Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering
Department of Physical Electronics

The effect of macroparticle number on particle-in-cell simulation results and computational demands in laser plasma physics

MASTER'S THESIS

Author: **Viktor Kocur**

Supervisor **Ing. Jan Pšikal, Ph.D.**

Academic year: 2016/2017

- Zadání práce -

- Zadání práce (zadní strana) -

Acknowledgment:
I would like to express my gratitude to my supervisor Ing. Jan Pšikal, Ph.D. for the useful comments, remarks and engagement through the learning process of this master's thesis.

Čestné prohlášení:
Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl jsem všechnu použitou literaturu.

V Praze dne 4. 5. 2017 Viktor Kocur

Název práce:

Vliv počtu makročástic na přesnost a časovou náročnost částicových simulací ve fyzice laserového plazmatu

Autor: Viktor Kocur

Obor: Informatická fyzika

Druh práce: Diplomová práce

Vedoucí práce: Ing. Jan Pšikal, Ph.D.
Katedra fyzikální elektroniky, Fakulta jaderná a fyzikálně inženýrská, České vysoké učení technické v Praze

Abstrakt: Tato práce se zabývá numerickými efekty spojenými s makročásticemi v particle-in-cell (PIC) simulacích ve fyzice laserového plazmatu. Je prezentován přehled fyziky interakce ultraintenzivních laserových pulsů s ionizovanými terči. Algoritmus PIC metody je popsán a jeho numerické vlastnosti jsou teoreticky prozkoumány. Teoreticky je odvozená obecná závislost chyb v simulaci na počtu makročástic. Navrženy jsou 2D a 3D simulace interakce ultraintenzivního laseroveho pulsu s ionizovaným terčem tak, aby bylo možné empiricky zkoumat numerické výsledky spojené s počtem makročástic v simulaci a dalšími parametry. PIC kód EPOCH je použit k realizaci simulací na výpočetních zdrojích poskytnutých službou Metacentrum VO a počítačovým klastrem ARIS. Výsledky těchto simulací a numerické efekty, které vznikly díky makročásticím jsou podrobeny analýze. Dodatečné 2D a 3D simulace jsou realizovány za účelem zjištění dopadu počtu makročástic v simulaci a dalších simulačních parametrů na časovou náročnost PIC simulací.

Klíčová slova: kód EPOCH, laserové plazma, particle-in-cell, ultrakrátké laserové impulzy

Title:

The effect of macroparticle number on particle-in-cell simulation results and computational demands in laser plasma physics

Author: Viktor Kocur

Abstract: This thesis deals with the numerical effects related to the macroparticle number in particle-in-cell (PIC) simulations in laser plasma physics. Overview of physics of ultra-intense laser pulse interaction with ionized targets is presented. The PIC algorithm is described and its numerical properties are theoretically examined. A general dependence of the errors in the simulation on the macroparticle number is derived. Simulations of interaction of ultra-intense laser pulse with an ionized target is designed in 2D and 3D to empirically examine the numerical effects related to the macroparticle number and other parameters. PIC code EPOCH is used to execute simulations on computational resources provided by Metacentrum VO and cluster ARIS. Results of these simulations are analyzed and the numerical effects that occur in the simulations are examined and discussed. Additional 2D and 3D simulations are performed to determine the impact of macroparticle number and other simulation parameters on the computational demands of PIC simulations.

Key words: EPOCH code, laser plasma, particle-in-cell, ultrashort laser pulses

Contents

Introduction

This thesis deals with the particle-in-cell method and its application in laser plasma physics. Specifically, this thesis investigates the impact of the macroparticle number on the results and computational demands of simulations of interactions of ultra-intense laser pulses with ionized targets. This is a continuation of research interest in the particle-in-cell method and its application, which has been the central topic of previous academic projects [1, 2].

The particle-in-cell method is the most widely used simulation method employed in contemporary research of laser plasma physics. Simulations are crucial in this area of physics, as they allow the study of various relevant physical mechanism and effects with greater detail than in laboratory experiments. The cost of the simulations is usually smaller than realization of laboratory experiments and hence most of the planned experiments are firstly simulated to determine the feasibility of the experimental setup.

The particle-in-cell method has been in use since the 1950's. Increase in the computational capabilities since then enabled simulations of highly complex physical phenomena with great accuracy. The method has been theoretically examined in prominent textbooks [3, 4] in the 1970's. Multiple parts of the general algorithm have been improved and used in practice, but very little research has been directed at the numerical properties of the contemporary particle-in-cell codes [5, 6]. Lack of such research is the motivation for this thesis.

The bachelor's thesis [1] explored the numerical aspects of the contemporary particle-in-cell method with regards to the size of the cells used in the simulations. The results and findings were compared to the results presented in the aforementioned textbooks. One crucial result of the bachelor's thesis was that one of the often cited stability conditions, which was very restrictive and relevant for the older particle-in-cell codes is not relevant for contemporary codes and their use. This thesis aims to explore the numerical effects in similar fashion, but this time with regards to the number of particles used in the simulation.

In the first chapter of this thesis a general framework of theoretical description of processes relevant to plasma physics is established. This is followed by further theoretical exploration of topics related to the physics of ultra-intense laser pulse interaction with ionized targets, such as descriptions of various mechanisms of absorption of the pulse energy by the target. Practical issues regarding generation of the ultra-intense pulses are also discussed along with latest experimental results in this field. Selected applications of the interactions of ultra-intense laser pulses with ionized targets are also presented.

The core algorithm of the particle-in-cell method is described in the first part of the second chapter of this thesis. The individual steps of the algorithm are taken from the particle-in-cell code EPOCH [6]. Stability conditions of the code are theoretically discussed. Two distinct numerical mechanisms, which cause inaccuracies in the simulations are described in detail. One deals with the effects caused by the finite grid and the other deals with the effects caused by the representation of the distribution function via macroparticles.

The second part of the chapter deals with the practical aspects of running the simulations, as well as with the code EPOCH, which is used to run particle-in-cell simulations for the purposes of this thesis. Procedures necessary to run the simulations and process their outputs are briefly described. Properties of the computational resources on which the simulations were run, computer clusters at Metacentrum VO and cluster ARIS, are presented.

In the last chapter a simulation setup of a laser pulse interaction with an ionized target is designed in a matter that is suitable for the exploration of the numerical effects presented in the second chapter. Multiple simulations of the designed interaction are performed in 2D and 3D version of the EPOCH code along with several simulations used to determine the computational demands of the code EPOCH. Results of these simulations are presented and two different coefficients are used to determine whether the theoretical predictions hold in the real simulations. The results are then discussed with regards to the theoretical predictions of the numerical effects as well as the practical implications they have, especially with regards to an effective use of computational resources.

Chapter 1

Physics of ultra-intense laser pulse interaction with ionized targets

1.1 Basic concepts

In order to describe mechanisms of laser interaction with targets, introduction of a few basic concepts of plasma physics is necessary. Plasma is a quasi-neutral system of charged and neutral particles, which exhibits collective behavior [7]. Collective behavior means that the particles are not affected by the nearest particles, e.g. collisions in neutral gas or Coulomb collisions between two charged particles due to their self-generated microscopic fields, but by multiple particles through macroscopic fields. Quasi-neutrality is defined using the Debye length and plasma frequency. These and few other crucial concepts in plasma physics are presented in this section. We will refer to these concepts in other sections of this chapter as well as in the second chapter of this thesis.

1.1.1 Maxwell's equations

In order to describe the macroscopic fields in plasma we use Maxwell's equations (1.1 - 1.4) [8]. These form the basis of electrodynamics. They describe how electric ($\boldsymbol{E}$) and magnetic ($\boldsymbol{B}$) fields behave, their dependence on each other as well as on the current ($\boldsymbol{j}$) and the spatial charge density ($\boldsymbol{\rho}$).

$$\nabla \cdot \boldsymbol{E} = \frac{\rho}{\epsilon_0} \tag{1.1}$$

$$\nabla \cdot \boldsymbol{B} = 0 \tag{1.2}$$

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t} \tag{1.3}$$

$$\nabla \times \boldsymbol{B} = \mu_0 \boldsymbol{j} + \frac{1}{c^2}\frac{\partial \boldsymbol{E}}{\partial t} \tag{1.4}$$

1.1.2 Maxwell-Boltzmann distribution

Particles in gases have varying velocities. If a gas is in a thermal equilibrium the most probable way the particles are distributed in the velocity phase space is given by the Maxwell-Boltzmann distribution (1.5) [7], where m is the particle mass, k_B is the Boltzmann constant, $\boldsymbol{v}$ is the particle velocity and T is the temperature of the system. In following text we will refer to the Maxwell-Boltzmann distribution as Maxwellian distribution.

$$f(\boldsymbol{v}) = \left(\frac{m}{2\pi k_B T}\right)^{\frac{3}{2}} exp\left(-m\frac{v_x^2 + v_y^2 + v_z^2}{2k_B T}\right) \tag{1.5}$$

It is important to note that even if a system is not Maxwellian, it may be possible to find directions in which it is. Similarly, it may be possible to separate a non-Maxwellian system into subsets with each subset having a distinct temperature. Therefore it is possible for a system to be described by multiple temperatures with regard to different directions and subsets of particles.

In some situations it is often more useful to use the thermal velocity as a measure of the systems temperature. Thermal velocity is defined as the root mean square of the velocity in one dimension. For a given temperature of particles with mass m_s, the thermal velocity follows the the following formula:

$$v_{ts} = \sqrt{\frac{k_B T}{m_s}}. \tag{1.6}$$

1.1.3 Debye length

Debye length is a measure of how far the electrostatic effect of a plasma persists. If we assume that a test particle with the charge of q_t is inserted into the plasma, the resulting electric potential would depend on distance (r) as follows:

$$\phi_t(r) = \frac{q_t}{4\pi\epsilon_0 r} e^{-r/\lambda_{De}}. \tag{1.7}$$

Debye length is located in the denominator inside the exponent. For higher distances we can assume the potential to be negligible, since it vanishes exponentially. This effect is called Debye shielding and we can consider the plasma to be quasi-neutral from such distances. This defines the spatial requirement of quasi-neutrality [9].

The magnitude of the Debye length can be expressed as (1.8). It depends on the electron temperature T_e and the electron number density n_e.

$$\lambda_{De} = \sqrt{\frac{\epsilon_0 k_B T_e}{n_e e^2}} \tag{1.8}$$

1.1.4 Electron plasma frequency

If we study a system in which electrons are pulled away from the state of equilibrium and assuming that the ions have infinite mass, which is reasonable since they are usually heavier than the electrons by a few orders of magnitude, the electrons will oscillate around the ions. The frequency of these oscillations is known as the electron plasma frequency. Magnitude of this frequency is given by the formula (1.9) [10], which depends on the electron charge (e) and mass (m_e) as well as the electron number density (n_e).

$$\omega_{pe} = \sqrt{\frac{e^2 n_e}{\epsilon_0 m_e}} \tag{1.9}$$

The electron plasma frequency allows us to define the temporal requirement of quasi-neutrality (1.10). A system is quasi-neutral if the characteristic time (τ) of any relevant physical process within the system fulfills this condition [10].

$$\tau \gg \omega_{pe}^{-1} \tag{1.10}$$

Along with electron plasma frequency a similar quantity is defined for the ions. The ion plasma frequency (1.11) is used to describe processes related to the movement of ions. In its formula the ion mass (m_i) and charge (Ze) are present along with the ion number density (n_i).

$$\omega_{pi} = \sqrt{\frac{Z^2 e^2 n_i}{\epsilon_0 m_i}} \tag{1.11}$$

1.1.5 Landau damping

The oscillations described in the previous subsection manifest themselves as electron plasma waves. These waves have the following dispersion relation [10]:

$$\omega^2 = \omega_{pe}^2 + 3k^2 v_{Te}^2. \tag{1.12}$$

Further analysis of electron plasma waves leads to the derivation of Landau damping. This effect describes how the plasma waves are damped without any collisional mechanisms involved. Using the kinetic description of plasma with perturbation theory and the properties of complex integral, it is possible to derive the dispersion relation (1.13), where f_0 is the unperturbed electron distribution function.

$$\omega^2 = \omega_{pe}^2 + i\pi\omega^2 \frac{\omega_{pe}^2}{k^2} \frac{\partial f_0}{\partial v}\bigg|_{v=\frac{\omega}{k}} \tag{1.13}$$

The imaginary term in the dispersion relation causes damping of the wave. Particles with velocities close to ($v_\phi = \omega/k$) will exchange a lot of energy with the wave. If their velocity is smaller than v_ϕ they will get accelerated by the wave. In the opposite case, the particles will get decelerated by the wave. If we assume Maxwellian distribution, then there will be more slower particles, thus the wave will lose more energy than it will gain and is therefore damped [11].

1.2 Description of plasma

In order to handle plasma theoretically, it is necessary to introduce a mathematical model which will allow us to describe and derive various processes occuring in real plasmas.

1.2.1 Kinetic description

The most general description of plasma utilizes descriptions of the individual particles via Klimontovich equation. This approach is not directly appropriate in order to achieve relevant results [10]. Instead, we construct a distribution function defined on a phase space, denoted as $f_s(\boldsymbol{x}, \boldsymbol{p}, t)$, wherein $\boldsymbol{x}$ and $\boldsymbol{p}$ are respectively the vectors of position and velocity in the phase space. The term $f_s(\boldsymbol{x}, \boldsymbol{p}, t) d\boldsymbol{x} d\boldsymbol{p}$ then represents the amount of particles in the phase volume $(\boldsymbol{x}, \boldsymbol{x} + d\boldsymbol{x}) \times (\boldsymbol{p}, \boldsymbol{p} + d\boldsymbol{p})$ at time t.

Under the assumption that f_s is differentiable, it is possible to describe the evolution of the distribution function in time using the Vlasov equation (1.14).

$$\left[\frac{\partial}{\partial t} + \frac{\boldsymbol{p}}{m_s\gamma} \cdot \frac{\partial}{\partial \boldsymbol{x}} + q_s\left(\boldsymbol{E} + \frac{\boldsymbol{p}}{m_s\gamma} \times \boldsymbol{B}\right) \cdot \frac{\partial}{\partial \boldsymbol{p}}\right] f_s = 0 \tag{1.14}$$

In this equation, f_s is the aforementioned distribution function for the particles of type s, q_s and m_s is their charge and mass respectively. The relativistic factor is denoted as γ. $\boldsymbol{E}$ and $\boldsymbol{B}$ are the centered macroscopic field vectors. Different particle types can interact through these vectors. This equation represents only the interactions by means of the collective plasma effects. In a case that the collisions would not be negligible, their effects would have to be included on the right side of the equation via a collision term [10].

1.2.2 Fluid description

Under the assumption that the system is in a local thermodynamic equilibrium, i.e. due to collisions the distribution of particles is Maxwellian, it is possible to derive the fluid description of the system from the kinetic one. This is done by calculating the moments of the non-relativistic Vlasov equation (eq. 1.14 with $\gamma = 1$ and $\boldsymbol{p} = m_s\boldsymbol{v}$).

From the first moment of the equation we can derive the law of conservation of mass (1.15).

$$\frac{\partial n_s}{\partial t} + \nabla(n_s \boldsymbol{v}_s) = 0 \tag{1.15}$$

The second moment of the equation leads us to the law of conservation of momentum (1.16). This equation is quite often called the Navier-Stokes equation.

$$\frac{\partial \boldsymbol{v}_s}{\partial t} + (\boldsymbol{v}_s \nabla)\boldsymbol{v}_s = -\frac{1}{\rho_s}\nabla \cdot \overset{\leftrightarrow}{P^s} + \frac{q_s}{m_s}(\boldsymbol{E} + \boldsymbol{v}_s \times \boldsymbol{B}) \tag{1.16}$$

The first term on the right side of the equation is the divergence of the pressure tensor $\overset{\leftrightarrow}{P^s}$ divided by mass density ρ_s. The second term represents the Lorentz force.

The third moment of the Vlasov equation yields the law of conservation of energy (1.17).

$$\frac{3}{2} n_s k_b \frac{\partial T_s}{\partial t} + \frac{3}{2} n_s k_b (\boldsymbol{v}_s \nabla) T_s + \nabla \boldsymbol{q}_s + P^s_{ik} \frac{\partial \boldsymbol{v}_{si}}{\partial r_k} = 0 \tag{1.17}$$

This equation contains the temperature T_s and the conductive hear flux $\boldsymbol{q}_s$. The last term of the equation on the left side represents the work done due to pressure.

