 macro-particles per cell.
- They are represented by 30 macro-particles per cell. The resolution of spatial grid is 33nm and the time step satisfies the CFL condition [2].

The simulations have been performed on the Q3 node of the Quantum Hyperion cluster at FNSPE. The input file that used to start the EPOCH simulation can be seen in appendix A.

## 4.2 Temperature fitting

The results of the simulations are transformed into histograms as it was described in chapter 2. We fixed the histogram size to 1000 bins with their width scaling with the maximum electron energy. In several following pages, we will describe a strategy we used to find the temperature from all energy histograms. The effectiveness of this strategy varies for different histograms, because it depends on multiple properties.

To get the best possible hot electron temperature fit, three things are important. First, because of the imperfections of the histogram related to the beginning and the end of the energy spectrum, it is helpful to narrow down the fitted region. Secondly, it is still necessary to perform a good fit automatically ideally without many issues. Last but not least, the strategy has to take into account that the energy range and temperatures can for different histograms from the same dataset vary by several orders.

### Fitting strategy

Before fitting the data, it is essential to prepare it appropriately. In practice, this involves removing a few bins in the beginning of the electron energy spectrum. The lower energy bins can be removed, because they do not contribute to hot electrons temperature very much. Also, they can contain error, as the low energy electrons need more time to reach the virtual detector and the simulation can end before that happens. This is usually the case for histograms from simulations with lower laser intensity. Thirdly, we made an approximation by considering Boltzmann distribution which does not work for small energies very well. Cutting the beginning is therefore justified.

We also cut the end of the spectrum for reasons discussed in chapter 2. We cut it in a place where there is a first empty bin. The reason is that it is not practical to work with empty bins in the logarithmically transformed version of the histogram. In special cases, it would be possible to work histograms cut with less strict approach. These cut-offs ensure that the analysis focuses on the relevant and meaningful parts of the spectrum.

The result of the cut-off of histogram 2.1 can be seen in figure 4.1. Notice that the data show non-symmetrical noise in the part of the spectrum with the highest energies. This can be attributed to the logarithmic scale which can skew and seemingly magnify the noise for lower electron counts.

For the purpose of the fitting, it is suitable to approximate the histogram data as points with  $x$  values equal to the centre of the corresponding histogram bin and  $y$  value equal to count.

The Jacquelin method was implemented in programming language python as a class with number of exponential terms as a parameter. However, the lack of the numerical stability for more exponential terms usually causes issues for more than three terms because of reasons discussed at the end of chapter 2.

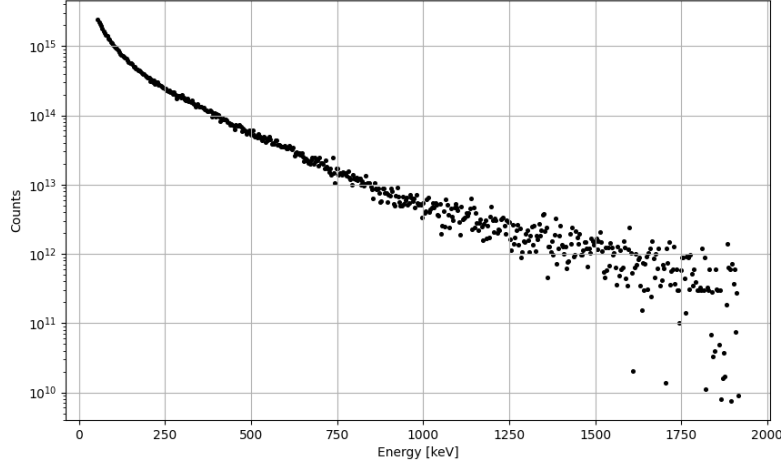


Figure 4.1: An example of trimmed histogram with simulation parameters  $I = 10^{19} \text{ W.cm}^{-2}$ ,  $L = 0.1 \mu\text{m}$  and  $\alpha = 10^\circ$ .

As we said multiple times before, apart from the hot electron distribution, the other present distributions are not important to us. It is reasonable to claim, that the quality of fit is improved, if the fitted part is as small as possible but still containing the section where hot electrons dominate.

