Alma Mater Studiorum
Università degli Studi di Bologna

Dottorato di Ricerca in Fisica
Ciclo XXVII

Numerical and Analytical Methods for Laser-Plasma Acceleration Physics

Settore Concorsuale di Afferenza: 02/B3

Settore Scientifico Disciplinare: FIS/07

Coordinatore: Prof. Fabio Ortolani

Presentata da:

Francesco Rossi

Relatore:
Prof. Giorgio Turchetti

Tutor:
Prof. Armando Bazzani

Esame Finale Anno Accademico 2013-2014

Contents

Introduction

Particle accelerators have had a profound impact on many fundamental science discoveries and they are the basis of important technologies such as synchrotrons and free electron lasers. Conventional schemes are limited by the electrical breakdown limit of the radiofrequency (RF) cavities they use for generating the electric fields that accelerate charged particles. In fact, in those cavities, the accelerating field is bound to $\sim 100MV/m$ and this has implied that present high energy accelerators are tens of kilometers long and cost billions of dollars.

Over the last decade, the laser wakefield acceleration technique (LWFA) started to emerge as a breakthrough electron acceleration technology and as a possible alternative for building electron accelerators that could overcome the limitations of the conventional, RF-based, accelerators.

To accelerate particles, the resonant accelerating structures in the RF-based schemes have been replaced in LWFA by the electric field generated by a plasma wake (the wakefield), driven by a relativistically intense, short, laser pulse. The plasma wave generated by the laser driver is the result of the gradient in laser field energy density providing a force (i.e., the ponderomotive force) that creates a space charge separation between the (underdense) plasma electrons and the neutralizing ions. The wakefield can exceed several hundreds gigavolts per meter in peak amplitude and propagates through the plasma at relativistic velocities, following the laser pulse with a phase velocity of the order of the group velocity of the laser driver. This ability to sustain extremely large acceleration gradients enables compact accelerating schemes.

The LWFA technique was proposed 35 years ago by Dawson and Tajima [23] and, in the last decade, the rapid progress of the laser technology boosted the development of LWFA accelerators. Current laser-plasma accelerator (LPA) experiments, requiring ultra-short (tens of femtoseconds) and powerful pulses ($10\,TW - 1PW$), operate at relativistic intensities $I_0 \gtrsim 10^{18}W/cm^2$.

Using a $9-cm$-long capillary discharge waveguide (with plasma density $\sim 7{\cdot}10^{17}cm^{-3}$)

to drive a sub-petawatt laser pulse ($0.3PW$), the group led by Wim Leemans at Lawrence Berkeley National Laboratory has recently demonstrated in experiments a scheme that produces high quality electron beams (6% rms energy spread, $0.3mrad$ rms divergence, $6pC$ charge) with record-breaking energies, up $4.2GeV$ [5].

Laser plasma accelerators are interesting candidates for applications to future high energy colliders [11] and radiation sources [12, 13, 14].

The acceleration in a LPA is limited by dephasing, the distance at which the accelerated beam outruns the accelerating part of the wakefield, and by the evolution of the laser pulse. During its propagation in the plasma, the pulse undergoes both transverse evolution (diffraction/relativistic self-focusing/plasma wave guiding) and longitudinal evolution (self-steepening, energy depletion and redshifting). The evolution of the pulse affects the accelerating properties of the wake, and hence the dynamics of the accelerated bunch.

In an LPA, charged particles need to be injected at the correct phase of the wakefield, as for any accelerating structure. In the "bubble" regime, in which the ponderomotive force of the intense laser pulse transversally expels ambient electrons and forms of a trailing electron ellipsoidal cavity moving at relativistic velocity (a bubble wake), it has been observed, in experiments and in Particle-in-Cell simulations that, *in some cases,* electrons from the background plasma itself can be (self-)injected and accelerated in the wake. Denoting with L_0, λ_0 the rms length and wavelength of the laser pulse, respectively, the bubble regime can be accessed if the pulse is quasi-resonant $k_pL \sim 1$, (where $\omega_p = k_pc = \sqrt{4\pi n_0 e^2/m_e}$ is the plasma frequency for a plasma of density n_0 and c is the speed of light in vacuum) and if the peak normalized vector potential of the laser satisfies $a_0 \gtrsim 2$ ($a_0 \simeq 8.5 \cdot 10^{-10}(I_0[W/cm^2])^{1/2}\lambda_0[\mu m]$).

The self-injection mechanism is probably the simplest injection technique to access experimentally, but several other injection techniques have been developed (including colliding pulses [15, 16], tailored density profiles [17, 18, 19] and ionization-induced injection [20, 21]).

Understanding the self-injection mechanism and the propagation of short and intense laser pulses in a underdense plasma are therefore topics of fundamental importance in the field of laser-plasma accelerators.

In Chapter 1, the laser evolution and plasma wave excitation by a weakly relativistically intense, short-pulse laser propagating in a preformed parabolic plasma channel is discussed, including the effects of pulse steepening and energy depletion. Analytical expressions for the laser energy depletion, the pulse self-steepening rate, the laser intensity centroid velocity, and the phase velocity of the plasma wave are derived in 3D, as in our upcoming contribution [22], and in the weakly relativistic intensity regime $a_0 < 1$.

Due to the high nonlinearities, to study higher intensity regimes in 3D, numerical simulations are generally required. In Chapter 2, reviewing the recent results in Ref. [8], the nonlinear bubble regime is systematically studied by means of particle-in-cell simulations, run with the ponderomotive PIC code INF&RNO [26, 27, 28] under controlled conditions. The bubble wake properties and the importance of the bubble wake velocity in the self-injection process are investigated.

Due to the high nonlinearity and complexity of the phenomena involved, numerical simulations are fundamental tools for studying of laser plasma interaction, for modeling LPA experiments, for designing them and for developing new theories (as in Chapter 2). Particle-in-Cell (PIC) codes (Ref. [39], and Chapter 3 for a general introduction) provide an accurate kinetic description of plasmas and are very established tools in the LPA community.

The most complete ("full") physical model for studying laser-plasma interactions is the Vlasov equation, providing a 6D phase space kinetic description of the plasma, coupled with the Lorentz force, the relativistic equations of motion and Maxwell equations for Electrodynamics. PIC codes discretize the Vlasov equation sampling the phase space with spatially-shaped computational particles.