The equations (1.15 - 1.17) are not complete. In order to complete the system it is necessary to add a state equation. This would allow us to describe the plasma as a system of two liquids (electrons and ions) and use various hydrodynamic techniques. This approach is useful especially for interaction of plasma with longer laser pulses (ns) and lower intensity ($\leq 10^{15}$W/cm^2), since under such circumstances, the processes leading to the validity of local thermodynamic equilibrium dominate [10].

1.2.3 Numerical simulations

Many processes occurring in real laser plasma interaction are impossible to describe analytically. In such cases it is often possible to study these effects using computer simulations.

Using the particle description of plasma, it is possible to simulate microscopic processes at the level of individual particles. It is possible to utilize molecular dynamics or Monte-Carlo methods at this level [12]. This approach is mostly used in simulations of smaller volumes within plasmas in experimental devices such as tokamaks.

In the field of laser plasma physics, particle simulations are often successfully used. This approach differs from the microscopic one mentioned above in the fact that the simulated particles represent multiple real particles. These simulated particles are therefore referred to as macroparticles. They move on top of an Eulerian grid on which the macroscopic fields are calculated. This method of plasma simulation is called particle-in-cell (PIC). This is the most often used method in the field of laser plasma physics. This thesis deals with the numerical aspects of these simulations. The second chapter of this thesis deals with an in-depth theoretical description of the method.

Another method utilizing macroparticles is the treecode method [13]. Interaction of the macroparticles is calculated as a sum of the effects of all the other particles. This algorithm has a time complexity of $O(n \log n)$ thanks to grouping of macroparticles in clusters. In general, this method

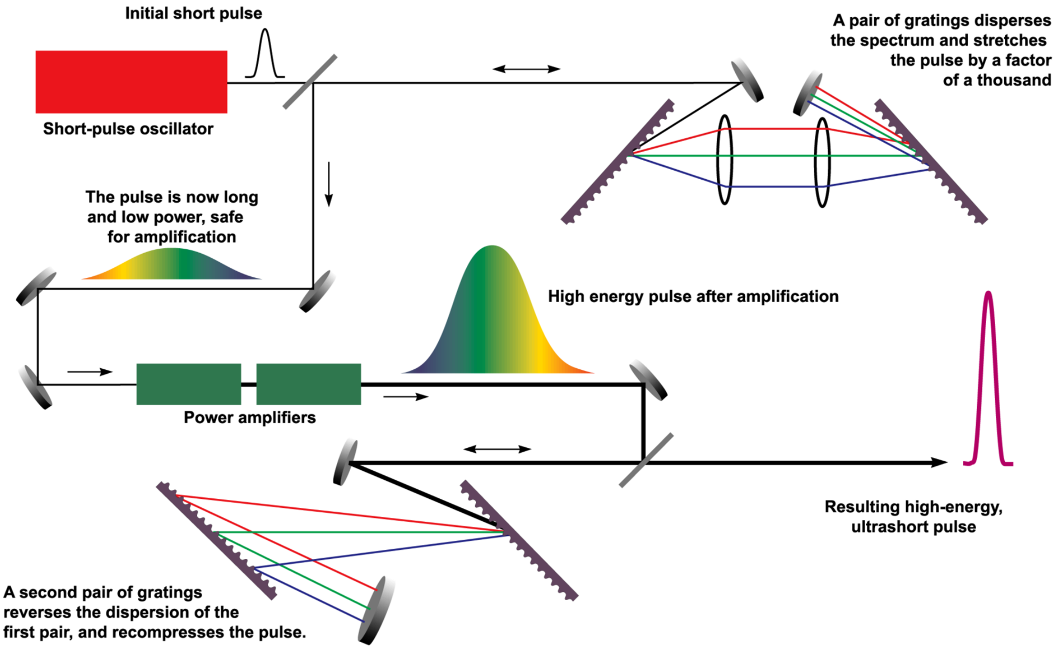
Initial short pulse
Short-pulse oscillator
A pair of gratings disperses the spectrum and stretches the pulse by a factor of a thousand
The pulse is now long and low power, safe for amplification
High energy pulse after amplification
Power amplifiers
Resulting high-energy, ultrashort pulse
A second pair of gratings reverses the dispersion of the first pair, and recompresses the pulse.

Figure 1.1: CPA pulse generation scheme. [15]

is less efficient than PIC, however in some cases when studying complex behavior, it may achieve better efficiency than a PIC algorithm which would require a very fine mesh and thus would be computationally expensive [14].

In case the simulated plasma is in a local thermodynamic equlibrium, it is possible to use the fluid description. As mentioned above, this allows the use of various Lagrangian and Eulerian approaches to simulations including their combination.

1.3 Ultra-intense laser pulse generation

The basic requirement for advances in the field of ultra-intense laser pulse interaction with ionized targets is the ability to generate such pulses. This sections deals with methods used to generate them.

1.3.1 CPA

The ultra-intense pulses are so intense that they damage optical components. Chirped-pulse amplification (CPA) [16] is a method that makes it possible to increase the energy of a pulse without occurrence of high intensities in the process. This is achieved via a reversible chirping of a pulse and a following recompression to a short pulse.

CPA scheme works as follows (see fig. 1.1). A low-energy ($\sim 10^{-9}$ J) ultrashort ($10^{-12} - 10^{-14}$ s) pulse is generated by a laser utilizing the mode-locking technique. This pulse is chirped on a diffraction grid or in an optical fiber. The pulse is now around thousand times longer than the initial pulse. Next, the pulse is amplified in multiple optical amplifiers. The pulse is amplified by around 6 to 9 orders of magnitude. Finally, the pulse is recompressed to the original short length [17].

1.3.2 OPCPA

Properties of some crystals allow for optical parametric amplification (OPA). This amplification works by having the signal pulse propagate through the crystal together with a pump pulse of lower wavelength. The pump photons transform into the photons of the same wavelength as the photons of the signal pulse as well as the same amount of so-called idler photons. The energy of an idler photon is equal to the energy difference between a signal and a pump photon. Thanks to this, the crystal material is not heated. This way, three pulses are leaving the crystal: the strengthened signal pulse, the idler pulse and the residual pump pulse [18].

This method can be used within CPA when amplifying the chirped pulse. Combination of these methods is referred to as optical parametric chirped pulse amplification (OPCPA) [19].

1.3.3 Latest results

At present, multiple laser systems with power above 1 PW are operational. The most powerful ones reach around 1.5 PW [20]. The highest intensity reached in the focal point of a laser has been in the order of magnitude of 10^{22}W/cm^2 [21]. At the intensities around 10^{23}W/cm^2 the plasma in the target would reach ultra-relativistic regime. Under the influence of a pulse of that such intensity, the protons would be accelerated to relativistic speeds, which would open the doors to a whole new field of experiments.

1.4 Interaction of ultra-intense laser pulse with ionized targets

This section deals with the basic physical description of interactions of ultra-intense laser pulses with ionized targets. These basic physical concepts allow us to understand and describe more complicated physical processes.

1.4.1 Laser pulse description

Laser pulse is an electromagnetic wave propagating towards the target. In order to describe it, we will use the scalar and vector potentials. These two potential describe the electric and magnetic fields via the following equations [8]:

$$\boldsymbol{E} = -\nabla\phi - \frac{\partial \boldsymbol{A}}{t} \tag{1.18}$$

$$\boldsymbol{B} = \nabla \times \boldsymbol{A} \tag{1.19}$$

Assuming the Coulomb gauge ($\nabla \cdot \boldsymbol{A} = 0$) and an additional assumption that $\nabla\phi = 0$, both fields are fully described by the vector potential.

An important property of a pulse is its polarization. The vector potential of a circularly polarized pulse, propagating along the z-axis, has the following form:

$$\boldsymbol{A} = A_0 \left[cos\left(\omega\tau\right), sin\left(\omega\tau\right), 0\right] \tag{1.20}$$

This form contains the retarded time $\tau = t - z/c$. In this kind of wave the vectors of the electric and magnetic field rotate in the plane perpendicular to the z-axis.

The vector potential of a linearly polarized pulse, again propagating along the z-axis, has the following form:

$$\boldsymbol{A} = A_0 \left[cos\left(\omega\tau\right), 0, 0\right] \tag{1.21}$$

In this case, the electric field oscillates along the x-axis, while the magnetic field field oscillates along the y-axis.

Depending on the intensity of the pulse relativistic effects may occur, hence it is often useful to define the so-called dimensionless amplitude (a_0) as follows:

$$a_0 = \frac{eA_0}{m_e c} = \frac{eE_0}{m_e c\omega} \tag{1.22}$$

If the value of a_0 is equal to one, the kinetic energy of an electron oscillating due to the fields of the pulse is equal to the electron rest energy. In the case of $a_0 > 1$, the electron is approaching the speed of light. Therefore in such case it is necessary to assume relativistic effects.

The intensity of a pulse (1.23) is given by the energy flux averaged over one wave period.

$$I = \left\langle \left| \boldsymbol{E} \times \frac{\boldsymbol{B}}{\mu_0} \right| \right\rangle = \frac{\epsilon_0 c}{2} \left|E_0\right|^2 \tag{1.23}$$

This definition allows us to determine the relationship between the intensity of a pulse and its dimensionless amplitude (1.24). For pulses with intensity exceeding 10^{18} W/cm^2 at 1 μm laser wavelength relativistic effects have to be taken into account.

$$I\lambda^2 = a_0 \cdot 1.37 \cdot 10^{18} \left[\mathrm{W}\mu\mathrm{m}^2\mathrm{cm}^{-2}\right] \tag{1.24}$$

So far we have assumed that the pulses are plane waves. This is not the case in reality, where pulses are contained to a small area in the plane perpendicular to the direction of their propagation. Usually, such pulses have a Gaussian profile and they are focused on the front side of the target.

1.4.2 Ponderomotive force

Charged particles oscillate under the influence of a variable electric field. If we time-average the kinetic energy of a particle in such a field, we obtain a formula for the ponderomotive potential

(1.25), where ω is the frequency of the pulse and $\boldsymbol{E}$ is the amplitude of the electric field.

$$W_p(\boldsymbol{r}) = \frac{e^2}{4m\omega^2} |\boldsymbol{E}|^2 \tag{1.25}$$

From this potential one can easily derive a formula for the so-called ponderomotive force:

$$\boldsymbol{F}_p = -\nabla W_p = -\frac{e^2}{4m\omega^2} \nabla |\boldsymbol{E}|^2 \tag{1.26}$$

In the relativistic case of $a_0 > 1$, the formula transforms to (1.27) [9], where $\langle\gamma\rangle$ is the time-averaged relativistic factor of an electron (1.28)

$$\boldsymbol{F}_p = -m_e c^2 \nabla \langle\gamma\rangle \tag{1.27}$$

$$\langle\gamma\rangle = \sqrt{1 + \frac{\langle\boldsymbol{p}\rangle^2}{(m_e c)^2} + \frac{e^2 \left\langle \boldsymbol{A}^2 \right\rangle}{(m_e c)^2}} \tag{1.28}$$

As can be seen from these formulas, the ponderomotive force is nonlinear. It increases the particle density in areas where the electric field is weaker. This force affects mostly the electrons. However, these electrons later pull the heavier ions towards them due to the electrostatic forces. This results in the pulse pushing the charged particles away.

1.5 Pulse propagation in plasma

The dispersion relation (1.29) is important for propagation of a laser pulse in plasma. This relation contains the electron plasma frequency mentioned in the first section of this chapter.

$$\omega^2 = \omega_{pe}^2 + c^2 k^2 \tag{1.29}$$

From this relation we can deduce that the electromagnetic wave of frequencies lower than ω_{pe} will not propagate through the plasma. Since laser pulses are monochromatic, we can define the critical density (1.30), which is dependent on the frequency of the pulse.

$$n_c = \frac{\epsilon_0 m_e \omega^2}{e^2} \tag{1.30}$$

In case the electron number density of a plasma is higher than the critical density, the laser pulse won't propagate through it and will be deflected instead. We refer to such plasma as an overdense plasma. In the other case, e.g. when the electron number density of a plasma is smaller than the critical density, we refer to the plasma as underdense.

If we assume relativistic effects ($a_0 > 1$), the dispersion relation (1.29) gets modified into the form (1.31) [22]. This modified dispersion relation depends on the polarization and the dimensionless amplitude (1.32).

$$\omega^2 = \frac{\omega_{pe}^2}{\gamma_L} + c^2k^2 \tag{1.31}$$

$$\gamma_L = \begin{cases} \sqrt{1 + a_0^2} & \text{for circular polarization} \\ \sqrt{1 + \frac{a_0^2}{2}} & \text{for linear polarization} \end{cases} \tag{1.32}$$

In case the condition (1.33) is satisfied, the wave will propagate through an overdense plasma. This regime is referred to as relativistic transparency.

$$n_c < n_e < \gamma_L n_c \tag{1.33}$$

1.6 Absorption mechanisms

Absorption mechanisms play an important role in the interaction of laser pulses with plasma. Multiple distinct absorption mechanisms are described in this section.

1.6.1 Collisional absorption

Collisional absorption is often referred to as the inverse bremsstahlung. Bremsstahlung is a mechanism in which radiation is emitted when two charged particles collide, usually an electron and an ion. After the collision, some of the kinetic energy of particles is transformed into the energy of an emitted photon.

Collisional heating is the inverse mechanism. During a collision a photon is absorbed and its energy is transformed into the energy of colliding particles. Following Coulomb collisions of those particles with others distribute the energy to the rest of the plasma.

This mechanism works well when enough collisions occur in relation to the length of the pulse in time. These conditions are satisfied for longer pulses (> 1 ns) and lower intensities ($< 10^{16}$W/cm^2). Under such circumstances the collional absorption can yield up to 80% absorption [23].

Other mechanisms are also relevant under such circumstances [24–26], however, they are not important for the study of more intense laser pulses and require a steep interface between the plasma and vacuum [23].

1.6.2 Resonance absorption

This absorption mechanism occurs when linearly polarized pulses are obliquely incident to the plasma interface. Regarding this mechanism, it is important to distinguish two possible ways the pulse is polarized. If the electron number density of the plasma is a linear function of the z coordinate ($n_e(0) = 0$) and the pulse propagates in the y-z plane, then if the electric field of the pulse oscillates along the x-axis and the magnetic field oscillates in the y-z plane, the pulse is s-polarized.

If the situation is reversed, e.g. the electric field oscillates in the y-z plane and the magnetic field oscillates along the x-axis, the pulse is p-polarized.

The electric field of a p-polarized pulse has a singularity in $n_e(z) = n_c$. This singularity results in resonance and hence produces plasma waves. These waves are then damped by various collisional and non-collisional mechanisms. Thanks to this phenomenon, the resonance absorption is efficient even for pulses of higher intensities. With the correct incidence angle and a p-polarized pulse, the resonance absorption mechanism can yield up to 50% absorption [27].

1.6.3 Brunel heating

Brunel heating, also referred to as vacuum heating, occurs similarly when a p-polarized pulse is obliquely incident to the plasma interface. Additionally, if the oscillation radius of an electron in the electric field of the pulse is higher than the depth of the inclining part of the plasma profile (1.34), then the electric field pulls the electrons into the vacuum in front of the target.

$$\frac{a_0 c}{\omega} \geq L \tag{1.34}$$

Next, the electrons are accelerated back into the plasma. Under such circumstances the electron will travel into the overdense plasma, where the outside electric field is shielded. Therefore the electron won't be decelerated by the outside field, which would happen if the field wasn't shielded [28]. Kinetic energy of the accelerated electrons will then dissipate into the plasma via other processes and the plasma is thus heated.

Analytical models predict that this kind of absorption can yield only rarely more than 10% absorption [22]. In relativistic cases the Brunel heating is dominated by the $\boldsymbol{j} \times \boldsymbol{B}$ heating, since the Brunel heating mechanism requires higher incidence angles and the absorption occurs only during a small part of the laser wave period.

1.6.4 $\boldsymbol{j} \times \boldsymbol{B}$ heating

In perpendicularly incident pulses the electric field perpendicular to the plasma interface is null, therefore the Brunel heating won't occur. In case that the pulse is intense enough, the magnetic term of the Lorentz force becomes prominent. Similarly as with the previous heating mechanism, the electrons are pulled into the vacuum in front of the target, where they are accelerated. Next, the electrons are pushed inside the target, where they won't be decelerated thanks to the shielding effect of the plasma. The electrons are pulled from the target and subsequently accelerated twice per pulse period. Analytical model predicts constant absorption at 15 % [22].