That being said, the next step aim to cut (or rather ignore) the beginning histogram once again, but now in more sophisticated way. We found out that the vast majority of the histograms can be fit by the three-exponential Jacquelin method. Even though some do not have three distributions present, the fit does not fail and gives us valuable information.

The basic idea of this step can be seen in figure ???. We fit the histogram using the three-exponential Jacquelin method. There is one exponential most dominant for small energies. Corresponding electron distribution temperature is the lowest of all three distributions. We identify the energy  $E_{\text{crit}}$  at which the terms are equal using equation:

$$E_{\text{crit}} = \frac{\ln(a_1/a_2)}{b_2 - b_1} \quad (4.1)$$

and use this  $E_{\text{crit}}$  as threshold for the cut-off.

## Chapter 5

### Hot electron temperature modelling

## Chapter 6

### Comparison to contemporary data



# Conclusion

# Bibliography

- [1] F. S. Acton, *Numerical Methods That Work*. The Mathematical Association of America, 1990, pp. 245–257, ISBN: 0-88385-450-3.
- [2] T. D. Arber *et al.*, “Contemporary particle-in-cell approach to laser-plasma modelling”, *Plasma Physics and Controlled Fusion*, vol. 57, 11 Sep. 2015, ISSN: 13616587. DOI: 10.1088/0741-3335/57/11/113001.
- [3] P. L. Bartlett, “The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network”, *IEEE TRANSACTIONS ON INFORMATION THEORY*, vol. 44, pp. 525–536, 2 1998. DOI: 10.1109/18.661502.
- [4] F. N. Beg *et al.*, “A study of picosecond laser-solid interactions up to  $10^{19}$  w  $\{cm\}^{\dagger} - 2\}$ ”, *Physics of Plasmas*, vol. 4, pp. 447–457, 2 Feb. 1997, ISSN: 1070-664X. DOI: 10.1063/1.872103.
- [5] C. K. Birdsall and A. B. Langdon, *Plasma physics via computer simulation*. McGraw-Hill, 1985, ISBN: 0070053715.
- [6] C. M. Bishop, *Pattern Recognition and Machine Learning*, M. Jordan, J. Kleinberg, and B. Scholkopf, Eds. Springer, 2006, ISBN: 0-387-31073-8.
- [7] F. Brunel, “Not-so-resonant, resonant absorption”, *Physical Review Letters*, vol. 59, pp. 52–55, 1 Jul. 1987, ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.59.52.
- [8] H.-b. Cai, W. Yu, S.-p. Zhu, and C.-y. Zheng, “Short-pulse laser absorption via  $j \times b$  heating in ultrahigh intensity laser plasma interaction”, *Physics of Plasmas*, vol. 13, 11 Nov. 2006, ISSN: 1070-664X. DOI: 10.1063/1.2372463.
- [9] F. F. Chen, *Introduction to plasma physics and controlled fusion*. Plenum Press, 1984, ISBN: 0306413329.
- [10] Y. Q. Cui, W. M. Wang, Z. M. Sheng, Y. T. Li, and J. Zhang, “Laser absorption and hot electron temperature scalings in laser-plasma interactions”, *Plasma Physics and Controlled Fusion*, vol. 55, 8 Aug. 2013. DOI: 10.1088/0741-3335/55/8/085008.
- [11] J. Dawson, “One-dimensional plasma model”, *Physics of Fluids*, vol. 5, pp. 445–459, 4 1962. DOI: 10.1063/1.1706638.
- [12] T. Esirkepov, “Exact charge conservation scheme for particle-in-cell simulation with an arbitrary form-factor”, *Computer Physics Communications*, vol. 135, pp. 144–153, 2001. DOI: 10.1016/S0010-4655(00)00228-9. [Online]. Available: [www.elsevier.nl/locate/cpc](http://www.elsevier.nl/locate/cpc).