In this numerical view, the smallest scale to resolve is the laser pulse wavelength, of the order of the micron (μm). In LWFA, the longest physical scale of interest is the acceleration length, which can range in current designs and experiments from the order of the millimeter (mm), up to the order of the meter (m) (lower densities, longer propagation distances resulting in higher energies). Due to this extremely large scale separation (the ration can be greater than 10^6), 3D simulations of laser-plasma acceleration are extremely demanding in terms of computational power, even with modern top supercomputers.

In part II, we review the numerical methods and the numerical and computational optimizations that allow to accurately model the 3D Physics of laser plasma accelerators with present supercomputing architectures.

In particular, in Chapter 4 the challenges and benefits of porting the PIC algorithms to the massively parallel Graphics-Processing-Unit (GPU) architecture are discussed. Exploiting massive parallelism present in applications, GPUs deliver exceptional performance in term of computational throughput and memory bandwidth, but the PIC core algorithms need to be redesigned for satisfying the constraints imposed by the intrinsic parallelism of the architecture. The code *jasmine,* a relativistic, multi-GPU efficient PIC code, implementing a "full" 3D model, is presented. *jasmine* is part of the efforts made by the computational laser-plasma Physics group at the University of Bologna, that developed novel high order schemes for PICs in the code framework *ALaDyn* [3].

The code *jasmine* has been used to model recent LWFA experiments run with the $220TW$ INFN-LNF FLAME laser system (Frascati Laser for Acceleration and Multidisciplinary Experiments) and with the $10TW$ laser system installed at the Intense Laser Irradiation Laboratory (ILIL) of the INO of the CNR in Pisa, with the goal of studying optically driven electron beam sources for Thomson scattering [41]. Some results of the numerical modeling campaigns are presented in Sections 7.1 and 7.2.

A key to success for multi-GeV acceleration LPAs is the realization of a much longer interaction lengths (therefore scaling to lower plasma densities). Due to the scale separation (and numerical dispersion issues), scaling to interaction lengths greater than the centimeter with a full PIC model is prohibitive with present architectures (GPUs included). Reduced models [38, 36, 26] and running the simulation in an optimal Lorenz boosted frame [30, 33, 32, 31] have been proposed to overcome this limitations and allow for the simulations of multi-GeV LPA stages.

In Chapter 5, the reduced-model, cylindrical (r-z), code INF&RNO (INtegrated Fluid & paRticle simulatioN cOde) [26, 27, 28], developed at Lawrence Berkeley National Laboratory, is presented. INF&RNO uses the envelope approximation for describing the laser pulse and the ponderomotive force approximation for the laser-plasma interaction. In the envelope approximation, the driver characteristic length (of the order of the plasma length in most LWFA) is, in principle, the smallest scale to resolve and, being much longer than the laser wavelength, the min/max scale separation is less dramatic than in a full PIC model. Nevertheless, numerical simulations using reduced models are still computationally challenging, requiring up to tens of thousands CPU core hours. Efficient parallelization is therefore still necessary. The advanced numerical schemes in INF&RNO require the use of advanced parallelization methods, and the efforts for achieving scalability and efficient parallelization on different architectures are described in Chapter 5.

An even stronger model reduction is the quasi-static approximation (QSA) [45, 46] that, separating the timescales of the driver evolution and of the electrons in the wake, can be successfully applied for developing powerful numerical codes [47, 49], allowing to accurately simulate laser evolution and wakefield generation very efficiently. In Chapter 6, the implementation of a quasi-static module in the INF&RNO framework is discussed. A load-balanced, pipelining-based parallelization technique is presented and a 0.5 meters long acceleration stage, accelerating electron bunches up to $9.6GeV$ in the quasi-linear wakefield driven by a BELLA-class laser pulse [25] is presented as a code benchmark in section 6.5.

Please download the latest version of this thesis from:

`http://physycom.unibo.it/rossi/rossi_phd_thesis.pdf`

Acknowledgements

Let me thank the Giorgio Turchetti and Pasquale Londrillo, who started the computational plasma Physics for laser plasma accelerators activity at the University of Bologna and raised my collaborators and younger mentors. They created connections and collaborations with leading research institutions, such as CNR-INO-ILIL and Lawrence Berkeley National Laboratory, which I visited for three summers during my graduate studies. At LBNL, I wish to thank in particular Carlo Benedetti, for being my guide in the field.

This thesis is dedicated to Sofia, Emma and Rolando, who have been always next to me.

I acknowledge support by the University of Bologna, the INFN and by the Department of Energy, under the Office of Science contract No. DE-AC02-05CH11231. Computational resources at CINECA, INFN-CNAF, National Energy Research Scientific Computing Center (NERSC), and INFN-QuonG were used for the topics presented in this thesis.

Papers

- “Towards robust algorithms for current deposition and dynamic load-balancing in a GPU particle in cell code” F. Rossi, P. Londrillo, A. Sgattoni, S. Sinigardi, and G. Turchetti *AIP Conf. Proc. 1507, 184 (2012)* http://proceedings.aip.org/resource/2/apcpcs/1507/1/184_1

- “Numerical investigation of electron self-injection in the nonlinear bubble regime” Benedetti, C. and Schroeder, C. B. and Esarey, E. and Rossi, F. and Leemans, W. P., *Physics of Plasmas (1994-present), 20, 103108 (2013),* DOI:http://dx.doi.org/10.1063/1.4824811

- “High energy electrons from interaction with a structured gas-jet at FLAME”, G. Grittani, M. P. Anania, G. Gatti, D. Giulietti, M. Kando, M. Krus, L. Labate, T. Levato, P. Londrillo, F. Rossi, *Nuclear Instruments & Methods in Physics Research A (2013), Volume 740, 11 March 2014, Pages 257–265*

- “Acceleration with self-injection for an all-optical radiation source at LNF” L.A. Gizzi, M.P. Anania, G. Gatti, D. Giulietti, G. Grittani, M. Kando, M. Krus, L. Labate, T. Levato, Y. Oishig, F. Rossi *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms Volume 309, 15 August 2013, Pages 202–209*