1.6.5 Hot electrons

When considering only the collisional heating mechanisms, all the electrons are heated. Therefore it is possible to assign a single temperature to them.

If the electrons are heated via non-collisional mechanism, a small part of them is heated to higher energies than the rest. This creates a population of so-called hot electrons. Since they usually have a Maxwellian distribution, it is possible to assign them a temperature, which is higher than the temperature of the remainder of the electrons [22].

1.7 Applications

Advances in the physics of laser interaction with plasma allow for various implementations in other fields of physics as well as direct practical applications.

1.7.1 K-α radiation

Hot electrons, which have been accelerated in the front of the target travel inside it. They can even enter the part of the target where the matter has not been ionized yet. Hot electrons collide with other particles. During these collisions, ionization of atoms may occur. Ions with a missing electron seek to transition to a lower energy state as soon as possible. This transition can happen either as the Auger effect, or by emission of a characteristic photon.

Auger effect occurs when an inner-shell vacancy is filled by an electron from the outer shell. Residual energy from this electron is transferred to another electron in the outer shell. The electron can gain enough energy to leave the atom. This effect dominates in atoms with lower atomic number.

Excess energy gained from filling of the inner-shell vacancy can also transform into the energy of an emitted photon. Properties of this radiation are determined by the levels between which the electron transitions. This kind of radiation is called K-α. It is isotropic and the plasma is often transparent to this radiation.

K-α radiation has a few interesting properties. It is almost monochromatic in a narrow part of the x-ray spectrum. Pulses of this radiation are very short, generally in femto and picoseconds. The efficiency of the energy conversion from the laser pulse can be around 10^{-4} and is therefore efficient enough for further applications [23].

If the pulses are generated with high enough repeating frequency, it is possible to use this mechanism as a source of radiation for various applications in imaging diagnostics [29]. It is also possible to utilize this radiation to study the behavior of hot electrons [30, 31].

1.7.2 Ion acceleration

Energy absorbed from the pulse directly accelerates predominantly the electrons. These accelerated electrons can exert electromagnetic force on the ions and thus accelerating them as well.

This kind of acceleration can occur on the front side of the target. Radiation pressure (due to the ponderomotive force) of the incoming pulse pushes the electrons inside the target. This creates a positively charged layer in front of the target. An electrostatic field appears and pushes the ions

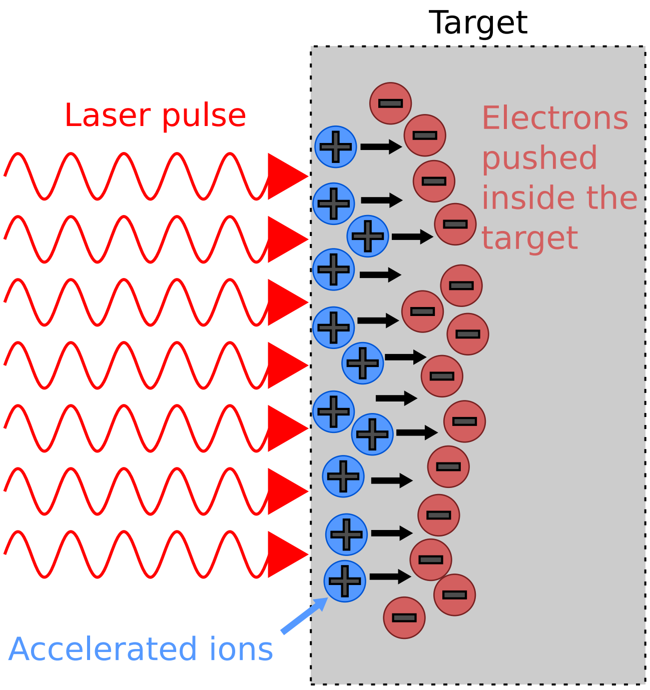

Figure 1.2: RPA mechanism.

inside the target. This mechanism is referred to as radiation pressure acceleration (RPA) and is demonstrated in fig. 1.2. Maximum energy for a proton accelerated by this mechanism is given by (1.35) [32].

$$E_{imax} \approx \frac{Zm_ec^2a_0^2}{m_i\gamma_L} \tag{1.35}$$

Ions can also be accelerated in the back of the target. Hot electrons can travel through the target and travel into the area behind it. A resulting electric field pulls them back and an electron layer few Debye lengths thick forms behind the target. The electric field accelerates the ions behind the target. This mechanism is referred to as target normal sheath acceleration (TNSA) and is schematically depicted in fig. 1.3.

The maximum energy of the ions accelerated by TNSA mechanism is given by (1.36) and (1.37), where t_{acc} is a time during which the acceleration occurs, ω_{pi} is the ion plasma frequency and e_n is the Euler number.

$$E_{imax} \approx 2ZT_eln^2\left(\tau + \sqrt{\tau^2+1}\right) \tag{1.36}$$

$$\tau = \frac{\omega_{pi}t_{acc}}{\sqrt{2e_n}} \tag{1.37}$$

TNSA is the most commonly observed ion acceleration mechanism in experiments [9]. In 2016 TNSA was sucessfully used to accelerate protons to the energy of 85 MeV with a laser pulse of in-

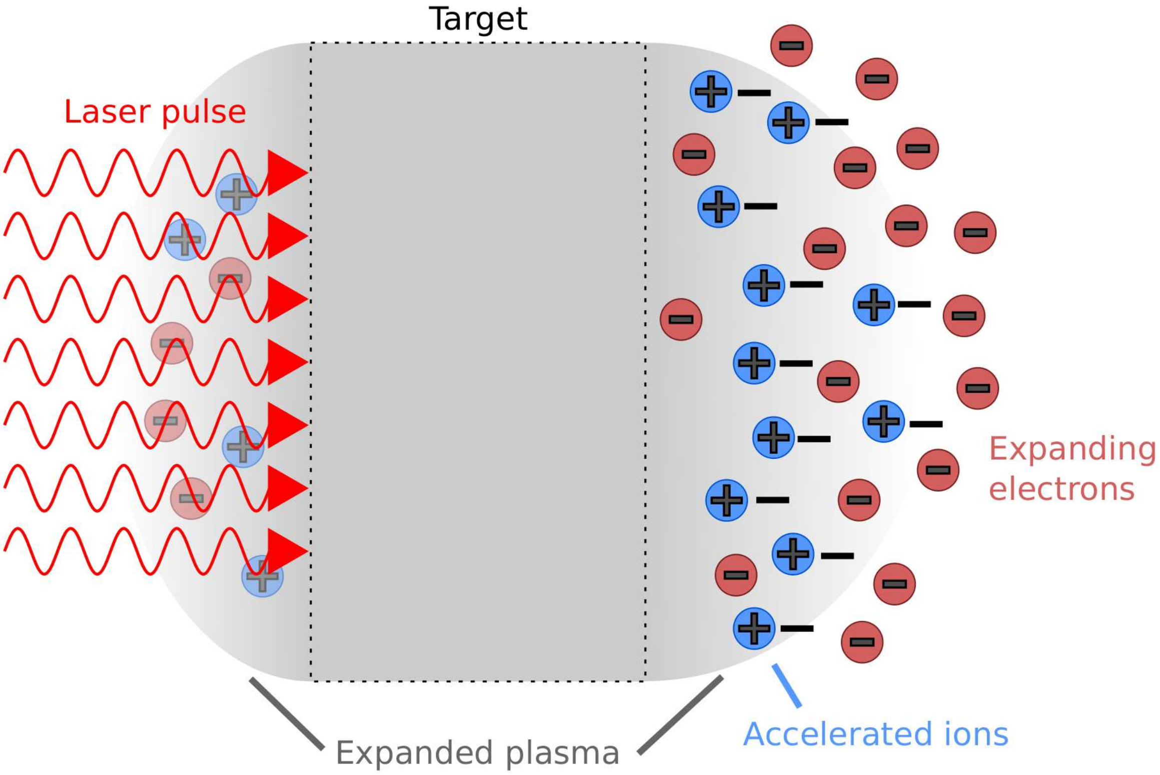

Figure 1.3: TNSA mechanism.

tensity under 10^{20}W/cm^2 [33]. PIC simulations predict that further optimization of this experiment may yield protons with energies of up to 200 MeV.

1.7.3 Electron acceleration

When an intense pulse propagates through an underdense plasma the ponderomotive force pushes the electrons out. If the pulse length is similar to the Debye length, then strong plasma waves, referred to as wakefields, occur. Their phase velocity is similar to the group velocity of the pulse [34]. The electric field of such a wave is longitudinal, e.g. it points in the direction in which the pulse propagates. This field can accelerate electrons to very high speeds. Such acceleration mechanism is called laser wakefield acceleration (LWFA).

Pulse duration plays an important role in this mechanism. Most efficient conversion of energy of the pulse to the kinetic energy of electrons is achieved when the pulse is shorter than the plasma period ($T_{pe} = 2\pi/\omega_{pe}$) [35]. If the pulse is longer, the wakefields self-modulate, this regime is called self-modulating LWFA (SM-LWFA) and allows for efficient excitation of wakefields [36].

In relativistic regime ($a_0 > 1$) with pulses strong enough, wavebreaking of the wakefields can occur. This means that the electrons are trapped in the potential of the first oscillation. This entrapment leads to very efficient acceleration of large amounts of electrons. Using multiple pulses in this regime can lead to better results [35].

Utilizing multiple weaker pulses is also viable. In this case the generated wakefields sum up coherently if the pulses have the repeating frequency close to the electron plasma frequency. Electrons are then accelerated in the first oscillation of the wakefield after the last pulse has passed. This regime is called multi pulse LWFA (MP-LWFA).

In 2013 a pettawatt laser used in a two-level LWFA scheme was sucessfuly used to accelerate electrons to 3 GeV [37].

Chapter 2

Particle-in-cell simulation

This chapter deals with the particle-in-cell method. The general algorithm is described in the first three sections of this chapter. Various PIC codes utilize different algorithms and schemes, but the general principle is the same. The fourth section deals with the numerical aspects of the simulations. Any particular algorithms or schemes presented in the first four sections of this chapter are taken from the EPOCH code [6], which is an implementation of Plasma Simulation Code by Hartmut Ruhl [5]. The last three sections deal with the technical aspects of running the PIC simulations using the EPOCH code and performing analysis of their results.

2.1 Basic principle

Solution of the Vlasov equation (1.14) would be a powerful tool to understanding the physical processes in plasmas. Sadly, the equation is 7-dimensional and using standard numerical methods to find the solution would be immensely computationally expensive.

The PIC method replaces the distribution function f_s with macroparticles [38]. The relationship between the distribution function and the macroparticles can be described using the following relation:

$$f_s(\boldsymbol{x}, \boldsymbol{p}, t) = \sum_{k=1}^{N} N_k S_x(\boldsymbol{x} - \boldsymbol{x}_k) S_p(\boldsymbol{p} - \boldsymbol{p}_k). \tag{2.1}$$

Each macroparticle represents N_k real particles and has its own position $\boldsymbol{x}_k$ and momentum $\boldsymbol{p}_k$. The funcions S_x and S_p represent how a macroparticle "looks" in the position and momentum phase space respectively and are usually referred to as shape functions. Choice of the shape functions has an important impact on the numerical properties of the algorithm. The next section is dedicated to them.

Solution of the Vlasov equation requires the calculation of the macroscopic electric and magnetic fields. In PIC, this is achieved by calculating them on an Eulerian grid. The particles move on top of this grid. The macroparticles and fields interact via interpolation to and from the grid.

The benefits of using PIC method lie in the fact that the computational resources are focused in the areas of the phase space with the most particles. This approach also allows for very efficient parallelization and is therefore very well suited for modern architectures of computational resources.

2.2 Shape functions

Shape functions have to be non-negative functions with compact support satisfying the following condition:

$$\int_{supp(S)} S(\boldsymbol{x})d\boldsymbol{x} = 1. \tag{2.2}$$

In practice, the only momentum shape function used is the Dirac delta function. Using any other function would result in change of the spatial shape function in time, which is undesirable. Due to the fact that the momentum shape function is always the same, in the following text we will use the term shape function to indicate only the spatial shape function.

Considering possible spatial shape functions we can propose further requirements. Symmetry of the shape function is a desirable property. The shape function should also allow for computationally inexpensive interpolation. This can be achieved with functions that have small support compared to the cell size of the Eulerian grid and are piecewise low order polynomials.

These requirements are all satisfied by a class of functions called b-splines. The definition of b-splines is recursive, so it is necessary to define the zeroth order b-spline (2.3).

$$S_0(x) = \begin{cases} 1, & if|x| \leq \frac{\Delta x}{2} \\ 0, & otherwise \end{cases} \tag{2.3}$$

Splines of higher order are defined as convolutions of S_0 with itself. We can recursively define b-spline of the n-th order as follows [39]:

$$S_n(x) = S_{n-1} * S_0. \tag{2.4}$$

For further analysis of the numerical properties of the PIC algorithm it is often useful to apply the Fourier transformation. The transformed shape function (denoted with a hat) has the following form:

$$\hat{S}_n(k) = \left(\frac{sin\left(\frac{k\Delta x}{2}\right)}{\frac{k\Delta x}{2}} \right)^{n+1}. \tag{2.5}$$

EPOCH code can be compiled with one of three available b-splines: S_0,S_1,S_3. These are referred to as the tophat, triangular and smooth shape functions respectively due to their shapes (see fig. 2.1). In the following text and especially in figures we may refer to them as TH, ST and S3 respectively.

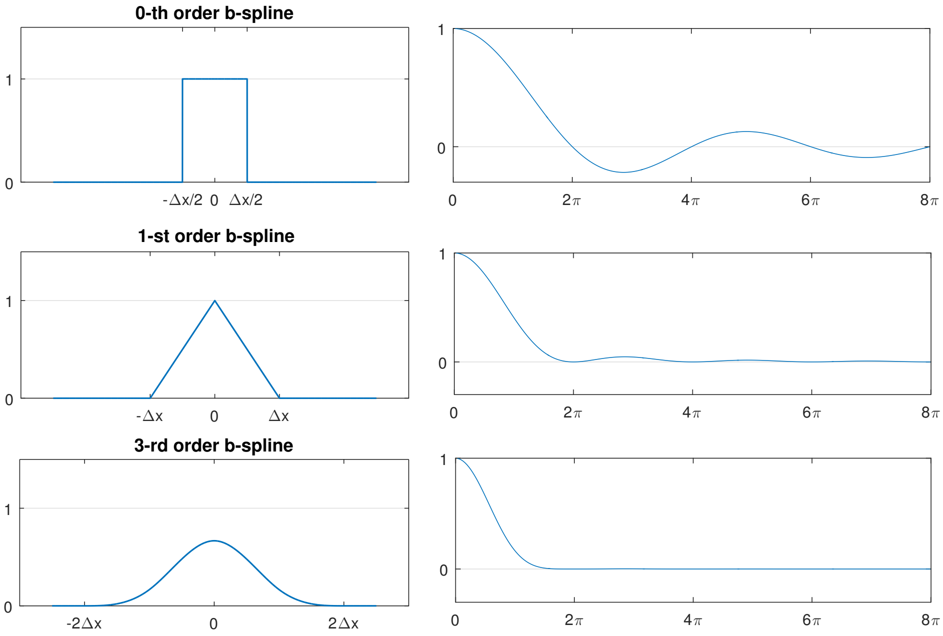

Figure 2.1: Left: B-spline shape functions. Right: Fourier images of b-splines.

2.3 Computational cycle

Computational cycle of the PIC algorithm is relatively simple. The simulation starts by initialization followed by four steps, which are repeated until the end of the simulation.

2.3.1 Initialization

At the start of the simulation the Eulerian grid is constructed. Macroparticles are distributed in space according to the initial conditions. Momenta of the macroparticles are most often initialized with a Maxwellian distribution, but it is possible to set them up differently. After that the electric and magnetic fields are initialized. They have to satisfy the equations (1.1) and (1.2) in discrete form as well as any boundary conditions set up at the edge of the Eulerian grid.