- [13] P. Gibbon, “Introduction to plasma physics”, CERN, 2014, pp. 51–65, ISBN: 9789290834250. DOI: 10.5170/CERN-2016-001.51.
- [14] M. G. Haines, M. S. Wei, F. N. Beg, and R. B. Stephens, “Hot-electron temperature and laser-light absorption in fast ignition”, *Physical Review Letters*, vol. 102, 4 Jan. 2009, ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.102.045008.
- [15] J. M. Hokanson, *Numerically stable and statistically efficient algorithms for large scale exponential fitting*, Aug. 2013. [Online]. Available: <https://hdl.handle.net/1911/102224>.
- [16] K. Holmström and J. Petersson, “A review of the parameter estimation problem of fitting positive exponential sums to empirical data”, *Applied Mathematics and Computation*, vol. 126, pp. 31–61, Feb. 2002. DOI: 10.1016/S0096-3003(00)00138-7. [Online]. Available: [https://doi.org/10.1016/S0096-3003\(00\)00138-7](https://doi.org/10.1016/S0096-3003(00)00138-7).
- [17] S. Ingrassia and I. Morlini, “Neural network modeling for small datasets”, *Technometrics*, vol. 47, pp. 297–311, 3 Aug. 2005. DOI: 10.1198/004017005000000058.
- [18] J. Jacquelin, *Regressions et equations integrales*, 2014. [Online]. Available: <https://www.scribd.com/doc/14674814/Regressions-et-equations-integrales>.
- [19] C. J. Joachain, N. J. Kylstra, and R. M. Potvliege, *Atoms in Intense Laser Fields*. Cambridge University Press, Dec. 2011, ISBN: 9780521793018. DOI: 10.1017/CB09780521793018.
- [20] T. Kluge, T. Cowan, A. Debus, U. Schramm, K. Zeil, and M. Bussmann, “Electron temperature scaling in laser interaction with solids”, *Physical Review Letters*, vol. 107, 20 Nov. 2011, ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.107.205003.
- [21] W. L. Kruer, *The Physics Of Laser Plasma Interactions*. CRC Press, Aug. 2019, ISBN: 9781003003243. DOI: 10.1201/9781003003243.
- [22] P. Lecca, M. Lecca, C. A. Maestri, and M. Scarpa, “Biexponential fitting for noisy data with error propagation”, *Mathematical Methods in the Applied Sciences*, vol. 44, pp. 10 154–10 171, 13 Sep. 2021, ISSN: 0170-4214. DOI: 10.1002/mma.7396.
- [23] A. Macchi, *A Superintense Laser-Plasma Interaction Theory Primer*, 1st ed. Springer Dordrecht, 2013, ISBN: 978-94-007-6124-7. DOI: 10.1007/978-94-007-6125-4. [Online]. Available: <http://www.springer.com/series/8902>.
- [24] P. Michel, *Introduction to Laser-Plasma Interactions*. Springer International Publishing, 2023, ISBN: 978-3-031-23423-1. DOI: 10.1007/978-3-031-23424-8.
- [25] K. G. Miller, D. R. Rusby, A. J. Kemp, S. C. Wilks, and W. B. Mori, “Maximizing mev x-ray dose in relativistic laser-solid interactions”, *Physical Review Research*, vol. 5, 1 Mar. 2023, ISSN: 2643-1564. DOI: 10.1103/PhysRevResearch.5.L012044.