- “Transport and energy selection of laser generated protons for postacceleration with a compact linac” Stefano Sinigardi, Giorgio Turchetti, Pasquale Londrillo, Francesco Rossi, Dario Giove and Carlo De Martinis, Marco Sumini *Phys. Rev. ST Accel. Beams, Volume 16, Issue 3* http://prst-ab.aps.org/abstract/PRSTAB/v16/i3/e031301

- “The LILIA experiment: Energy selection and post-acceleration of laser generated protons” G. Turchetti, S. Sinigardi, P. Londrillo, F. Rossi, M. Sumini, D. Giove,

and C. De Martinis *AIP Conf. Proc. 1507, 820 (2012)*
http://proceedings.aip.org/resource/2/apcpcs/1507/1/184_1

- "IRIDE: Interdisciplinary research infrastructure based on dual electron linacs and lasers" M. Ferrario, . . . , F. Rossi, ... *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*
http://dx.doi.org/10.1016/j.nima.2013.11.040

- "High quality proton beams from hybrid integrated laser-driven ion acceleration systems" Stefano Sinigardi, Giorgio Turchetti, Francesco Rossi, Pasquale Londrillo, Dario Giove, Carlo De Martinis, Paul R. Bolton. *Nuclear Instruments and Methods in Physics Research A 740 (2014) 99–104*

- "High energy electrons from interaction with a 10 mm gas-jet at FLAME" G. M. Grittani, M. P. Anania, G. Gatti, D. Giulietti,, M. Kando, M. Krus, L. Labate, T. Levatof, Y. Oishi, F. Rossi and L. A. Gizzi *Proc. of SPIE Vol. 8779 87791B-1*

- "Laser Pulse Propagation and Phase Velocity of Laser-driven Plasma Waves in the Weakly-Relativistic Intensity Regime", C. Benedetti, F. Rossi, C. B. Schroeder, E. Esarey, and W. P. Leemans, *Submitted to Physical Review E*

- "Laser-plasma acceleration of electrons for radiobiology and radiation sources" L.A. Gizzi, L. Labate, F. Baffigi, F. Brandi, G.C. Bussolino, L. Fulgentini, P. Koester, D. Palla, F. Rossi, *Submitted to Nuclear Instruments and Methods B*

Part I.

Laser Wakefield Acceleration Theoretical Studies

In this part, the pulse propagation and the properties of the plasma wake are studied in the weakly-relativistic and bubble regimes.

In Chapter 1, we investigate and characterize the laser evolution and plasma wave excitation by a weakly relativistically intense ($a_0 < 1$), short-pulse laser propagating in a preformed parabolic plasma channel, including the effects of pulse steepening, frequency redshifting, and energy depletion. Wakefield properties and laser driver evolution are topics of fundamental importance in the field of laser-plasma interaction because they determine the dynamics of accelerated electrons in laser-driven plasma-based accelerators.

Starting from the envelope equation for the laser and the linearized quasi static plasma equations for the wakefield, analytical expressions for the quantities (initial energy depletion rate, intensity transport velocity and intensity peak velocity) governing the evolution of a short gaussian pulse propagating in a under-dense plasma channel have been derived in the weakly relativistic ($a_0 < 1$) regime. Analytical results have been validated numerically with accurate simulations performed with the 2D-cylindrical, ponderomotive code INFERNO [26, 27, 28].

In the same regime, an expression for the initial velocity of the excited plasma wave has been derived. The temporal evolution of the wake velocity has been numerically investigated and it has been shown that its oscillations temporally match the one of the laser intensity peak, rather than the laser intensity centroid. The transverse shape oscillations that an initially gaussian short pulse undergoes in a matched plasma channel significantly affect the wake velocity, leading to minimum values substantially lower that the laser linear group velocity or even laser 1D intensity transport velocity and, as the laser propagates, longitudinal pulse evolution (red-shifting and steepening) further decreases the phase velocity of the wake.

In Chapter 2, the nonlinear bubble regime, reached at higher pulse intensities $a_0 > 2$, is systematically studied by means of Particle-in-Cell simulations, run with the PIC code INF&RNO under controlled conditions.

The scaling of the bubble shape and size with the laser intensity has also been analyzed, showing significant deviation from a round bubble for $a_0 > 5$.

It has been proven that, even for a non-evolving driver (and consequently, non-evolving wake) propagating at a prescribed velocity, self-injection occurs, and the dependence of the threshold for self-injection on laser driver intensity and wake velocity has been explored.

Studying the injection threshold in a stationary bubble wake, the evolution of the pulse was decoupled from the self-injection mechanism. If the driver evolves (because of self-focusing, plasma wave guiding and/or self-steepening), the bubble wake velocity is no longer equal to the driver velocity, but is determined by the driver evolution.

The actual bubble phase velocity, significantly different from the laser driver group velocity, was found to be the relevant parameter to be considered for the self-injection physics. The evolution of wake velocity shows a complex behavior due to the interplay of different nonlinear effects, but, in our simulations, the minimum value of the wake velocity, measured at the center of the wake, can be expressed by the simple empirical expression $\gamma_0^{min} \simeq 2.4 \cdot \sqrt{\frac{k_0}{k_p}}$, and this value is independent of a_0.

1. Laser Propagation and Phase Velocity of Plasma Waves in the Weakly Relativistic Regime

The laser pulse depletion rate, propagation velocity and the phase velocity of plasma waves are of fundamental importance to many areas of Plasma Physics. For example, energy depletion rate and wake phase velocity determine the dynamics of the accelerated electrons in laser plasma accelerators (LPAs) [1].

The wake phase velocity strongly depends on the pulse propagation velocity and determines the dephasing length, the distance for a relativistic particle to move out of an accelerating phase. Therefore, it limits the maximum energy gain of the accelerated electrons [63], as well as the trapping/injection threshold for background plasma electrons [60, 61] and the maximum amplitude of the plasma wave [64]. The pulse depletion rate also limits the energy gain and the quality of the accelerated electron bunch.

A calculation of these quantities is essential for the design and understanding of present and future LPA experiments.