2.3.2 Step 1: Interpolation

In order to apply Lorentz force on the macroparticles it is necessary to perform interpolation of the fields from the grid onto the macroparticles. Shape functions allow us to analytically deter-

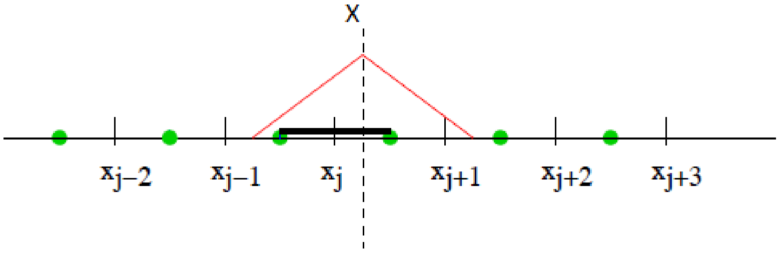

Figure 2.2: A triangular macroparticle in position X. Fields are known at points x_k.

mine how the magnitudes of fields at gridpoints get combined to calculate the forces acting on a macroparticle [40].

We will show how a field is interpolated onto a particle in 1D simulation with the triangular shape function. It is important to note that the electric and magnetic fields are calculated in different positions on the grid, this will be discussed further in subsection 2.3.5. Fields are calculated at positions x_k and their magnitude is F_kj. Assuming a macroparticle positioned at point X and the closest gridpoint to X is x_j (see fig. 2.2), we can calculate the total field acting on a macroparticle F_p using (2.6).

$$\begin{aligned} F_p = \frac{1}{\Delta x}\Bigg[&\int_{X-\Delta x}^{x_j-\frac{\Delta x}{2}} F_{j-1}\left(1-\frac{X-x}{\Delta x}\right)dx + \int_{x_j-\frac{\Delta x}{2}}^{X} F_j\left(1-\frac{X-x}{\Delta x}\right)dx \\ &+ \int_{X}^{x_j+\frac{\Delta x}{2}} F_j\left(1-\frac{x-X}{\Delta x}\right)dx + \int_{x_j+\frac{\Delta x}{2}}^{X+\Delta x} F_{j+1}\left(1-\frac{X-x}{\Delta x}\right)dx\Bigg] \end{aligned} \tag{2.6}$$

These integrals can be solved analytically to obtain:

$$F_p = \frac{1}{\Delta x}\left[\frac{1}{2}F_{j-1}\left(\frac{1}{2}+\frac{x_j-X}{\Delta x}\right)^2 + F_j\left(\frac{3}{4}-\frac{\left(x_j-X\right)^2}{\Delta x^2}\right) + \frac{1}{2}F_{j+1}\left(\frac{1}{2}-\frac{x_j-X}{\Delta x}\right)^2\right]. \tag{2.7}$$

Interpolation of macroparticles is dependent on the shape function chosen and dimension in which the simulation is performed. Computational demands increase with order of the shape function as well as with the number of spatial dimensions. For example, a macroparticle with a triangular shape function in 2D overlaps 9 (3^2) cells while a macroparticle with a smooth shape function in 3D overlaps 125 (5^3) cells.

2.3.3 Step 2: Particle movement

In this step, the macroparticles are moved according to the laws of motion for a particle in electromagnetic field (2.8) - (2.10). In a non-relativistic case the relativistic factor γ is set to 1.

$$\frac{\partial \boldsymbol{x_k}}{\partial t} = \frac{\boldsymbol{p_k}}{\gamma m_k} \tag{2.8}$$

$$\frac{\partial \boldsymbol{p_k}}{\partial t} = q_k \left(\boldsymbol{E_k} + \frac{\boldsymbol{p_k}}{\gamma m_k} \times \boldsymbol{B_k} \right) \tag{2.9}$$

$$\gamma = \sqrt{1 + \left(\frac{\boldsymbol{p_k}}{m_k c} \right)^2} \tag{2.10}$$

Various discrete schemes for the calculation of the motion of particles can be used. The so-called leap-frog method is often utilized in many PIC codes. This method computes momentum one half-step after the calculation of position. This leads to higher order and stability of the scheme.

EPOCH utilizes the leap-frog scheme, but the positions are saved every half-step. In order to move from the time $t = n\Delta t$ to $t = (n+1)\Delta t$, the macroparticle is first moved to the position $x^{n+1/2}$ using the momentum p^n. The interpolation occurs in time $t = (n + 1/2)\Delta t$. After the interpolation, p^{n+1} is calculated and the macroparticle is moved to x^{n+1}. This particle movement scheme ensures second order accuracy [5].

2.3.4 Step 3: Current density calculation

After the movement of the particles, the current densities are calculated. This step is similar to the interpolation step, there are however important differences. In this case we interpolate charge density from the macroparticles onto the grid. In the same scenario as in subsection 2.3.2, the contribution of the macroparticle p to the quantity F_j^p at gridpoint x_j is calculated as (2.11) [40], where F_p is the quantity assigned to the macroparticle.

$$F_j^p = F_p \cdot \begin{cases} \frac{3}{4} - \frac{|X-x_j|^2}{\Delta x^2}, & |X - x_j| \leq \frac{\Delta x}{2} \\ \frac{1}{2}\left(\frac{3}{2} - \frac{|X-x_j|}{\Delta x}\right)^2, & \frac{\Delta x}{2} < |X - x_j| \leq \frac{3\Delta x}{2} \\ 0, & \frac{3\Delta x}{2} < |X - x_j| \end{cases} \tag{2.11}$$

This calculation can be equivalently formulated using the so-called interpolation function W (2.12). The interpolation function is just a convolution of the shape function with S_0.

$$F_j^p = F_p W(X - x_j) = F_p (S_n * S_0)(X - x_j) = F_p S_{n+1}(X - x_j). \tag{2.12}$$

Omitting diagnostic output we usually interpolate only the charge density. EPOCH uses the Villasenor-Buneman scheme to calculate the current density from the interpolated charge density. In this scheme, after the particles are pushed from $x^{n-1/2}$ to $x^{n-1/2}$ a central numerical difference is used to calculate $\partial\rho/\partial t$. Using the law of conservation of density (2.13) with divergence replaced by one sided differences, the current density can be calculated.

$$\nabla \boldsymbol{j} = \frac{\partial \rho}{\partial t} \tag{2.13}$$

Advantage of this scheme is that if the Poisson equation (2.14) is satisfied after the initialization of the simulation, it will be satisfied at all time-steps [41].

$$\nabla^2 \phi = \frac{\rho}{\epsilon_0} \tag{2.14}$$

2.3.5 Step 4: Field calculation

The last step in the cycle is the field calculation. This can be done in multiple ways. Early 1D electrostatic PIC codes solved the Poisson equation (2.14) numerically [3]. Modern codes require calculation of magnetic fields, which complicates the computation.

EPOCH uses finite difference time domain method (FDTD). Fields are calculated on a Yee grid. In 3D this means that the electric fields are calculated in the middle of the faces of a cubical cell while magnetic fields are calculated in the middle of the edges of the cube [42]. This special distribution of gridpoints allows us to use central differences, which in turn leads to second order accuracy.

Assuming that in previous step we have calculated $\boldsymbol{j}^n$, we can use discretized form of (1.3) and (1.4) to calculate the fields in the next step as follows:

$$\begin{aligned}
\boldsymbol{B}^n &= \boldsymbol{B}^{n-1/2} - \frac{\Delta t}{2}\left(\nabla^+ \times \boldsymbol{E}^{n-1/2}\right), \\
\boldsymbol{E}^n &= \boldsymbol{E}^{n-1/2} + \frac{\Delta t}{2}\left(c^2\nabla^- \times \boldsymbol{B}^n - \boldsymbol{j}^n\right), \\
\boldsymbol{E}^{n+1/2} &= \boldsymbol{E}^n + \frac{\Delta t}{2}\left(c^2\nabla^+ \times \boldsymbol{B}^n - \boldsymbol{j}^n\right), \\
\boldsymbol{B}^{n+1/2} &= \boldsymbol{B}^n - \frac{\Delta t}{2}\left(\nabla^- \times \boldsymbol{E}^{n+1/2}\right).
\end{aligned} \tag{2.15}$$

In this calculation ∇^+ and ∇^- are difference operators on the Yee grid. ∇^+ is a forward difference operator while ∇^- is a backwards difference operator [5].

Equations (2.15) are not complete. We need to define boundary conditions, which mostly change the calculation of the fields only on the boundaries of the simulated space, but can also affect the calculations at inner gridpoints. EPOCH allows the users to choose from multiple different boundary conditions.

2.4 Stability and numerical effects

Knowledge of stability conditions is crucial for all numerical methods. Method is unstable when errors in the partial solutions add up to physically unreasonable results [43]. In PIC codes this instability most often presents itself as drastic increase in the energy of the simulated system.

Stability itself is not a guarantee of physically sound results. Even if the simulation is stable its results may be affected by various numerical artifacts. In this section the stability conditions will be presented as well as other numerical effects important for practical use of PIC codes.

2.4.1 Parameters

PIC algorithm has a few general parameters. These parameters are important for accuracy, stability and computational demands of any simulation.

Two most important parameters are the time-step Δt and the spatial step Δx, also referred to as cell width. In 2D and 3D simulations spatial steps in other directions Δy and Δz are also important. In most cases the spatial steps are all equal. Therefore, if not specified otherwise, we will use Δx to denote spatial step in all directions.

Parameter that is most relevant for this thesis is the macroparticle number. We will denote this parameter as N_{ppc}. During the initialization of simulation each cell that contains matter is initialized with N_{ppc} macroparticles in it. Due to this, N_{ppc} is not the total amount of particles divided by the total amount of cells. In many codes a lower limit can be set so that particles are initialized only in cells with density higher than a given threshold.

The choice of shape function can be considered a parameter. Shape function can have a tremendous impact on stability and accuracy. Therefore we can consider it a parameter for the purpose of this thesis.

Boundary conditions have similarly strong impact on the stability and numerical accuracy of the simulation. There are separate boundary conditions for fields and particles. Standard options for fields are periodic, open and absorbing conditions. These can be simple, affecting only the calculation of fields near the boundary or more complex affecting the optical properties of the whole simulated area to achieve better results. One example of the latter are the convolutional perfectly matched layers boundary conditions, which are also implemented in EPOCH.

In the case of particle boundary conditions it is necessary to determine what happens when a macroparticle moves beyond the boundary. Options are reflective, periodic or open boundaries.

2.4.2 CFL condition

The Courant–Friedrichs–Lewy (CFL) condition is a necessary stability condition for simulation of electromagnetic waves [44]. In the case of PIC it has the form of (2.16). If this condition is not satisfied the electromagnetic waves will propagate nonphysically.

$$c\Delta t \leq \Delta x \tag{2.16}$$

This condition coincidentally also prohibits macroparticles from moving further than one cell per one time-step. This is desirable, since otherwise the calculation of currents would be inaccurate.

EPOCH has this condition automatically implemented. In its 2D version EPOCH sets the time-step according to (2.17).

$$\Delta t = \frac{0.95}{c} \cdot \frac{\Delta x \cdot \Delta y}{\sqrt{\Delta x^2 + \Delta y^2}} \tag{2.17}$$

In case that all the spatial steps are equal, the formula for the time-step will be simplified into (2.18), where D is the dimension of the code.

$$\Delta t = \frac{0.95}{c} \cdot \frac{\Delta x}{\sqrt{D}} \tag{2.18}$$

2.4.3 Electron plasma frequency condition

In order for the macroparticles to represent the propagation of plasma waves, it is necessary for them to be able to undergo oscillations with the electron plasma frequency. Analysis of macroparticles undergoing such oscillations has been presented in previous thesis [1]. The relevant result is the relationship (2.19) between the electron plasma frequency ω_{pe} and frequency in the simulation ω.

$$\frac{\omega_{pe}\Delta t}{2} = \pm sin\left(\frac{\omega\Delta t}{2}\right) \tag{2.19}$$

In order for this equation to have only real roots, it is necessary to fulfill the condition (2.20). If this condition is not met the imaginary roots in simulated frequency would result in nonphysical change of amplitudes of plasma waves, which would result in numerical instability.

$$\omega_{pe}\Delta t \leq 2 \tag{2.20}$$

Even if this condition is met, the simulated waves have slightly deformed frequency which is undesirable. It is therefore better to have even smaller time-step than the condition requires.

2.4.4 Effects due to finite grid

Grid introduces numerical artifacts to the simulation. Fourier analysis allows us to describe the effect of aliasing [45]. Aliasing occurs when waves of higher frequencies cannot be represented on a discrete grid. Such waves merge with waves of lower frequencies. In PIC this effect leads to nonphysical instability.

We will theoretically describe the effects of aliasing of plasma waves in 1D PIC simulations. The highest wavenumber that can be represented by the grid is given by:

$$k_{max} = \frac{\pi}{\Delta x}. \tag{2.21}$$

If we try to represent waves with higher wavenumbers they will appear to the grid as waves of lower frequency. Modes k_p are aliased with mode k. This relationship is described by (2.22), where $p \in \mathbb{Z}$.

$$k_p = k - 2 \cdot p \cdot k_{max} \tag{2.22}$$

If we analyze the dispersion relation for plasma waves considering the effects of aliasing, we will get the imaginary part of the dispersion relation in the form of (2.23) [3], where f_0 is the velocity distribution function.

$$Im\,(\epsilon(\omega, k)) = -\pi \frac{\omega_{pe}^2}{k^2} \sum_p \hat{S}^2(k_p) \frac{k k_p}{k_p^2} \frac{\partial f_0(\omega/k_p)}{\partial v}. \tag{2.23}$$

This would be equivalent to the Landau damping if there was no sum over the aliases. Aliasing causes nonphysical instability since the energy the wave loses is not the same as the energy the particles receive. If the condition (2.24) is not satisfied, modes that are affected by Landau damping are aliased with those that aren't.

$$\frac{\omega_{pe}}{k_{max}} \leq v_{Te} \Rightarrow \Delta x \leq \frac{v_{Te}\pi}{\omega_{pe}} \Rightarrow \Delta x \leq \pi\lambda_{De} \tag{2.24}$$

In (2.23) the aliases are summed over with a factor of $\hat{S}^2(k_p)$. Higher order shape functions thus result in diminished effect of aliasing on dispersion relation. The condition (2.24) is therefore not so relevant for the stability of codes with higher order shape functions. It is important to note however that aliasing is still present, but it doesn't lead to dramatic heating which can be observed in PIC simulations using the lower order shape function [1].

2.4.5 Effects due to macroparticles

Replacing the distribution function with macroparticles introduces numerical artifacts as well. In this subsection we will investigate the effects of the macroparticle number on the accuracy of PIC simulations.

A general formula (2.25) for the rate of nonphysical heating occurring in PIC simulations has been presented in [4].

$$r \propto \frac{\Delta x^2}{N_{ppc}} \tag{2.25}$$

The formula (2.25) has been derived for a one-dimensional electrostatic PIC code, which solves the Poisson equation to obtain the electric fields. Modern PIC codes, such as EPOCH, calculate the electromagnetic fields in significantly different way (see section 2.3). The formula (2.25) may therefore not hold for such PIC codes. For this reason, we will attempt to derive the general dependence of errors on the macroparticle number in PIC simulations, which utilize the same core schemes and algorithms as EPOCH.

To obtain such a dependence we will analyze the effect of macroparticle number during the third step of the PIC algorithm. In this step the charge density is interpolated onto the grid. As an example we will work with 1D Villasenor-Buneman scheme which is utilized in EPOCH. The law of conservation of charge (2.13) in discretized form (2.26) allows us to calculate the current densities [5]. We use superscript to denote time-step and subscript to denote position on the grid.

$$j_{k+1}^{n+1} - j_{k}^{n+1} = -\frac{\Delta x}{\Delta t}\left(\rho_{k+\frac{1}{2}}^{n+\frac{3}{2}} - \rho_{k+\frac{1}{2}}^{n+\frac{1}{2}}\right) \tag{2.26}$$

This can be rewritten as (2.27) using the interpolation function W. We use X to denote position of a particle and x to denote position of a gridpoint to avoid confusion. The summation in the expression is performed over the set of all particles.

$$j_{k+1}^{n+1} - j_{k}^{n+1} = -\frac{\Delta x}{\Delta t}\sum_{p} n_p q_p \left[W\left(X_p^{n+\frac{3}{2}} - x_{k+\frac{1}{2}}\right) - W\left(X_p^{n+\frac{1}{2}} - x_{k+\frac{1}{2}}\right)\right] \tag{2.27}$$

We will attempt to approximate errors in the current calculation that occur due to small macroparticle number. First we assume that only the electrons contribute to the errors in calculation. This assumption is reasonable since the ions are heavier by several orders of magnitude and thus slower, therefore they do not contribute to the current calculation as much as the electrons.