- [26] M. Mokhomo and N. J. Matjelo, “Fitting sum of exponentials model using differential linear regression”, *International Research Journal of Engineering and Technology*, 2021, ISSN: 2395-0072. [Online]. Available: [www.irjet.net](http://www.irjet.net).
- [27] T. Pfeifer, C. Spielmann, and G. Gerber, “Femtosecond x-ray science”, *Reports on Progress in Physics*, vol. 69, pp. 443–505, 2 Feb. 2006, ISSN: 0034-4885. DOI: 10.1088/0034-4885/69/2/R04.
- [28] D. Potts and M. Tasche, “Parameter estimation for exponential sums by approximate prony method”, *Signal Processing*, vol. 90, pp. 1631–1642, 5 May 2010. DOI: 10.1016/j.sigpro.2009.11.012.
- [29] G. R. B. Prony, “Essai exp´erimental et analytique sur les lois de la dilatabilit´e de fluides ´elastiques et sur celles de la force expansion de la vapeur de l’alcool, ‘a diff´erentes temp´eratures.”, *Journal de l’Ecole Polytechnique (Paris)*, vol. 1, pp. 24–76, 1795.
- [30] C. E. Rasmussen, “Gaussian processes in machine learning”, in Springer, 2004, pp. 63–71. DOI: 10.1007/978-3-540-28650-9\_4.
- [31] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, Nov. 2005, ISBN: 9780262256834. DOI: 10.7551/mitpress/3206.001.0001.
- [32] C. Reich, P. Gibbon, I. Uschmann, and E. Förster, “Yield optimization and time structure of femtosecond laser plasma k-alpha sources”, *Physical Review Letters*, vol. 84, pp. 4846–4849, 21 May 2000, ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.84.4846.
- [33] F. Ricardo, “Computational challenges in laser-plasma interactions”, in D. A. Jaroszynski, R. Bingham, and R. Cairns, Eds. CRC Press, Mar. 2009, ISBN: 9780429146954. DOI: 10.1201/9781584887799.
- [34] H. Schwöerer, “Generation of x-rays by intense femtosecond lasers”, in Sep. 2004, pp. 235–254. DOI: 10.1007/978-3-540-39848-6\_17.
- [35] Z.-M. Sheng *et al.*, “Absorption of ultrashort intense lasers in laser–solid interactions”, *Chinese Physics B*, vol. 24, 1 Jan. 2015, ISSN: 1674-1056. DOI: 10.1088/1674-1056/24/1/015201.
- [36] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression”, *Statistics and Computing*, vol. 14, pp. 199–222, 3 Aug. 2004, ISSN: 0960-3174. DOI: 10.1023/B:STC0.0000035301.49549.88.
- [37] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting”, *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 1 2014.
- [38] H. Takabe, *The Physics of Laser Plasmas and Applications - Volume 1*. Springer, 2020. [Online]. Available: <http://www.springer.com/series/15614>.
- [39] The, GPY, and Authors, *{Gpy}: A gaussian process framework in python*, 2014. [Online]. Available: <http://github.com/SheffieldML/GPy>.
- [40] D. Tskhakaya, K. Matyash, R. Schneider, and F. Taccogna, “The particle-in-cell method”, *Contributions to Plasma Physics*, vol. 47, pp. 563–594, 8-9 2007. DOI: 10.1002/ctpp.200710072.

- [41] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer New York, 2000, ISBN: 978-1-4419-3160-3. DOI: 10.1007/978-1-4757-3264-1.
- [42] J. Wang *et al.*, “Three-dimensional parallel unipic-3d code for simulations of high-power microwave devices”, *Physics of Plasmas*, vol. 17, 7 Jul. 2010, ISSN: 1070664X. DOI: 10.1063/1.3454766.
- [43] S. Wilks and W. Kruer, “Absorption of ultrashort, ultra-intense laser light by solids and overdense plasmas”, *IEEE Journal of Quantum Electronics*, vol. 33, pp. 1954–1968, 11 1997, ISSN: 00189197. DOI: 10.1109/3.641310.
- [44] W. J. Wiscombe and J. W. Evans, “Exponential-sum fitting of radiative transmission functions”, *JOURNAL OF COMPUTATIONAL PHYSICS*, vol. 24, pp. 416–444, 4 1977. DOI: 10.1016/0021-9991(77)90031-6.
- [45] V. Yanovsky *et al.*, “Ultra-high intensity- 300-tw laser at 0.1 hz repetition rate”, *Optics Express*, vol. 16, p. 2109, 3 2008, ISSN: 1094-4087. DOI: 10.1364/OE.16.002109.
- [46] K. S. Yee, “Numerical solution of initial boundary value problems involving maxwell’s equations in isotropic media”, *IEEE Transactions on Antennas and Propagation*, vol. 14, pp. 302–307, 3 May 1966. DOI: 10.1109/TAP.1966.1138693.
- [47] F. Zhang and L. J. O’Donnell, “Support vector regression”, in Elsevier, 2020, pp. 123–140. DOI: 10.1016/B978-0-12-815739-8.00007-9.