In laser plasma accelerators, the laser ponderomotive force drives the electron plasma wave. An important parameter in the discussion of intense laser-plasma interactions is the normalized laser strength parameter a_0, defined as the peak amplitude of the normalized vector potential of the laser field $a = eA/m_e c^2$. The laser strength parameter is related to the peak laser intensity I_0 by $I_0 = (\pi c/2)(m_e c^2 a_0/e\lambda)^2$, which yields $a_0 \simeq 7.32 \times 10^{-19} \lambda_0^2[\mu m] I_0[W/cm^2]$, where a linearly polarized laser field is assumed, $\lambda = 2\pi/k$ is the laser wavelength, m_e is the electron mass, e is the electron charge, c is the speed of light in vacuum and $\omega = ck$ is the laser frequency in vacuum.

For low laser intensities $a_0^2 \ll 1$, the phase velocity of the plasma wave is approximately the group velocity of the laser. For a low-intensity laser pulse propagating in a uniform, underdense plasma ($\omega_p^2/\omega_0^2 \ll 1$), the linear laser group velocity is $v_g/c = 1 - \omega_p^2/2\omega_0^2$

in the one-dimensional (1D) limit, where $\omega_p = k_p c = 2\pi c/\lambda_p = (4\pi n_0 e^2/m_e)^{1/2}$ is the plasma frequency and n_0 is the unperturbed neutral plasma number density. In the linear regime, the Lorentz factor of the plasma wave is therefore $\gamma_p \simeq \gamma_g = \omega_0/\omega_p$.

Some approximate expressions have been calculated in other limited regimes. Lu et al. used particle-in-cell simulations to estimate a constant phase velocity $\gamma_p = \omega_0/\sqrt{3}\omega_p$ in the blowout regime ($a_0 \sim 4$) and a phase velocity of $\gamma_p = \sqrt{a_0}\omega_0/\omega_p$in the nonlinear 1D regime [56, 55]. Earlier work numerically showed that $v_p < v_g$ for the nonlinear 1D regime [62]. More typically, in literature, the wake velocity v_p has been approximated by the linear group velocity of the laser $v_p \simeq v_g$.

Schroeder et al. [7] have shown that this is a poor approximation in the nonlinear regime ($a_0 > 1$), which is of interest for the present LPA experiments. Investigating the evolution of a short and intense pulse in an under-dense plasma, they show that the wake phase velocity is determined by the nonlinear laser intensity transport and laser evolution. The nonlinear intensity transport and group velocities of the laser pulse and the nonlinear phase velocity of the excited plasma wave were computed assuming a 1D, broad pulse limit.

For the 3D geometry, Schroeder et al. [58] have obtained a theory for the wake velocity in the low intensity $a_0 \ll 1$ regime, in which the plasma density perturbation can be neglected for describing the laser evolution.

In addition, the wake phase velocity evolves because of the driver's energy depletion process [7]. In the work of Shadwick et al. [59], an analytical theory describing the process of energy depletion of short ($k_p L \sim 1$) and intense ($a_0 > 1$) laser pulses propagating in a under-dense plasma was developed using the 1D wave equation.

In this chapter, we investigate the propagation of weakly relativistic $a_0 < 1$ laser pulse in a under-dense plasma channel, in 3D. We analytically derive an expression for the laser energy depletion rate and we compute the velocity of the pulse intensity centroid, of the laser intensity peak and the phase velocity of the excited plasma wave. These are calculated by using the envelope approximation for the laser evolution and the linearized quasi-static approximation for the plasma response. The analytical solutions are shown to be in good agreement with numerical solutions of the full quasi-static equations obtained with the ponderomotive code INF&RNO [26, 27, 28].

1.1. Quasi static equations in the weakly relativistic regime

We adopt non-dimensional, "comoving" variables defined as $\zeta = k_p(z—ct)$ (longitudinal) and $r = k_p r_{\text{physical}}$ (transverse), where $k_p = \omega_p/c$, ω_p is the plasma frequency corresponding to the chosen reference density n_0, and c is the speed of light. The time is also rescaled with ω_p, that is $\tau = \omega_p t$.

Laser propagation is considered in a cold, collisionless, under-dense plasma (with immobile ions) and the laser pulse is described using an envelope model. Denoting by $a_\perp = eA_\perp/mc^2$ the normalized vector potential of the laser, the envelope $\hat{a}$ is defined by $a_\perp = \frac{\hat{a}}{2} e^{i(k_0/k_p)\zeta} + c.c.$. The envelope evolves according to [1]:

$$\left[\nabla_\perp^2 + 2\left(i\frac{k_0}{k_p} + \partial_\zeta\right)\partial_\tau - \partial_\tau^2\right]\hat{a} = \rho\hat{a}, \tag{1.1.1}$$

where $2\pi/k_0$ is the central laser wavelength, $\rho = \frac{1}{\gamma}\frac{n}{n_0}$ is the (normalized) plasma proper density and γ is the relativistic gamma factor associated with the local plasma fluid velocity.

The plasma response affects the laser envelope evolution by means of the proper density term in equation 1.1.1. Solutions in the nonlinear regime can be obtained numerically by integrating the full system of plasma/wakefield quasi-static equations coupled self-consistently with the envelope equation. Analytical or semi-analytical solutions can be obtained in the weakly relativistic limit $|\hat{a}| < 1$, where the laser contribution can be treated as a perturbation. In this limit, $\rho = \rho_0 + \delta\rho$, where $\rho_0(r) = n_0(r)/n_0$ is the unperturbed background density (in our case, a parabolic plasma channel of radius R $\rho_0 = 1 + 4r^2/R^4$) and $\delta\rho$ satisfies

$$\left(\partial_\zeta^2 + 1\right)\delta\rho = -\left(1 - \nabla_\perp^2\right)\frac{|\hat{a}|^2}{4} \tag{1.1.2}$$

The electromagnetic wakefield is described by fields normalized to $E_0 = m_e c\omega_p/e$, where m_e and e are respectively mass and charge of the electron.

We use the notation

$$\hat{a} = a_0 \exp\left(-r^2/W^2\right)\exp\left(-\zeta^2/L^2\right) \tag{1.1.3}$$

to indicate a monochromatic gaussian laser pulse with peak intensity a_0, normalized transversal spot size at focus (waist) $W = w_0$ and length L.