Considering that contributions of the ions are negligible, we can express the right side of (2.27) as a sum of partial contributions by each macroparticle that represents electrons. The amount by which each electron macroparticle contributes to the calculation is dependent on particle position $X_p^{n+\frac{1}{2}}$, velocity $V_p^{n+\frac{1}{2}}$ and the position of the gridpoint where we calculate the charge density x_k.

Under assumption that each macroparticle represents the same number of real particles, we can rewrite (2.27) as (2.28), where Q_e^{tot} is the total charge of electrons in the simulations and N_e^{tot} is the total amount of electron macroparticles. R represents the change in density obtained by interpolation before and after the movement of a particle, which in (2.27) is represented by the expression in the square brackets. We also introduce $J_{k+\frac{1}{2}}$ to shorten the expressions in the following text.

$$j_{k+1}^{n+1} - j_{k}^{n+1} = J_{k+\frac{1}{2}} = -\sum_{p\in e^-} q_p R\left(X_p^{n+\frac{1}{2}}, V_p^{n+\frac{1}{2}}, x_{k+\frac{1}{2}}\right) = -\frac{Q_e^{tot}}{N_e^{tot}}\sum_{p\in e^-} R\left(X_p^{n+\frac{1}{2}}, V_p^{n+\frac{1}{2}}, x_{k+\frac{1}{2}}\right) \tag{2.28}$$

We can notice that the right side of the equation (2.28) can be considered the mean of all partial contributions of the macroparticles (2.29), where we denote the mean of a variable with a bar. If we assume that the positions $X_p^{n+\frac{1}{2}}$ are independent and identically distributed random variables (*iid*) with uniform distribution and that the velocities are *iid* with Maxwellian distribution we can treat the current density calculation and its error using simple statistical methods.

$$J_{k+\frac{1}{2}} = -Q_e^{tot} \cdot \overline{R} \tag{2.29}$$

In order to determine the errors occurring in the current calculation we will be interested in the standard deviation of $J_{k+\frac{1}{2}}$. Thanks to assumptions we have made we can apply the Bienaymé formula (2.30).

$$\sigma^2\left(J_{k+\frac{1}{2}}\right) = Var\left(J_{k+\frac{1}{2}}\right) = Q_e^{tot} \cdot Var\left(\overline{R}\right) = Q_e^{tot} \cdot \frac{Var(R)}{N_e^{tot}} \tag{2.30}$$

This analysis provides us with general dependence of root mean square of the errors in the current calculation on the macroparticle number (2.31). Note that the variance of the interpolation function ($Var(R)$) depends on the shape function used.

$$\sigma\left(J_{k+\frac{1}{2}}\right) \propto \sqrt{\frac{Var(R)}{N_{ppc}}} \tag{2.31}$$

Strong assumptions we have made in the model presented in this subsection enable us to determine only the general dependence of numerical accuracy of the simulation on the macroparticle number. It is important to note that in simulations the effects of finite grid and macroparticles are interconnected. In real use of PIC codes one must take care to ensure that all desired physical mechanisms are accurately simulated.

2.4.6 Current smoothing

In order to suppress the undesirable numerical effects we can utilize current density smoothing. This method uses convolutional smoothing of the calculated currents. Current smoothing of the x components of the currents in 2D follows the formula (2.32), where $\mathbb{J}_x$ is the matrix of the x components of the current calculated at the gridpoints.

$$\mathbb{J}_x^s = \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} * \mathbb{J}_x. \tag{2.32}$$

In Fourier space this convolutional smoothing acts as a low pass filter. This means that higher frequencies are suppressed. Aliasing has thus smaller effect on the simulation which is beneficial.

Negative effects mentioned in the previous subsection are also reduced by smoothing. Theoretical descriptions of the effects of current smoothing in PIC algorithm is outside of the scope of this thesis. We will discuss these effects further in the fourth chapter of this thesis along with simulation results.

2.5 EPOCH

Extendable PIC Open Collaboration (EPOCH) is a parallel relativistic PIC code developed at the University of Warwick. It is written in Fortran 90 using MPI (message passing interface) for parallelization. It is based on an older code PSC developed by Hartmut Ruhl [5].

The latest version of the code is 4.6.2 released in October 2016. In February 2015 the official support for development of EPOCH has been dropped. This has slowed down the advancement of the project, but occasional updates and bugfixes are still being released. EPOCH can be downloaded from GitHub repository mantained by the University of Warwick [46].

EPOCH has several important features. It includes various optional physics models to simulate interesting phenomena such as collisions of particles, QED mechanisms and others. Many tools for visualization and analysis of data generated by the algorithm are provided in the standard download of the code. Dynamic load-balancer is also implemented within the code. This feature dynamically distributes the calculations among the processing units (CPUs) to ensure efficient utilization of computational resources.

Codes for each 1D, 2D and 3D calculations are separate and need to be compiled separately. Compilation requires MPI 1.2 compatible libraries. Editable makefile enables the user to specify some properties of the compiled code. Most important for this thesis is the ability to specify the shape function.

2.5.1 Additional diagnostic implementation

EPOCH in version 4.6.2 allows users to calculate the total energy of all particles in the simulation, however the ability to calculate total energy of a specific particle type was missing. For purposes of this thesis the code was altered to add this diagnostic option.

In order to add a new diagnostic option the file `src/io/diagnostics.F90` was edited. A subroutine that calculates the total energy of a specific particle type was added along with a subroutine to perform this calculation for all particle types and print the results in the output file. Minor changes to the files `src/shared_data.F90` and `src/deck/deck_io.F90` were also necessary in order to let the program know that a new diagnostic option was added.

2.6 Computational resources

Performing multiple PIC simulations requires access to computational resources. Most of the simulations and their analysis was performed on computational resources provided by Metacentrum VO. Computationally demanding 3D simulations were performed on cluster ARIS.

2.6.1 Metacentrum VO

Metacentrum virtual organization (VO) is open to all academical students and researchers in the Czech Republic. Registered users are granted access to computational resources provided by the collaborating institutions. The organization manages development, user support and security of the whole system [47]. Metacentrum provides over 13000 CPUs for over a thousand users. Users can use most of the computational resources without limitation, however members of some institutions that provide computational resources to Metacentrum have access to priority queues. Metacentrum also provides data storage, data archivation and cloud services.

In order to use the resources of Metacentrum users have to submit jobs to queues. Details of job submission and other practical details are discussed in the following section.

2.6.2 ARIS

ARIS is the name of a Greek supercomputer, deployed and operated by GRNET (Greek Research and Technology Network) in Athens. ARIS consists of 532 computational nodes. 442 of those are so-called thin nodes. Each of them has 20 cores at 2.8 GHz with 64 GB RAM. This kind of architecture is suitable for demanding computational tasks that are able to utilize MPI efficiently [48].

2.7 Running simulations and output processing

In this section we will briefly introduce the procedures necessary to run PIC simulations using EPOCH.

2.7.1 input.deck

Initial conditions and general settings of the simulations are determined by a text file with designated name "input.deck". The file is organized into blocks. Each block contains settings specific to a certain part of the simulation. Control block sets up the simulation size and grid properties. Boundary and Laser blocks set up the boundary conditions and properties of the incoming laser wave respectively. Particles are set up using species block. Each type of particle needs to have its own species block. Constant block can be used to set up constants that can be used in other blocks for easier changes in the deck file. Diagnostic outputs have to be set up in Output blocks. It is possible to output a restart file that can later be used to start the simulation from a certain point. This is useful if the user is not sure that the simulation will finish before the scheduled time for a job is up or as a means to check partial results in the simulation in order to prevent excessive expenditure of computational resources. More detailed description of an input.deck file has been presented in previous thesis [1].

When dealing with multiple simulations at once it is useful to use an import line within the input.deck file. This causes the program to replace that line with the contents of the file imported. This enables us to have one primary deck file with most of the settings, while having separate input.deck files, which set up only the setting which we want to be varied among the simulations.

2.7.2 Scheduling system

Cluster computing requires a system which enables the users to run their programs without affecting the programs run by other users. Such systems are called scheduling systems and can be implemented in slightly different ways, however the general idea is usually the same. User submits a job which has specified resources it will need to run. A job can be a single command, but usually it is a shell script executing multiple commands. Some scheduling systems allow for interactive jobs as well. After a job is submitted it is put in a queue, where it waits until the resources it needs are available. Once that happens the job is executed. Queues may utilize a fairshare mechanism, which prioritizes jobs by users who have used fewer resources in recent time.

For the past 6 years Metacentrum VO used Portable Batch System (PBS) Torque. In 2016 Metacentrum started migration to a new scheduling system PBS Professional.

A typical script used to run multiple PIC simulations in PBS Pro has the following form:

```
 1  #!/bin/bash
 2  #PBS -N epoch2DFull
 3  #PBS -l walltime=48:00:00
 4  #PBS -l select=1:ncpus=32:mem=4gb
 5  #PBS -j oe
 6  #PBS -m e
 7
 8  module add openmpi-1.6.5-intel
 9
10  cd /storage/brno6/home/kocurvik/EPOCH/epoch-4.6.2/epoch2d
11
12  mv MakefileST Makefile
13  make clean
14  make COMPILER=intel
15  chmod -R 777 bin
16  echo Data/ST/60P | mpirun -np 32 bin/epoch2d
17  ...
18  echo Data/ST/03P | mpirun -np 32 bin/epoch2d
19
20  mv Makefile MakefileST
```

First commented line of the script specifies the shell which the script should run in. In the next lines, the properties of the job are specified. First two lines of those are the name of the job and the amount of time to be allocated for the job. It is necessary to allocate enough time to run the job with the "walltime" parameter, but selecting longer times leads to longer waiting times to run the job. Line 4 specifies the amount of nodes, cpus per node and memory (RAM) per node the job needs to run on. Last two commented lines specify that the standard output and error streams should be saved in a text file. Line 6 specifies that the standard error should also be sent by email to the user. It is possible to specify some other parameters of the job such as type of machine or a queue to which the job should be added.

The actual commands are in the uncommented lines. First, it is necessary to load the MPI libraries. Compilation of EPOCH comes next followed by the execution of the simulations. Manipulation of the makefile before the compilation is necessary since we need different makefiles for simulations with different shape functions.

In order to submit a job to the system it is necessary to remotely log in to a frontend of Metacentrum. Calling a command "qsub job.sh" will submit the job to the system. If there are machines satisfying the desired properties, the job is submitted to a queue.

2.7.3 Output processing

EPOCH produces output in the form of .sdf (self-defining format) files. Part of the EPOCH repository is a folder with multiple tools to read these files in various third party programs. One of the programs that can be used to read the output files is MATLAB, which has been used for the purposes of this thesis.

Matlab function "GetDataSDF(file.sdf)", provided in the EPOCH repository, returns an array of data saved in "file.sdf". We use this function within bodies of Matlab functions to analyze the output of simulations and create visualizations from it.

For purposes of this thesis we ran multiple simulations and therefore it was necessary to run Matlab scripts on the results of multiple simulations at the same time. In order to achieve this we run a Matlab script which runs a second script on the results of a specific simulation. Examples of such scripts have been presented in previous thesis [1].

Chapter 3

Simulations and results

This chapter deals with the simulations performed for the purposes of evaluating the dependence of numerical accuracy on macroparticle number.

In this chapter we will use the terms triangular shape function (ST) and smooth shape function (S3) , which are described in section 2.2. Simulations using the triangular shape function with the current smoothing algorithm enabled will be referred to with the label "ST-S" to distinguish them from the simulations with the current smoothing algorithm disabled (ST, S3).

3.1 Simulation parameters

Simulations used for this thesis were set-up to represent a laser pulse perpendicularly incident to a fully ionized hydrogen foil at solid density.

The simulated area spans 30 μm in the x-direction and 4 μm in the y-direction. In 3D simulations, the size of the area in the z-direction is also 4 μm. The hydrogen target is positioned 14 μm from the left boundary of the simulated area and is 992 nm thick. The simulation layout can be seen in fig. 3.1. The hydrogen target has a density of 5.22 $\cdot 10^{22}$ particles per cm^3. Its initial temperature is set to 500 eV, which is equivalent to roughly 5.8$\cdot 10^6$ K. The particles are frozen during the first 40 fs of the simulation. That is the time it takes for the laser pulse to almost reach the target. Freezing the particles during the initial phase of the simulation allows for reduced computational demands and negligible impact of nonphysical plasma expansion. The simulation finishes after 160 fs, when the reflected pulse had enough time to leave the simulation area.

The incident laser pulse is represented by a plane wave. This allows us to simulate the interaction of the pulse with the target in a smaller simulation area. The plane wave propagates into the simulated area along the x-axis. The wavelength of the pulse is 800 nm, which is equivalent to a laser period of around $2.67 \cdot 10^{-15}$ s. The pulse is 10 periods long, which is roughly 27 fs. Time profile of the wave is defined by (3.1), where A_0 is the maximum amplitude of the pulse and T_0 is the laser period. Such a time profile of the pulse approximates a Gaussian temporal profile of the intensity of the laser.

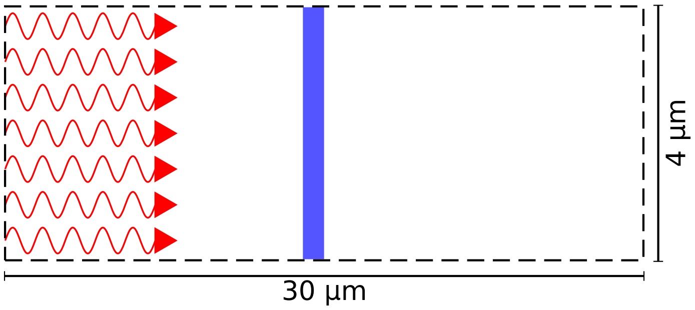

Figure 3.1: Initial configuration of the simulation. Red waves symbolize the direction in which the laser pulse is propagating.

$$A = A_0 \sin\left(\frac{2\pi t}{20T_0}\right) \tag{3.1}$$

The Debye length of the target is around 0.73 nm. The electron plasma frequency is approximately $1.29{\cdot}10^{16}$ rad/s, while the laser frequency is around $2.35{\cdot}10^{15}$ rad/s. The critical density for the laser pulse n_c is $1.74\ {\cdot}10^{21}$ cm^{-3}. The target is therefore overdense ($n_e = 30\ n_c$) and the pulse will be reflected.

The boundary condition used on the boundary from which the laser comes is set to be "simple_laser". This ensures that the laser pulse gets represented in the simulation properly and that any electromagnetic waves reflected by the target get absorbed by the boundary. The boundary condition used on the opposite end of the simulation is set as "simple_outflow". This boundary condition ensures that any incoming electromagnetic pulse is absorbed by the boundary and not reflected back into the simulation area. Any macroparticles that reach any of these two boundaries will be removed from the simulation.

Since the laser is represented by a plane wave, the boundary conditions on the remaining boundaries are all periodic, both in respect to the particles and the fields.

2D simulations were run with the cell sizes of 8, 16 and 64 nm. In simulations with cell size of 64 nm the size of the simulation area has been slightly modified to 4.032 μm × 30.016 μm. For all cell sizes, the 2D simulations were run with the triangular shape function with and without current smoothing (ST and ST-S) and with the smooth shape function without current smoothing (S3). 3D simulations were run with the cell size of 16 nm using the triangular shape function. All the simulations were run in batches with macroparticle numbers equal to 3, 6, 10, 15, 30 and 60 for both electrons and protons.

The maximum cell size in 2D simulations that fulfills the plasma frequency stability condition (2.20) is 69.2 nanometers for cases when the time-step is set by the default EPOCH setting (2.18). All of the 2D simulation setups thus fulfilled this condition. The cell size of 64 nm was specifically selected to observe how the macroparticle number affects the accuracy of a simulation that is close to being numerically unstable. The cell size choice in the 3D simulations guarantees that the plasma frequency condition is fulfilled.

3.2 Numerical effects

This section deals with the numerical effects observed in the performed simulations. The effects are quantified and discussed with regard to theoretical predictions.

3.2.1 Energy balance

The easiest way to observe nonphysical numerical effects in simulations is to evaluate the time evolution of the total energy in the simulations. Instability in the simulation presents itself most often via rapid heating of the particles [1]. This was not the case in the performed simulations, however differences in the accuracy of the simulations with varying macroparticle number and other parameters are apparent.