# Appendix A

## EPOCH input file

```
begin:constant
konstanta1 = 1e19
konstanta2 = 2.0
konstanta3 = 60

las_lambda    = 0.8 * micron
l1            = las_lambda
las_intensity_focus_w_cm2 = konstanta1
a0            = l1 / micron * sqrt(las_intensity_focus_w_cm2/1.37e18)
gamma0        = sqrt(1.0+a0^2)
pulse_length  = 30.0 * femto
spot_size     = 3.2 * micron
cpwl          = 30
simtime       = 800.0 * femto
theta         = konstanta3 / 180.0 * pi
ct            = cos(theta)
st            = sin(theta)
n_el_over_nc  = 3.0*gamma0
min_density   = 0.01
preplasma_length = konstanta2 * micron
foil_thickness = 4.0 * micron
profile_thickness = - loge(min_density) * preplasma_length
boundary_thickness = 1.0 * micron
x1 = 60 * micron - foil_thickness -
    profile_thickness - boundary_thickness
x2 = x1 + profile_thickness
x3 = x2 + foil_thickness
xfocus        = x2
yc            = - xfocus * tan(theta) * 0.5
sigma_to_fwhm = 2.0 * sqrt(log(2.0))
w0            = spot_size / sigma_to_fwhm
zR            = pi * w0^2 / l1
lfocus        = xfocus / ct
lfzr          = lfocus / zR
w1            = w0 * sqrt(1.0+lfzr^2)
intensity_fac2d = 1.0 / sqrt(1.0+lfzr^2) * ct
las_omega     = 2.0 * pi* 299792458.0 / l1
las_time      = 2.0 * pi / las_omega
wt            = pulse_length / sigma_to_fwhm
```

```

n_crit      = critical(las_omega)
n_el       = n_el_over_nc * n_crit
y_full     = 54.0 * micron
end:constant

begin:control
x_min = 0.0 * micron
x_max = 60.0 * micron
y_min = -55.0 * micron
y_max = 55.0 * micron
nx = (x_max-x_min) / ll * cpwl
ny = (y_max-y_min) / ll * cpwl
t_end = simtime
particle_tstart = x1 / ct / 299792458.0
dt_multiplier = 0.97
dlb_threshold = 0.5
use_random_seed = T
stdout_frequency = 10
smooth_currents = T
field_order = 2
end:control

begin:boundaries
cpml_thickness = 6
cpml_kappa_max = 20
cpml_a_max = 0.15
cpml_sigma_max = 0.7
bc_x_min = cpml_laser
bc_x_max_field = cpml_outflow
bc_y_min_field = periodic
bc_y_max_field = periodic
bc_x_min_particle = heat_bath
bc_x_max_particle = heat_bath
bc_y_min_particle = heat_bath
bc_y_max_particle = heat_bath
end:boundaries

begin:laser
boundary = x_min
intensity_w_cm2 = las_intensity_focus_w_cm2 * intensity_fac2d
lambda = ll
t_profile = gauss(time, pulse_length, wt)
t_end = 3.0 * pulse_length
phase = -2.0*pi*(y-yc)*st/ll+((pi/ll)*(((y-yc)*ct)^2))/
        (lfocus*(1.0+(1.0/lfzr)^2))-atan((y-yc)*st/zR+lfzr)
profile = gauss((y-yc)*ct,0.0,wl)
end:laser

begin:species
name = electron
charge = -1.0
mass = 1.0
dump = T
npart_per_cell = 30
density = if((x gt x1) and (x lt x2) and (abs(y) lt y_full),
            n_el*exp((x-x2)/preplasma_length), 0.0)

```

```

density = if((x gt x2) and (x lt x3) and (abs(y) lt y_full), n_el,
    density(electron))
temp_ev = 100.0
identify:electron
end:species

begin:species
name = proton
charge = 1.0
mass = 1836.0
dump = T
npart_per_cell = 30
density = density(electron)
temp_ev = 100.0
end:species

begin:probe
name = electron_back_probe
point = (x3-2.0*micron, 0.0)
normal = (1.0, 0.0)
ek_min = 1.602e-16
ek_max = -1.0
include_species:electron
dumpmask = always
end:probe

#begin:output
# dt_snapshot = 10.0*femto
# name = fields
# file_prefix = f
# restartable = F
# grid = always + single
# ex = always + single
# ey = always + single
# bz = always + single
#end:output

#begin:output
# dt_snapshot = 10.0*femto
# name = plasma
# file_prefix = n
# restartable = F
# grid = always
# number_density = always + single
# poynt_flux = always + single
# absorption = always
# total_energy_sum = always
#end:output

begin:output
dt_snapshot = 700.0*femto
name = prob
file_prefix = ppr
restartable = F
particle_probes = always
end:output

```