Assuming that $\frac{k_p}{k_0}\partial_\zeta |\hat{a}| \sim \frac{k_p}{k_0}|\hat{a}|/L \ll 1$, which holds for a short $L \lesssim 2$ pulse at early times (before depletion takes place), the envelope equation (1.1.1) simplifies to:

$$\partial_\tau \hat{a} \simeq -\frac{i}{2}\frac{k_p}{k_0}\left[\rho\hat{a} - \nabla_\perp^2 \hat{a} + i\frac{k_p}{k_0}\partial_\zeta\left(\rho\hat{a} - \nabla_\perp^2\hat{a}\right)\right], \tag{1.1.4}$$

in which the expansion of the operator $\left(1 - i\partial_\zeta \frac{k_p}{k_0}\right)$ has been used.

1.2. Expressions for Energy Depletion

The expression for the normalized pulse energy,

$$\begin{aligned} \varepsilon \ = \ & \frac{k_p^2}{k_0^2}\int d\zeta \int dr\, r \cdot \\ & \left\{\left[\frac{k_0}{k_p}\hat{a} - i\frac{\partial \hat{a}}{\partial \zeta}\right]\left[\frac{k_0}{k_p}\hat{a}^* + i\frac{\partial \hat{a}^*}{\partial \zeta}\right] + \frac{1}{2}\frac{\partial \hat{a}}{\partial r}\frac{\partial \hat{a}^*}{\partial r}\right\} \end{aligned}$$

derives from the integration of the electromagnetic energy density $dU = \frac{1}{8\pi}\left(\left(\overrightarrow{E}\right)^2 + \left(\overrightarrow{B}\right)^2\right)$ in the Coloumb gauge $\nabla \cdot \overrightarrow{A} = 0$ (in which $\overrightarrow{E} = -\frac{1}{c}\frac{\partial \overrightarrow{A}}{\partial t}$, $\overrightarrow{B} = \nabla \times \overrightarrow{A}$) for the laser field in its slow-varying complex envelope representation $\hat{a}$.

The laser energy depletion rate can be naturally split in two parts, namely the terms containing longitudinal derivatives and the terms containing transverse ones:

$$\partial_\tau \varepsilon = \partial_\tau \varepsilon_{\|} + \partial_\tau \varepsilon_\perp,$$

where

$$\begin{array}{ll} \partial_\tau \varepsilon_{\|} = & \frac{k_p^2}{k_0^2}\int d\zeta \int dr\, r\, \partial_\tau \left\{\left[\frac{k_0}{k_p}\hat{a} - i\frac{\partial \hat{a}}{\partial \zeta}\right]\left[\frac{k_0}{k_p}\hat{a}^* + i\frac{\partial \hat{a}^*}{\partial \zeta}\right]\right\} \\ \partial_\tau \varepsilon_\perp = & \frac{k_p^2}{k_0^2}\int d\zeta \int dr\, r\, \partial_\tau \left\{\frac{1}{2}\frac{\partial \hat{a}}{\partial r}\frac{\partial \hat{a}^*}{\partial r}\right\} \end{array} .$$

Using the laser envelope equation (Eq. (1.1.1)) and its complex conjugate to simplify the longitudinal terms, the expression for energy rate becomes

$$\partial_\tau \varepsilon = -\frac{k_p^2}{2k_0^2}\int d\zeta \int dr\, r \left\{\rho \partial_\zeta \left(|\hat{a}|^2\right)\right\} + \partial_\tau \varepsilon_\perp$$

The transverse term in the energy depletion rate is of higher order in $k_p/k_0 \ll 1$ than the longitudinal part, as can be shown by expanding the temporal derivative with the operatorial expansion of the envelope equation (1.1.4). Hence, integrating by parts, the energy depletion rate becomes:

$$\partial_\tau \varepsilon \simeq -\frac{k_p^2}{2k_0^2}\int d\zeta \int dr\, r\, (\partial_\zeta \rho)\, |\hat{a}|^2 + O\left(\frac{k_p^4}{k_0^4}\right). \tag{1.2.1}$$

The rate of change of the normalized intensity $Q = \int d\zeta \int dr\ r\ |\hat{a}|^2$, describing the early-time self steepening of a laser pulse, is therefore

$$\partial_\tau Q \simeq -\partial_\tau \varepsilon. \tag{1.2.2}$$

The same equivalence was found to be valid in the 1D limit [59, 7].

In the case of a weakly relativistic ($a_0 < 1$) gaussian pulse (1.1.3), the quasi-linear plasma response (1.1.2) can be used as an approximation for the proper density in (1.2.1), yielding a compact expression for the initial (when the shape of the pulse is still gaussian) value of the energy rate:

$$\partial_\tau \varepsilon / \varepsilon_0|_{\tau=0} \simeq -\frac{1}{32}\sqrt{\frac{\pi}{2}}\frac{k_p^2}{k_0^2} a_0^2 L e^{-\frac{L^2}{4}}\left(1+\frac{4}{W^2}\right) \tag{1.2.3}$$

Solid lines in figure 1.2.1 show the energy depletion rate given by Eq. 1.2.3 for gaussian pulses with different intensities and waists ($L = 2$ and $k_0/k_p = 20$) propagating in a matched parabolic plasma channel. The theory shows good agreement with the solution of the full envelope and quasi static plasma equations (obtained numerically with INF&RNO simulations), up to the limit $a_0 < 1$.

In the broad pulse limit $W \gg 1$, the energy rate is given by:

$$\partial_\tau \varepsilon / \varepsilon_0|_{\tau=0} \simeq -\frac{k_p^2}{k_0^2}\frac{1}{16e}\sqrt{\frac{\pi}{2}} a_0^2,$$

which provides a better prediction than the 1D theory result $\partial_\tau \varepsilon / \varepsilon_{0(1D)} = -\frac{k_p^2}{k_0^2}\frac{1}{8e}\sqrt{\frac{\pi}{2}} a_0^2$, as in the 1D model the laser intensity, and hence the energy depletion rate, is radially constant.