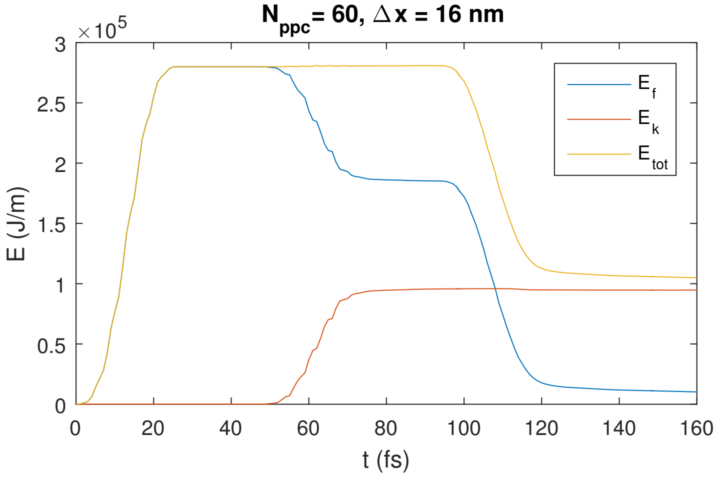

Figure 3.2: Development of the total field energy (E_f), total kinetic energy (E_k) and their sum (E_{tot}) in 2D simulation using the triangular shape function (ST) with 16 nm wide cells and 60 particles per cell.

Fig. 3.2 shows how the total energy of the electromagnetic field and the particles evolve in 2D simulation with the triangular interpolation function, 60 particles per cell and cell size of 16 nm. During the first 27 fs of the simulation, the energy of the electromagnetic fields increases as the laser pulse is entering the simulation area. At around 50 fs the laser pulse reaches the target. During the following 20 fs, the pulse interacts with the target. The energy of the pulse gets partially absorbed by the target resulting in the increase of the kinetic energy of the particles. The pulse is reflected by the target and leaves the simulation area 95 fs after the start of the simulation. At time 120 fs after the start of the simulation, the reflected pulse leaves the simulation area completely. After the pulse has left the simulation area, the total energy of the electromagnetic field slowly decreases due to smaller electromagnetic waves leaving the simulation area as well. The total kinetic energy of the particles remains almost constant after the interaction of the pulse with the target.

Observing Fig. 3.3 representing an energy plot for a simulation with the same parameters, but with only 3 particles per cell, we can notice that after the interaction of the pulse with the

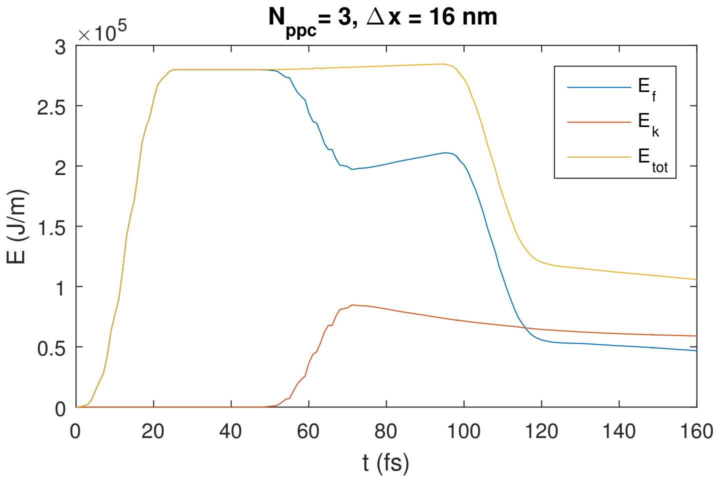

Figure 3.3: Development of the total field energy (E_f), total kinetic energy (E_k) and their sum (E_{tot}) in 2D simulation using the triangular shape function (ST) with 16 nm wide cells and 3 particles per cell.

target the kinetic energy of the particles decreases. We can also observe that the energy of the particles has been transformed into the energy of the electromagnetic field. The energy of the fields then decreases in the simulation as the electromagnetic waves leave the simulation area. It is also important to note that the total energy in the simulation slightly increases in the time interval from 55 fs to 100 fs. This nonphysical effect is due to numerical heating in the simulation described in subsection 2.4.4.

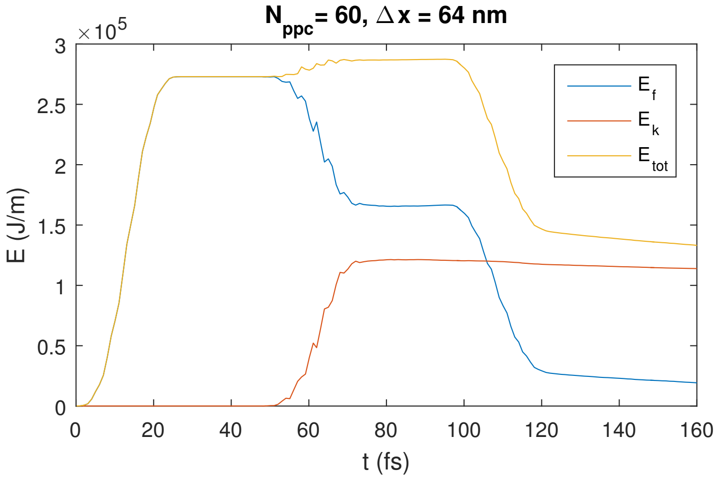

Figure 3.4: Development of the total field energy (E_f), total kinetic energy (E_k) and their sum (E_{tot}) in 2D simulation using the triangular shape function (ST) with 64 nm wide cells and 60 particles per cell.

Fig. 3.4 represents the energy plot of a 2D simulation using the triangular shape function with cell size of 64 nm and 60 particles per cell. We can notice that the numerical heating effect is stronger than in simulations with smaller cell sizes. Such simulation can still be considered stable, since the the strong numerical heating happens only for a short amount of time and after that the total energy remains somewhat constant. However, the numerical heating indicates that the simulation is not suitable for observation of physical phenomena, since any phenomena may have been significantly affected by the numerical effects.

3.2.2 Absorption

Absorption of the pulse energy by ionized targets is an important topic in the physics of laser plasma. PIC simulations are often used to study various absorption mechanisms. For purposes of this thesis we will only deal with the absorption coefficient and its dependence on the macroparticle number and other simulation parameters. The absorption coefficient is defined as (3.2), where $\max(E_k)$ is the maximum total kinetic energy of particles in the simulation, $E_k(0)$ is the initial total kinetic energy of the particles and E_p is the total energy of the incoming pulse. We define the absorption in this way in order for the coefficient not to be affected by the cooling of particles which occurs in some of the simulations.

$$C_{abs} = \frac{\max(E_k) - E_k(0)}{E_p} \tag{3.2}$$

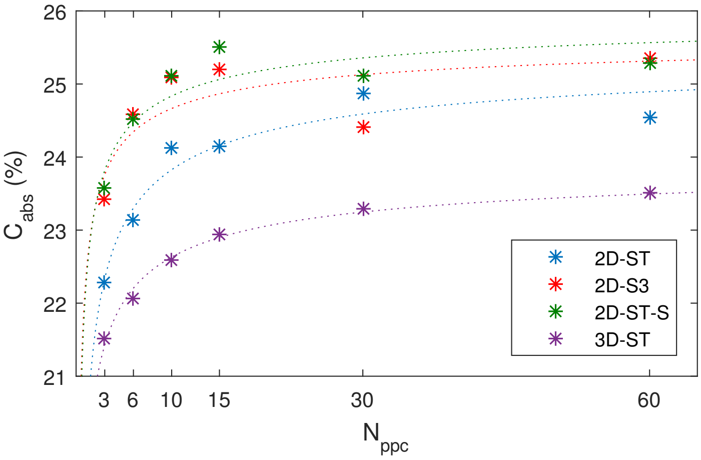

Figure 3.5: Absorption coefficients from 2D simulations with cell size of 8nm for multiple settings of shape function and current smoothing (2D-ST, 2D-S3, 2D-ST-S). Results from 3D simulations using the triangular shape function with cell size of 16 nm (3D-ST) are included for reference. Dotted lines represent fitted curves.

Absorption coefficients from 2D simulations with cells 8 nm wide are plotted in Fig. 3.5. Coefficients from simulations with cells 16 nm wide are plotted in Fig. 3.6. Both figures include results from 3D simulations with cells 16 nm wide.

In subsection 2.4.5 we proposed a relationship between the errors in simulations and the macroparticle number (2.31). We can fit the absorption coefficients with respect to the proposed

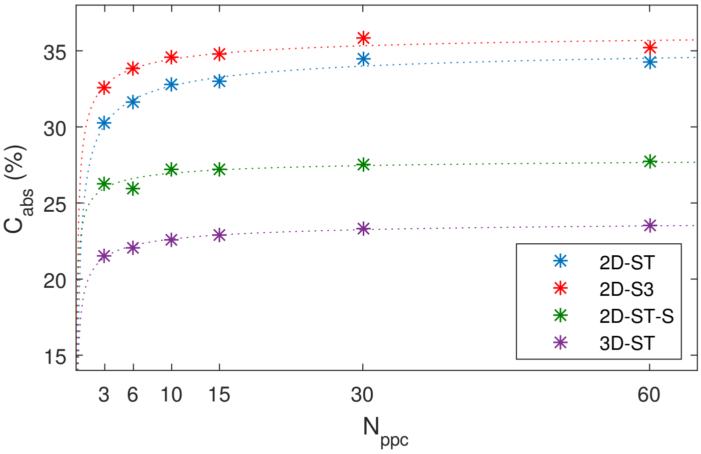

Figure 3.6: Absorption coefficients from 2D simulations with cell size of 16 nm for multiple settings of shape function and current smoothing (2D-ST, 2D-S3, 2D-ST-S) and from 3D simulations using the triangular shape function with cell size of 16 nm (3D-ST). Dotted lines represent fitted curves.

DIM	Δx	SF	A	B	RMSE	R^2
2D	8 nm	ST	-5.703	25.63	0.2818	0.9315
		S3	-3.464	25.76	0.4905	0.6234
		ST-S	-3.904	26.07	0.3455	0.8091
	16 nm	ST	-9.734	35.77	0.3253	0.9675
		S3	-6.629	36.54	0.3486	0.9231
		ST-S	-3.738	28.14	0.3967	0.7467
3D		ST	-4.616	24.09	0.089	0.9889

Table 3.1: Fitting for a relationship in the form of (3.3). Columns from left to right: dimension of the simulation, cell width, shape function and smoothing used, parameters of the fitted function A and B, root mean squared error and r-squared statistics.

relationship. We will fit the absorption coefficients in the form of (3.3), where A and B are the parameters we will determine using the Matlab curve fitting tool [49].

$$C_{abs}(N_{ppc}) = \frac{A}{\sqrt{N_{ppc}}} + B \tag{3.3}$$

Table 3.1 contains the parameters of the fitted curves. Values in the columns RMSE and R^2 indicate the goodness of fit. Root mean squared error (RMSE) is also known as the fit standard error and the standard error of the regression. It is an estimate of the standard deviation of the random component in the data. R-square statistic measures how successful the fit is in explaining the variation of the data. Put another way, R-square is the square of the correlation between the response values and the predicted response values. Values closer to 1 are better [50].

For most of the variants of the simulations, the fit is good enough to consider the relationship presented in subsection 2.4.5 to be relevant to estimate how the error depends on the macroparticle number. Fit of the results of the 3D simulations is especially good, which is promising since the total amount of macroparticles in 3D simulations is higher and therefore the dependence of the error in the absorption coefficient on the macroparticle number should be more consistent. Simulation settings for which the value of R-square is under 0.9 all have one or two data points, which are far from the fitted curve. The fact that these points lay far from the curve can be attributed to the stochastic nature of the errors, since the model we have presented for the relationship in the subsection 2.4.5 is derived with regards to stochastic errors.

The parameter B in this case is easy to interpret. According to our model, it is the value of the absorption coefficient that would be obtained in a simulation with an infinite macroparticle number. When considering the accuracy of this parameter it is important to take the rooted mean square error of the fit into account.

We can observe that simulations with cell size of 16 nm have significantly different absorption coefficients, which also presents itselfs in the differences in the parameter B. The difference between the absorption coefficients from 2D simulations with and without current smoothing is especially interesting. Such difference leads us to assume that the simulations with 16 nm wide cells are not accurate enough to represent physical phenomena relevant to the absorption of the energy of the laser pulse by the target.

Smaller differences can be observed also for simulations with smaller cell size. These smaller differences can be explained by the fact that some physical phenomena are approximated differently with the use of different shape functions and smoothing. Difference between the absorption coefficient in 2D and 3D simulation can be explained by the inability of 2D simulations to represent some physical processes at all. However, using the smaller cell size, the absorption coefficient are close enough.

The parameter A tells us how strong is the dependence of the accuracy of the simulation on the macroparticle number. As predicted, using only the triangular shape function without current smoothing leads to stronger negative impact of low macroparticle number. The fact that the current smoothing combined with the triangular shape function reduces the value of A quite successfully in the simulations with 16 nm wide cells is quite an interesting observation. In simulations with that cell size the smoothing has brought the parameter B closer to the results of simulations with the smaller cell size. However, in simulations with cells 8 nm wide, the use of the smooth shape function reduced the value of A more than smoothing of the current.

3.2.3 Cooling coefficient

As seen in the figures in subsection 3.2.1, there is a noticeable cooling of particles in simulations with smaller macroparticle number. We will quantify the rate at which this occurs. In order to do so we will use the cooling coefficient (3.4). This coefficient is defined for a given time t using the total kinetic energy of particles in the simulation $E_k(t)$ as follows:

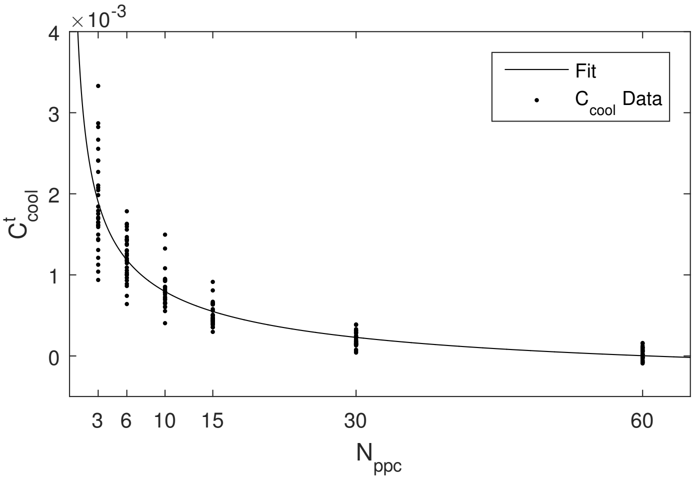

Figure 3.7: Dots represent cooling coefficients for times 129 fs, 130 fs ... 159 fs from the start of the simulation in 2D simulation with cells 16 nm wide, using the triangular shape function (ST). Line represents the fitted curve in the form of (3.5).

DIM	Δx	SF	A	B	RMSE	R^2
2D	8 nm	ST	$3.29{\cdot}10^{-3}$	$-4.58{\cdot}10^{-4}$	$1.25{\cdot}10^{-4}$	0.9397
		S3	$1.94{\cdot}10^{-3}$	$-3.46{\cdot}10^{-4}$	$8.92{\cdot}10^{-5}$	0.9137
		ST-S	$1.33{\cdot}10^{-3}$	$-2.71{\cdot}10^{-4}$	$8.18{\cdot}10^{-5}$	0.8559
	16 nm	ST	$4.23{\cdot}10^{-3}$	$-5.43{\cdot}10^{-4}$	$2.86{\cdot}10^{-4}$	0.8307
		S3	$3.47{\cdot}10^{-3}$	$-5.90{\cdot}10^{-4}$	$2.39{\cdot}10^{-4}$	0.8245
		ST-S	$2.83{\cdot}10^{-3}$	$-4.34{\cdot}10^{-4}$	$1.47{\cdot}10^{-4}$	0.8923
3D		ST	$4.74{\cdot}10^{-3}$	$-6.61{\cdot}10^{-4}$	$1.22{\cdot}10^{-4}$	0.9715

Table 3.2: Fitting for a relationship in the form of (3.5). Columns from left to right: dimension of the simulation, cell width, shape function and smoothing used, parameters of the fitted function A and B, root mean squared error and r-squared statistics.

$$C^t_{cool} = \frac{E_k(t) - E_k(t + 1\text{ fs})}{E_k(t)}. \tag{3.4}$$

This way we can obtain cooling coefficients for various times during a simulation. We will focus on the last 30 fs of the simulations (see Figs 3.2 - 3.4 in subsection 3.2.1) in order to isolate the numerical effects from the heating or cooling that occurs due to physical mechanisms. For each performed simulation we will therefore use coefficients $C^{129\text{ fs}}_{cool}, C^{130\text{ fs}}_{cool}, ..., C^{159\text{ fs}}_{cool}$ to determine their dependence on the macroparticle number. We will again attempt to fit a relationship in the form of (3.5) on the cooling coefficients for different variants of simulations.

$$C^t_{cool}(N_{ppc}) = \frac{A}{\sqrt{N_{ppc}}} + B \tag{3.5}$$

Table 3.2 shows the parameters obtained using the Matlab curve fitting tool with the same goodness of fit statistic as the table in the previous subsection. Due to having multiple data points representing the cooling coefficients at various times during the simulations, the goodness of fit statistics are not as clear as in the previous subsection. However, multiple data points ensure better representation of the relationships we are trying to fit. Therefore we can conclude that the relation (3.5) is relevant to the description of numerical effects in PIC simulation.