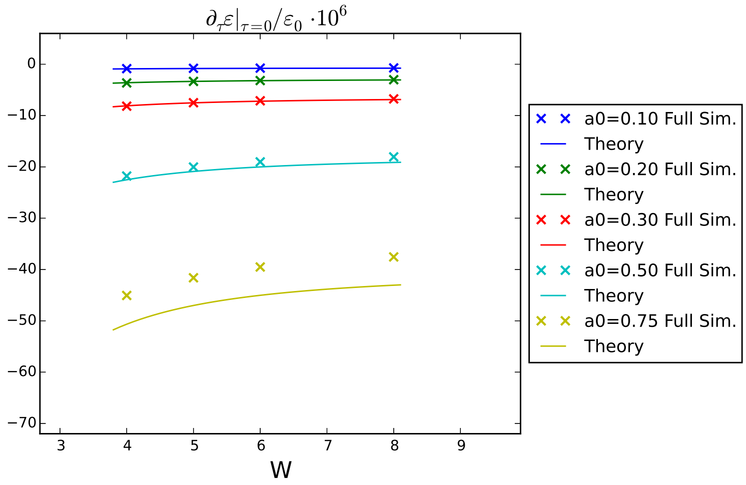


Figure 1.2.1.: Initial depletion rate for a resonant ($L = 2$) gaussian pulse propagating in a matched parabolic plasma channel with on-axis density $k_0/k_p = 20$. The theoretical expression for the normalized depletion rate (Eq. 1.2.3, solid lines) show good agreement with INF&RNO simulations in the weakly relativistic regime (crosses).

1.3. Expressions for Laser Intensity Transport Velocity and for the velocity of the Intensity Peak

The laser intensity transport velocity can be defined as $\partial_\tau \zeta_l = \beta_l - 1$, where ζ_l is the laser intensity centroid, defined as the position weighted by $|\hat{a}|^2$

$$\zeta_l = \frac{\int d\zeta \int dr\; r\; \zeta\; |\hat{a}|^2}{\int d\zeta \int dr\; r\; |\hat{a}|^2} = \frac{G}{Q}$$

The intensity transport velocity can be computed at early times using the operatorial expansion of the envelope equation (1.1.4). For an initially longitudinally-symmetric pulse $G_{\tau=0} = 0$ holds and the expression for the intensity transport velocity simplifies to:

$$\partial_\tau \zeta_l|_{\tau=0} = -\frac{\partial_\tau G}{Q_0} = -\frac{1}{Q_0}\frac{k_p^2}{2k_0^2}\int d\zeta \int dr\; r\; \{2\rho|\hat{a}|^2 + \zeta\rho\partial_\zeta|\hat{a}|^2 - \frac{\partial \hat{a}}{\partial r}\frac{\partial \hat{a}^*}{\partial r}\} \tag{1.3.1}$$

As derived in Ref. [58], the intensity transport velocity of a gaussian pulse propagating in a parabolic plasma channel of radius R is, in the low-power limit ($a_0 \ll 1$, $\rho = \rho_0$):

$$\beta_l|_{\tau=0} \simeq 1 - \frac{k_p^2}{2k_0^2}\left(1 + \frac{2W^2}{R^4}\right) - \frac{k_p^2}{k_0^2}\frac{1}{W^2}. \tag{1.3.2}$$

Considering propagation in vacuum ($\rho = 0$), only the diffraction term gives a contribution in the intensity transport velocity formula (Eq 1.3.1) and, hence, $\beta_l|_{\tau=0} \simeq 1 - \frac{1}{Q_0}\frac{k_p^2}{2k_0^2}\int d\zeta \int dr\, r \frac{\partial\hat{a}}{\partial r}\frac{\partial\hat{a}^*}{\partial r} = 1 - \frac{k_p^2}{k_0^2}\frac{1}{W^2}$.

In the weakly relativistic regime ($a_0 < 1$) the quasi-linear plasma response (1.1.2) can be used as an expression for the proper density in (1.3.1), yielding, for a gaussian pulse propagating in a parabolic plasma channel of radius R:

$$\begin{aligned}\beta_l = 1 - \tfrac{k_p^2}{2k_0^2}\Big\{&1 + \tfrac{2W^2}{R^4} + \tfrac{2}{W^2} - \tfrac{\sqrt{2}}{16}a_0^2\left(1 + \tfrac{4}{W^2}\right)\cdot \\ &L\left[\tfrac{3}{2}P_0(L) - P_2(L)\right]\Big\},\end{aligned}$$

in which $P_m(L) = \int_0^\infty du\, sin(Lu)\, u^m e^{-u^2}$.

In particular, in the case of a resonant Gaussian pulse ($L = 2$) propagating in a matched plasma channel ($R = W$):

$$\gamma_l|_{\tau=0} \simeq \frac{k_0}{k_p}\left(1 + \frac{4}{W^2}\right)^{-1/2}\left(1 + 0.0509\, a_0^2\right) \tag{1.3.3}$$

In the weakly relativistic regime, intensity-dependent effects increase the transport velocity, as results from a comparison between the weakly relativistic (Eq 1.3.3) and the low-power case (Eq 1.3.2), in which all the intensity dependent terms were neglected a priori. The increase of the transport velocity is due to the plasma density perturbation caused by the laser pulse, assumed to depend on the ponderomotive force according to a linearized relation (Eq 1.1.2) for $a_0 < 1$. Qualitatively, as the plasma goes through the pulse, the ponderomotive effects carve a lower density channel, in which laser propagation is faster.

The velocity of the on-axis point $\zeta_l^*(\tau)$ having the peak value of the intensity, for which$\partial_\zeta|\hat{a}|^2\big|_{r=0,\zeta=\zeta_l^*(\tau)} = 0$ holds, is given by

$$\partial_\tau\zeta_l^* = \beta_l^* - 1 = -\frac{\left.\frac{\partial^2|\hat{a}|^2}{\partial\zeta\partial\tau}\right|_{\zeta=\zeta_l^*,r=0}}{\left.\frac{\partial^2|\hat{a}|^2}{\partial\zeta^2}\right|_{\zeta=\zeta_l^*,r=0}} \tag{1.3.4}$$

Using the operatorial expansion of the envelope equation Eq. (1.1.4), Eq. (1.3.4) can be expressed at early times, for a monochromatic laser pulse, as

$$\partial_\tau \zeta_l^* = -\frac{k_p^2}{k_0^2} \frac{\partial_\zeta \left(a \partial_\zeta \left(\rho a - \nabla_\perp^2 a\right)\right)}{\partial_\zeta^2 |\hat{a}|^2}\Bigg|_{\zeta=\zeta^*, r=0} \tag{1.3.5}$$

For an initially Gaussian pulse, $\zeta_l^*(\tau = 0) = 0$ holds and Eq (1.3.5) simplifies to

$$\partial_\tau \zeta_l^*|_{\tau=0} = -\frac{k_p^2}{2k_0^2}\left(1 + \frac{4}{W^2} + (\rho - \rho_0)|_0 - 2\partial_\zeta^2 \rho\big|_0\right) \tag{1.3.6}$$

In the low power limit $a_0 \ll 1$, the density perturbation can be neglected ($\rho = \rho_0$) and the velocity of the intensity peak of a gaussian pulse propagating in a matched ($R = W$) plasma channel (Eq 1.3.6) is equal to the velocity of its intensity centroid (the intensity transport velocity, Eq 1.3.2).

As the intensity grows, the laser pulse ponderomotive interaction with the plasma creates an asymmetric longitudinal plasma density profile, that affects the intensity transport velocity (Eq 1.3.1) and the evolution of the pulse shape, that reflects in the velocity of the intensity peak (Eq 1.3.6). The proper density terms in Eq. (1.3.6) depend on the laser intensity and they can be computed for a weakly relativistic intense laser pulse ($a_0 < 1$) using Eq. (1.1.2) for the proper density perturbation. In the case of a weakly relativistic, initially gaussian pulse propagating in a matched plasma channel, the initial value of the intensity peak relativistic factor (1.3.6) is given by

$$\gamma_l^*|_{\tau=0} \simeq \frac{k_0}{k_p}\left[1 + \frac{4}{W^2} - 0.0436 a_0^2 \left(1 + \frac{8}{W^2}\right)\right]^{-1/2} \tag{1.3.7}$$

This theory, including the proper density perturbation, shows that the laser intensity peak velocity increases with the intensity, as the laser intensity transport velocity (Eq 1.3.3) does.

The plots in figure 1.3.1 show the Lorentz factors of the intensity centroid and intensity peak versus intensity and waist of various gaussian pulses ($L = 2$ and $k_0/k_p = 20$). In the weakly relativistic regime ($a_0 < 1$), the theory shows good agreement with the numerical solution of full the plasma/envelope equations.

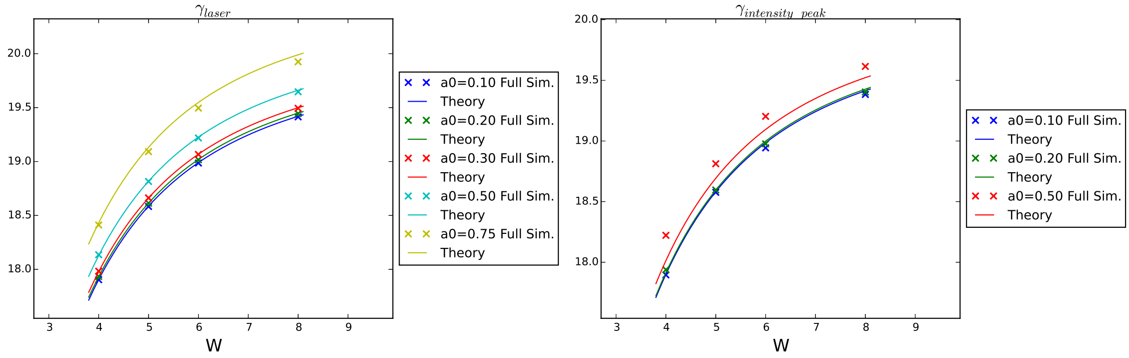


Figure 1.3.1.: Initial Lorentz factor of the intensity centroid (left) and of the intensity peak (right) versus intensity and waist of a gaussian laser pulse with $k_0/k_p = 20$ propagating in a matched parabolic plasma channel. The solid curves show the value predicted by (1.3.3) and the crosses are full numerical solutions computed with INF&RNO.

1.4. Expression for the Wake Phase Velocity

The plasma wave phase velocity is determined by the intensity transport velocity and the evolution of the laser. It can be defined as the velocity of the on-axis zero crossing point ζ_* of the accelerating field $E_z = -\frac{\partial\Psi}{\partial\zeta}$ (where Ψ the wake potential) located at the back the first wave period, satisfying

$$E_z(\zeta_*(\tau),\tau) = 0,\, \left.\frac{\partial\Psi}{\partial\zeta}\right|_{\zeta=\zeta_*,r=0} = 0$$

Imposing $E_z(\zeta_*(\tau),\tau) = 0$, $E_z(\zeta_*(\tau+\Delta\tau),\tau+\Delta\tau) = 0$ and taking the limit $\Delta\tau \to 0$, the phase velocity of the zero crossing point is given by

$$\frac{\partial\zeta_*}{\partial\tau} = \beta_* - 1 = -\frac{\left.\frac{\partial^2\Psi}{\partial\xi\partial\tau}\right|_{\zeta=\zeta_*,r=0}}{\left.\frac{\partial^2\Psi}{\partial\xi^2}\right|_{\zeta=\zeta_*,r=0}} \tag{1.4.1}$$

In the weakly relativistic regime ($a_0 < 1$), the quasi-static wake potential equations are well approximated by [1]

$$\frac{\partial^2\psi}{\partial\zeta^2} = -\psi + \frac{|\hat{a}|^2}{4}. \tag{1.4.2}$$

From the semi-analytic solution of Eq 1.4.2, an equation for the position of the zero-crossing point ζ_*can be found

$$\int_{\zeta_*}^{\infty} \cos\left(\zeta_* - \zeta\right) |\hat{a}|^2 = 0, \tag{1.4.3}$$

and the expression for its velocity Eq. 1.4.1 becomes

$$\beta_* - 1 = \frac{\int_{\zeta_*}^{\infty} d\zeta \cos\left(\zeta_* - \zeta\right) \partial_\tau |\hat{a}|^2}{|\hat{a}_{\zeta=\zeta_*}|^2 + \int_{\zeta_*}^{\infty} d\zeta \sin\left(\zeta_* - \zeta\right) |\hat{a}|^2} \tag{1.4.4}$$

The zero crossing point ζ_* of the wakefield generated by a short ($L \lesssim 2$) Gaussian pulse is relatively far from the pulse centroid and it is possible to approximate the laser envelope field to be null at such distances, $|\hat{a}_{\zeta \leq \zeta_*}|^2 \simeq 0$. In this case, the solution of Eq 1.4.3 is $\zeta_* = -\frac{3}{2}$. Furthermore, for a short pulse, the integrals in Eq 1.4.4 can be extended to infinity.