The parameter A represents how strongly the cooling coefficient depends on the macroparticle number. As in the previous subsection, this parameter is highest for 2D simulations using the triangular shape function without current smoothing. However this time, the current smoothing seems to be more effective in reducing the numerical cooling in the simulation than use of the smooth shape function.

The negative values of the parameter B in all the fitted simulations represent other mechanisms, possibly both numerical and physical, that cause heating of the particles. The nonphysical heating of particles with respect to the size of the cells has been studied in the previous thesis [1]. This numerical effect presents itself more strongly in the simulations with bigger cells. Use of higher order shape function reduces this effect as should the current smoothing. The values presented in table 3.2 are consistent with these properties of nonphysical numerical heating.

3.2.4 Simulated quantities

In this subsection we will demonstrate how the representation of some quantities in simulations changes with regards to the macroparticle number, choice of shape function, the current smoothing algorithm and the cell size.

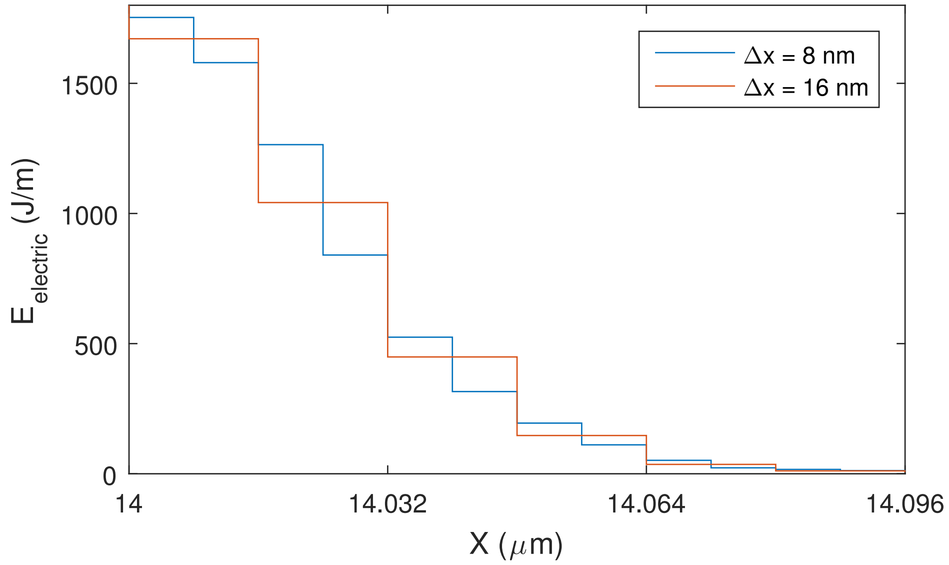

Figure 3.8: Energy of the electric fields in the first few cells of the target area at time 55 fs in 2D simulations using the triangular shape function and 60 particles per cell. The energy has been summed for all the cells with the same position on the x-axis.

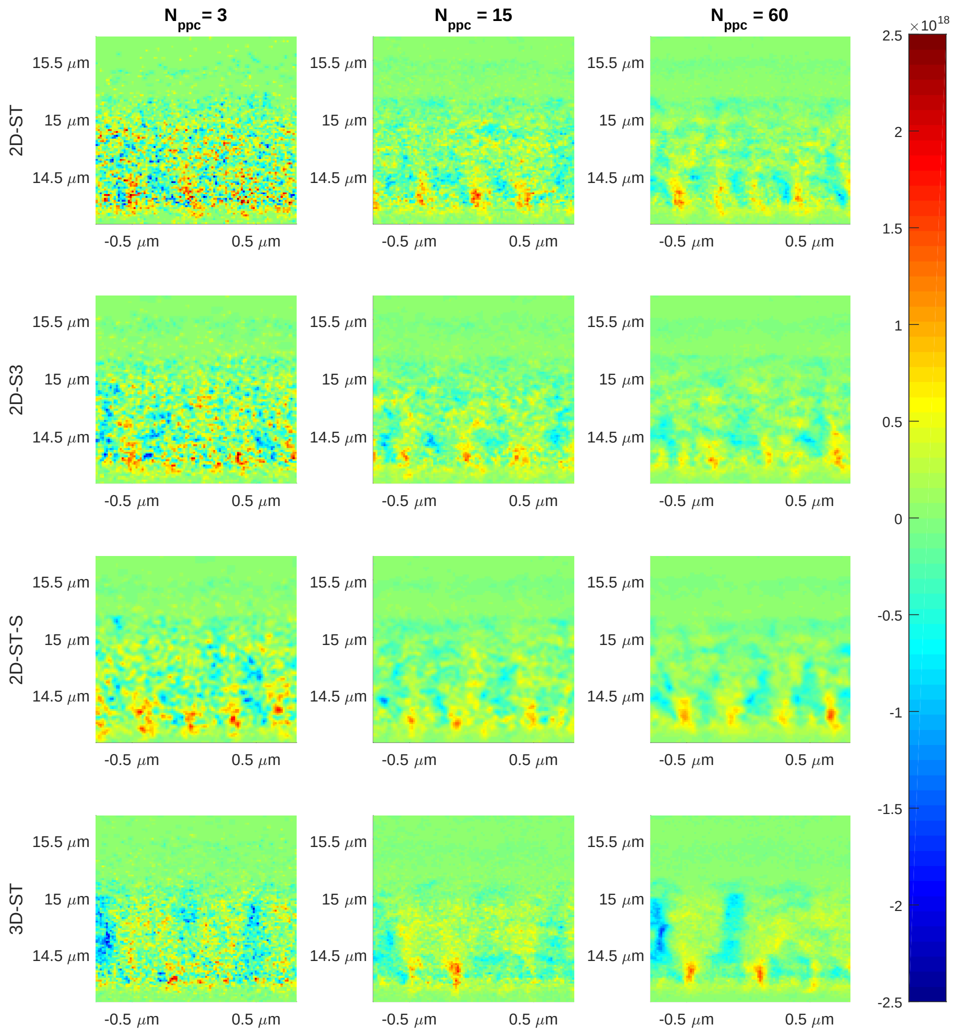

Figure 3.9: Magnitude of the x-component of the current density in simulations at time instant 70 fs for 2D and 3D simulations for various macroparticle numbers with cells 16 nm wide. The data from the 3D simulation has been taken as a cut at z = 0.

Figure 3.8 demonstrates how the field quantities are represented differently in the simulations with various cell sizes. The figure shows the energy of the electric field in the area where the laser pulse interacts with the target. This area is therefore crucial to the results of the whole simulation. The energy of the electric field is calculated using the formula (3.6), where ϵ_0 is the permitivity of vacuum. Note that in 2D simulations, the quantities in cells are treated as if the cells were 1 m wide in the remaining dimension, thus the energy is calculated as joules per meter.

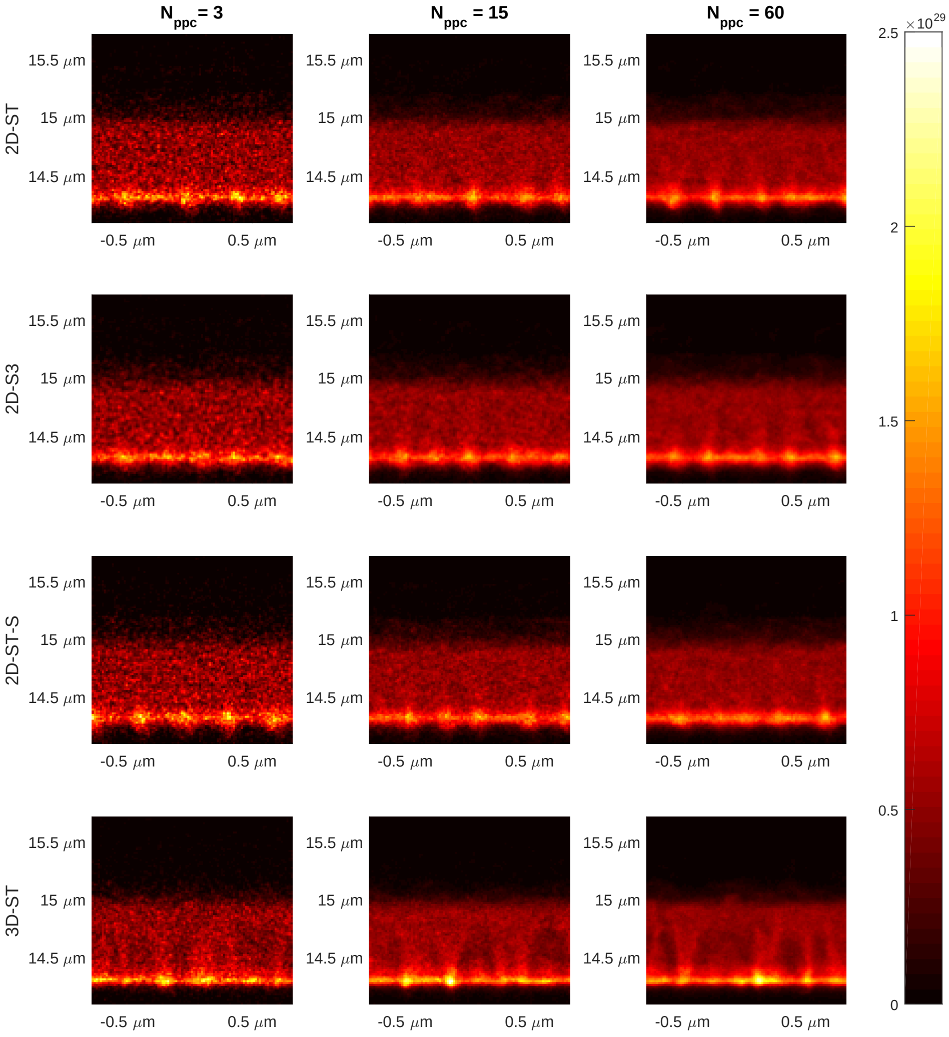

Figure 3.10: The electron number density in simulations at time instant 70 fs for 2D and 3D simulations for various macroparticle numbers with cells 16 nm wide. The data from 3D simulation have been taken as a cut at z = 0.

$$E_{electric} = \frac{1}{2}\epsilon_0 \boldsymbol{E}^2 \tag{3.6}$$

The energies of the fields in the simulations with different cell sizes appear to represent the same function with different sampling. Worse sampling introduces inaccuracies into the code, which is amplified by the fact that the ionized target is represented by fewer and larger macroparticles in the simulations with bigger cell size. We will discuss the possible implications these inaccuracies have on the absorption coefficient in the last section of this chapter.

Figure 3.9 shows how the representation of the currents differs with regards to the macroparticle number, the shape function and smoothing. It is important to note that in this case we are using data from simulations with the cell size of 16 nm, which in the 2D case, are deemed inaccurate. However, we can demonstrate the general differences in the simulations with various settings.

In the column $N_{ppc} = 60$ (showing the results from simulations with 60 macroparticles per cell), we can observe that there are areas in the form of filaments that have positive x-component of the current density. For the simulations with $N_{ppc} = 3$, these areas are almost unrecognizable and do not hold any shape, except for the simulation with the current smoothing algorithm enabled. The current density in all other simulations with the low macroparticle number is not smooth and resembles noise and not anything we would expect a physical quantity to look like. The ability of the current smoothing algorithm to eliminate this undesired effect is consistent with the observations from the previous two subsections, that the smoothing is better at reducing numerical effects in simulations with cell size of 16 nm.

The electron number density data from the same simulations shown in Figure 3.10 exhibits the same trends. It is also clear that in the case $N_{ppc} = 3$, many electron macroparticles are located at least few cells away from any other electron macroparticles. This is the reason for the huge fluctuations in the currents, which will be discussed further in the last section of this chapter.

Comparison of the 2D and 3D simulations, especially for higher macroparticle numbers, shows us that the filament structure can be observed in 3D simulations than more clearly than in 2D simulations. This can be attributed to the fact that this phenomenon cannot be fully represented in a 2D simulation.

3.3 Computational costs

When performing PIC simulations it is always important to consider the computational costs. Care must be taken to ensure that the simulations are run with optimal parameters so that the simulation is safely accurate and at the same time uses the smallest amount computational resources.

In order to determine the dependence of the simulation cost on the macroparticle number and other parameters, multiple simulations runs were performed in both 2D and 3D version of the code. The setup of the simulations was the same as described in section 3.1 with few exceptions. The simulations represented 100 fs of laser-plasma interaction in 2D version of the code and 1 fs in 3D version of the code, both produced no output and the particles have not been frozen for any part of the simulation. In order to run these simulations we reserved a whole node of the computer cluster. Reserving all the resources in a computer cluster node reduces any possible external effects on the duration of the simulations.

Table 3.3 shows how long it took to perform the 2D simulations on Metacentrum node Ida 5 (2x 10-core Intel E5-2650v3 3GHz) in seconds. A 3D simulation was also performed within the

		N_{ppc}					
Δx	SF	3	6	10	15	30	60
8nm	ST	18.89	22.47	27.70	36.38	61.43	80.79
	S3	21.64	30.10	41.48	51.81	93.84	177.36
	ST-S	19.11	24.39	33.83	39.54	63.73	111.09
16nm	ST	1.88	2.62	3.49	4.78	8.08	11.44
	S3	2.60	3.15	4.16	7.01	11.60	24.07
	ST-S	1.96	2.50	3.73	4.89	8.06	15.60

Table 3.3: Simulation times of 2D runs performed on node Ida 5 in seconds. First two columns indicate shape function used and width of cells, the next five columns represent simulation time for runs with various macroparticle numbers.

same job as the 2D simulations to compare the computational demands. The 3D simulation had 30 particles per cell performed, 16 nm wide cells and used the triangular shape function. It only ran for 1 fs of the simulated time, which took 294.98 seconds. Considering that the 2D simulations simulated 100 fs of laser-plasma interaction, we can assume that a full 100 fs 3D simulation would take roughly 3650 times longer than its 2D equivalent.

Results in table 3.3 are consistent with the results from the previous thesis [1]. Specifically, decrease in the size of cells presents itself in cubic increase of computation time (e.g. halving Δx increases the simulation time 8-fold).

		N_{ppc}					
Δx	SF	3	6	10	15	30	60
16nm	ST	71.97	110.28	206.91	291.19	348.62	579.32
	S3	149.14	244.85	350.36	469.63	817.05	1555.53
	ST-S	86.36	126.32	213.83	244.69	400.88	624.33

Table 3.4: Simulation times of 3D runs performed on node Krux 1 in seconds. First two columns indicate shape function used and the size of cells, the next five columns represent simulation time for runs with various macroparticle numbers.

Table 3.4 shows how long it took to perform the 3D simulations representing 1 fs of laser-plasma interaction on Metacentrum node Krux 1 (4x 16-core AMD Opteron 6376 2.3GHz) in seconds.

		N_{ppc}				
Δx	SF	3	6	10	15	30
16nm	ST	0:49:28	0:58:45	1:10:47	1:26:23	2:08:32

Table 3.5: Duration of full 160 fs 3D simulations with diagnostics output performed on cluster ARIS. First two columns indicate shape function used and the size of cellsthe next four columns represent simulation times in format h:mm:ss for various macroparticle numbers.

Table 3.5 shows how long it took for the full 160 fs 3D simulations with diagnostic output enabled to be calculated on 50 nodes of the ARIS cluster. Each node has 2x 10-core Intel Xeon E5-2680v2 2.8 GHz and 64 GB RAM, thus 500 CPUs in total were used for the simulations. Note that in these simulations the particles were frozen for the initial 40 fs. The simulation with $N_{ppc} = 60$ has been performed independently of the others and its duration has not been measured.

As expected, in both 2D and 3D, the computation time increases linearly with the macroparticle number. Part of the computation time is spent on calculations that relate to the particles. Rest of the simulation is spent on calculations of quantities on the grid. Use of the higher order shape function increases the former, while the use of current smoothing increases the latter. This means that the amount of time required for smoothing of the currents does not increase with the macroparticle number, but with the amount of gridpoints in the simulation.

3.4 Discussion

General formula (2.31) introduced in subsection 2.4.5 proved to be relevant for the relation between values of some directly observable and quantifiable coefficients in PIC simulations and the macroparticle number. However, the relation was derived only for the errors that occur during one time step of interpolation of the current densities. One could expect that the errors would cancel out to a certain degree and the coefficients for various macroparticle simulations would therefore be randomly distributed around a certain value, which would be the same regardless of the macroparticle number. It is therefore necessary to further discuss and describe the mechanism which results in the accumulation of errors in a manner that can be observed in the performed simulations.

Firstly, let us describe a mechanism by which the numerical cooling occurs and why this effect amplifies with a smaller macroparticle number. This effect occurs due to fluctuations in the charge density. Such fluctuations result in strong nonphysical currents, which in turn result in strong fields. These fields partially correct the fluctuation by slowing down the particles, while some energy of the field is radiated away into neighboring cells which may not contain any particles and thus the energy of the field can not be transferred back to the particles.

As an example of this effect, let us suppose expansion of plasma into vacuum. If the distribution function is sampled well enough by the macroparticles then no strong nonphysical fields emerge. But if there are only few macroparticles at the edge of the plasma boundary, a situation may occur in which only a single macroparticle moves in the direction of the expansion, which results in an error in the representation of the distribution function, so that the distribution function has a local minimum between the position of the macroparticle and the rest of the plasma. This situation leads to the creation of a strong field which pulls the macroparticle back into the plasma. However, even after the expanded particle is pulled back, portion of the field energy will remain in the cells at the plasma boundary and will get propagated via the Maxwell's equations in form of waves in all directions, including away from the macroparticles and thus some of the energy will never be transferred from the fields onto the particles.

The cooling effect is weaker in simulations with smaller cells due to the fact that each macroparticle has smaller charge and in total, there are more macroparticles in the simulation. Smaller cell size also results in a smaller time step, which means that the particles have better chance of correcting the numerical fluctuations. Furthermore some numerical effects that reduce accuracy due to finite grid may be weaker in the simulations with smaller cells.

Use of the higher order shape function reduces fluctuations due to the fact that macroparticles representation is smoother. The current smoothing algorithm works on a similar principle. However, the fluctuations are smoothed after they have been transferred onto the currents in the interpolation step. Results from the simulations presented in the previous section indicate that the current smoothing algorithm reduces the effect of particle cooling more successfully. However, it is possible that the smoothing introduces some other inaccuracies into the simulation that may alter some physical mechanisms, which can be critical to physical interpretation of the simulation. Exploration of such potential effects of the current smoothing algorithm is outside of the scope of this thesis.

Regarding the results of the simulations, it is important to address the issue of different absorption coefficients calculated in simulations with different cell sizes. This may be the result of undersampling of the laser pulse field interacting with the target. The difference in the way the pulse is represented in a thin surface layer of the target has been shown in Fig. 3.8. Undersampling enables the particles further inside the target to be subject to stronger fields than they would be if the sampling of the pulse was better. Larger cell size also results in the a longer timestep, meaning that the particles are under the effects of already stronger field for longer than they would be in the case of simulation with smaller cell size. The effect of the fields on the particles is therefore amplified due to inaccuracies caused by both the finite grid and time step. This potentially results in higher rates of absorption in the simulations with higher cell sizes.

The difference between the results of the 2D and 3D simulations in absorption can be explained by the inability of 2D simulations to emulate some of the mechanisms that occur during the interaction of the pulse with the plasma. The undersampling of the field and the fluctuations in the distribution function in general is also less of an issue in the 3D simulations, since there are more macroparticles in general and the representations of various quantities in the simulation are thus smoother and can get corrected more easily.

Computational demands of the simulations increase linearly with the macroparticle number. The use of higher order shape function in order to suppress undesirable numerical effects is not efficient. Use of the smooth shape function increases the computational time significantly in 2D. This is due to the fact that during the interpolation steps, the macroparticles with the smooth shape function overlap 25 cells, while the macroparticles represented by the triangular shape function overlap only 9 cells. In 3D, the use of the smooth shape function is even more computationally demanding as any macroparticle overlaps 125 cells compared with 27 cells being overlapped by the triangular shape function. The higher order shape function also requires computationally more expensive operations.

The current smoothing algorithm increases the computational demands of simulations only slightly. This is due to the fact that the smoothing is performed on the grid and not on the particles. In most PIC simulations, the computational time required to perform the steps that are calculated on the grid is smaller than the time consumed by the steps in which the macroparticles have to be considered. This also means that increasing the macroparticle number in a simulation with the current smoothing algorithm will only result in the same increase of computational time as increasing the macroparticle number in a simulation with the same shape function, but no current smoothing algorithm.

These properties of the smooth shape function and the current smoothing algorithm make the current smoothing algorithm a better choice when reducing any undesirable numerical effects related to the macroparticle number. Increasing the macroparticle number while using the triangular shape function is often cheaper in the terms of computation time than using the higher order shape function. This holds especially for 3D simulations. The smooth shape function may find its use in 1D simulations, where the computational demands are not that much higher than demands of the simulations with the triangular shape function.

Decreasing the cell width increases the computational time of the simulation with the power of $D + 1$, where D is the dimension of the code. The increase is due to the fact that the amount of both the macroparticles and the gridpoints increases with the power of D and the time step is decreased proportionally to the decrease of the cell width, which is in turn inversely proportional to the total amount of the time steps that need to be calculated. This makes the choice of optimal cell size more crucial than the choice of the macroparticle number. However, especially in simulations with small cells, the macroparticle number can have significant impact on the total computational demands.

Conclusion

The basics of the physics of interaction of ultra-intense laser pulses with ionized targets have been presented in the first chapter of this thesis. Understanding of the underlying physical processes leads to more valuable interpretation of the results of plasma simulations and detection and examination of the inaccuracies that arise due to numerical effects.

The second chapter provided a comprehensive overview of the particle-in-cell method. The conditions for numerical stability and accuracy introduced in this chapter are most crucial for the topic of this thesis. The derivation of the general dependence of the errors in the simulation on the macroparticle number has been presented. The derived relation (2.31) predicts that the general rate of inaccuracy in PIC simulations is inversely proportional to the square root of the macroparticle number.

Multiple 2D and 3D simulations were performed to examine the numerical effects and determine the validity of the proposed relation for the accuracy of the simulation with regards to the macroparticle number. The results of the simulations are consistent with the proposed relation. Thus, we can conclude that the relation can be considered relevant for further use when dealing with modern particle-in-cell codes.

The impact of the choice of the shape function and potential use of the current smoothing algorithm on the undesired numerical effects has also been examined based on the performed simulations. The current smoothing algorithm proved to be able to suppress the inaccuracies in simulations with low macroparticle number to equal or greater extent than the choice of the higher order shape function.

Mechanism of the numerical cooling of the particles has been discussed with regards to the amount of particles in the simulation. Numerical effects that can be attributed to the finite grid have also been discussed. Even though this was not the focus of this thesis, it was necessary as the various numerical effects are interconnected and affect each other.

Additional simulations were performed to determine the computational demands of the EPOCH code with regards to the macroparticle number and other parameters. The results confirmed the expectation that the computational time increases linearly with the macroparticle number. Optimal choice of the macroparticle number can therefore lead to significant reduction in the total time of computation, even though the optimization of the cell can lead to even larger reduction of computational time.

Interesting result of the performed simulations was that the current smoothing algorithm increases the total computational costs only slightly. This result makes it a significantly better choice over the use of higher order shape function when reducing any undesired numerical effects.

Results presented in this thesis are interesting for further study of the numerical aspects of the particle-in-cell method, as well as any practical use of the method, especially when using the code EPOCH to perform simulations in laser plasma physics.

Bibliography

[1] Kocur V. Numerické aspekty particle-in-cell simulací. Bachelor's thesis, 2015.

[2] Kocur V. Numerická přesnost a časová náročnost vícedimenzionálních particle-in-cell simulací ve fyzice laserového plazmatu. Research assignment, 2016.

[3] Birdsall C K and Langdon A B. *Plasma Physics via Computer Simulation*. Adam Hilger, 1999.

[4] Hockney R W and Eastwood J W. *Computer Simulation Using Particles*. Taylor & Francis, 1998.

[5] Ruhl H. *Classical Particle Simulations with the PSC code*. Academia.

[6] Arber T D, Bennett K, Brady C S, Lawrence-Douglas A, Ramsay M G, Sircombe N J, Gillies P, Evans R G, Schmitz H, Bell A R, and Ridgers C P. Contemporary particle-in-cell approach to laser-plasma modelling. *Plasma Physics and Controlled Fusion*, 57(11):1–26, November 2015.

[7] Chen F. *Úvod do fyziky plazmatu*. Ruhr-Universität Bochum, 1984.

[8] Štoll I. *Elektřina a magnetismus*. FJFI, ČVUT, 2003.

[9] Pšikal J and Klimo O. Základy fyziky laserového plazmatu. Lecture, 2014.

[10] Limpouch J. Základy fyziky plazmatu. Lecture, 2015.

[11] Kulhánek P. *Teorie plazmatu - studijní text pro FJFI ČVUT*. FJFI ČVUT, 2009-2015.

[12] Gibbon P. *Short pulse laser interactions with matter*. World Scientific Publishing Company, 2004.

[13] Pfalzner S and Gibbon P. A 3d hierarchical tree code for dense plasma simulation. *Computer physics communications*, 79(1):24–38, 1994.

[14] Christlieb A J, Krasny R, Verboncoeur J P, Emhoff J W, and Boyd I D. Grid-free plasma simulation techniques. *IEEE Transactions on Plasma Science*, 34(2):149–165, 2006.

[15] Perry M D. Multilayer dielectric gratings. *Science and Technology Review*, pages 24–33, 1995.

[16] Strickland D and Mourou G. Compression of amplified chirped optical pulses. *Optics communications*, 56(3):219–221, 1985.

[17] Backus S, Durfee C G, Murnane M M, and Kapteyn H C. High power ultrafast lasers. *Review of scientific instruments*, 69(3):1207–1223, 1998.

[18] Paschotta R. Optical parametric amplifiers. In *RP Photonics Encyclopedia*. 2016. Retrieved: 2.8.2016.

[19] Dubietis A, Jonušauskas G, and Piskarskas A. Powerful femtosecond pulse generation by chirped and stretched pulse parametric amplification in bbo crystal. *Optics Communications*, 88(4-6):437–440, 1992.

[20] Jeong TM and Lee J. Femtosecond petawatt laser. *Annalen der Physik*, 526(3-4):157–172, 2014.

[21] Bahk S-W, Rousseau P, Planchon T A, Chvykov V, Kalintchenko G, Maksimchuk A, Mourou G A, and Yanovsky V. Generation and characterization of the highest laser intensities (10 22 w/cm 2). *Optics letters*, 29(24):2837–2839, 2004.

[22] Pšikal J. *Ion Acceleration in Small-size Targets by Ultra-intense Short Laser Pulses (Simulation and Theory)*. PhD thesis, 2009.

[23] Klimo O. *Simulations of Ultrashort-Pulse Laser Solid-Target Interactions*. PhD thesis, 2007.

[24] Gamalii EG and Tikhonchuk VT. Effect of intense ultrashort light pulses on a substance. *JETP Lett*, 48(8), 1988.

[25] Yang T-Y B, Kruer W L, More R M, and Langdon A B. Absorption of laser light in overdense plasmas by sheath inverse bremsstrahlung. *Physics of Plasmas (1994-present)*, 2(8):3146–3154, 1995.

[26] Yang T-Y B, Kruer W L, Langdon A B, and Johnston T W. Mechanisms for collisionless absorption of light waves obliquely incident on overdense plasmas with steep density gradients. *Physics of Plasmas (1994-present)*, 3(7):2702–2709, 1996.

[27] Eliezer S. *The interaction of high-power lasers with plasmas*. CRC Press, 2002.

[28] Brunel F. Not-so-resonant, resonant absorption. *Physical Review Letters*, 59(1):52, 1987.

[29] Serbanescu C, Fourmaux S, Kieffer J-C, Kincaid R, and Krol A. K-alpha x-ray source using high energy and high repetition rate laser system for phase contrast imaging. In *SPIE Optical Engineering+ Applications*, page 745115. International Society for Optics and Photonics, 2009.

[30] Hares J D, Kilkenny J D, Key M H, and Lunney J G. Measurement of fast-electron energy spectra and preheating in laser-irradiated targets. *Physical Review Letters*, 42(18):1216, 1979.

[31] Martinolli E, Koenig M, Baton S D, Santos J J, Amiranoff F, Batani D, Perelli-Cippo E, Scianitti F, Gremillet L, Mélizzi R, et al. Fast-electron transport and heating of solid targets in high-intensity laser interactions measured by k α fluorescence. *Physical Review E*, 73(4):046402, 2006.

[32] Macchi A, Cattani F, Liseykina T V, and Cornolti F. Laser acceleration of ion bunches at the front surface of overdense plasmas. *Physical review letters*, 94(16):165003, 2005.

[33] Wagner F, Deppert O, Brabetz C, Fiala P, Kleinschmidt A, Poth P, Schanz V A, Tebartz A, Zielbauer B, Roth M, et al. Maximum proton energy above 85 mev from the relativistic interaction of laser pulses with micrometer thick ch 2 targets. *Physical review letters*, 116(20):205002, 2016.

[34] Esarey E, Schroeder C B, and Leemans W P. Physics of laser-driven plasma-based electron accelerators. *Reviews of Modern Physics*, 81(3):1229, 2009.

[35] Pukhov A. Strong field interaction of laser radiation. *Reports on progress in Physics*, 66(1):47, 2002.

[36] Sprangle P, Esarey E, Krall J, and Joyce G. Propagation and guiding of intense laser pulses in plasmas. *Physical review letters*, 69(15):2200, 1992.

[37] Kim H T, Pae K H, Cha H J, Kim I J, Yu T J, Sung J H, Lee S K, Jeong T M, and Lee J. Enhancement of electron energy to the multi-gev regime by a dual-stage laser-wakefield accelerator pumped by petawatt laser pulses. *Physical review letters*, 111(16):165002, 2013.

[38] Klimo O. Metody počítačové fyziky. Lecture, 2015.

[39] Cormier-Michel E, Shadwick BA, Geddes CGR, Esarey E, Schroeder CB, and Leemans WP. Unphysical kinetic effects in particle-in-cell modeling of laser wakefield accelerators. *Physical Review E*, 78(1):016404, 2008.

[40] Bennett K. *Developers Manual for the EPOCH PIC codes*. University of Warwick, 2013.

[41] Villasenor J and Buneman O. Rigorous charge conservation for electromagnetic field solvers. *Computer Physics Communications*, 69:306–316, 1992.

[42] Yee K. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *Antennas and Propagation, IEEE Transactions on*, 14:302–307, 1966.

[43] Limpouch J. Numerické metody. Lecture, 2014.

[44] Courant R, Friedrichs K, and Lewy H. On the partial difference equations of mathematical physics. *IBM Journal of Research and Development*, 11:215–234, 1967[1928].

[45] Shadwick B A, Schroeder C B, Leemans W, and Esarey E. Physical fidelity in particle-in-cell modeling of small debye-length plasmas. In *AIP Conference Proceedings*, volume 1086, pages 321–327. AIP, 2009.

[46] Dashboard | GitLab. `https://cfsa-pmw.warwick.ac.uk/`. Retrieved: 4.29.2017.

[47] O MetaCentru VO. `https://metavo.metacentrum.cz/cs/about/index.html`. Retrieved: 30.3.2017.

[48] Hardware overview - aris documentation. `http://doc.aris.grnet.gr/hardware/`. Retrieved: 31.3.2017.

[49] Matlab curve fitting toolbox. `https://www.mathworks.com/help/curvefit/index.html`. Retrieved: 17.4.2017.

[50] Evaluating goodness of fit. `https://www.mathworks.com/help/curvefit/evaluating-goodness-of-fit.html`. Retrieved: 17.4.2017.