The time derivative of the squared modulus of complex envelope $\partial_\tau |\hat{a}|^2$ in Eq. 1.4.4 can be computed at early times by using the operatorial expansion (Eq. 1.1.4) of the envelope equation and, hence, the integrals in Eq. 1.4.4 can be computed semi-analytically, for a weakly relativistically intense, monochromatic gaussian laser pulse.

In particular, for a resonant ($L = 2$) gaussian pulse, propagating in a matched parabolic plasma channel, the initial relativistic factor of the wake is

$$\gamma^*|_{\tau=0} \simeq \frac{k_0}{k_p} \left[1 + \frac{4}{W^2} - 0.192682 a_0^2 \left(1 + \frac{8}{W^2}\right)\right]^{-1/2} \tag{1.4.5}$$

Figure 1.4.1 shows the agreement of equation1.4.5 with full numerical solutions obtained with INF&RNO.

At later times, the evolution of the laser pulse modifies the position of the zero-crossing point and hence the phase velocity of the plasma wake. Equations describing the evolution of the slowly varying laser field envelope can be derived by analyzing the paraxial wave equation and by applying the source-dependent expansion method [101]. In the source-dependent expansion method, the laser field $\hat{a}$ is assumed to be well approximated by the fundamental Gaussian mode of the form

$$\hat{a} = a_0 \frac{W}{w(\zeta)} e^{-\xi^2/L^2} e^{-(1-i\alpha)\frac{r^2}{w(\zeta)^2}}, \ \alpha = \frac{1}{2}\frac{k_0}{k_p} w \frac{\partial w}{\partial \tau} \tag{1.4.6}$$

where the spot size $w(\zeta)$ temporally evolves as [57]

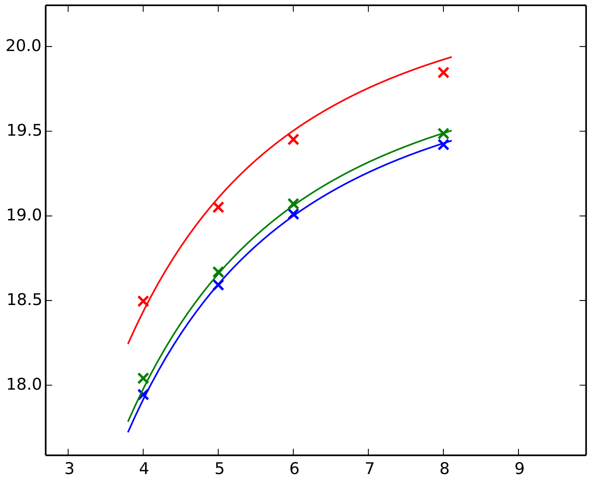


Figure 1.4.1.: Lorentz factor of the initial phase velocity of the zero crossing point located at the back of the first bucket for a resonant ($L = 2$) gaussian pulse propagating in a matched parabolic plasma channel and various intensities and waists.

$$\frac{\partial^2 w}{\partial \tau^2} = \left(\frac{k_p}{k_0}\right)^2 \frac{8}{w^3}\left[\frac{1}{2} - \int dr r\rho \left(\frac{2r^2}{w^2} - 1\right) e^{-\frac{2r^2}{w^2}}\right] \tag{1.4.7}$$

For a laser pulse described by the fundamental gaussian mode 1.4.6, the position of the zero crossing point located at the back of the first bucket is

$$\zeta * = -\frac{3}{2}\pi + \theta_*,$$

where θ_* is a correction that depends on the spot size longitudinal distribution

$$\tan\theta_* = \frac{\int_{-\infty}^{+\infty} d\zeta \sin\xi \frac{e^{-\zeta^2/L^2}}{w^2(\zeta)}}{\int_{-\infty}^{+\infty} d\zeta \cos\xi \frac{e^{-\zeta^2/L^2}}{w^2(\zeta)}} \tag{1.4.8}$$

In the case of a longitudinally symmetric pulse, the numerator in Eq 1.4.8 is zero and the zero-crossing point position is given by $\xi_* = -\frac{3}{2}\pi$, whereas, in general, the paraxial evolution prescribed by *Eq. 1.4.7* introduces asymmetries in the waist distribution $w(\zeta)$ and oscillations of the zero crossing point.

This theory predicts that the initial value for the wake velocity (1.4.5) is close to the velocity of the maximum intensity point (1.3.7). The correlation was further investigated, at later evolution times, with INF&RNO simulations. Figure 1.4.2 presents numerical results showing that the values of the relativistic factor of the peak of the pulse intensity and of the relativistic factor of the wake remain very close for a few matched Rayleigh lengths of propagation in the plasma channel, and their temporal evolution is qualitatively different from the relativistic factor of the laser intensity centroid. The damping at later times of the oscillations is due to the fact that a short laser driver, as the one used in this simulation, is not monochromatic. Each chromatic component of the pulse is characterized by a different oscillation frequency and the decoherence between these modes damps out the oscillations.

The laser intensity peak moves longitudinally as the pulse transversally evolves as in Eq. 1.4.7, due to the paraxial evolution of the transverse slices that interact with different plasma densities depending on their longitudinal position.

The oscillations of the plasma wake velocity that are observed at the matched Rayleigh length timescale are a consequence of the transverse evolution of the pulse. Such evolution alters its longitudinal symmetry, which induces variations in Eq. 1.4.8, and also determines to the position of the peak of the intensity.

On the contrary, the change in the intensity transport velocity, owing to longitudinal terms in the envelope equation, affect the wake velocity the over the pump depletion time scale.

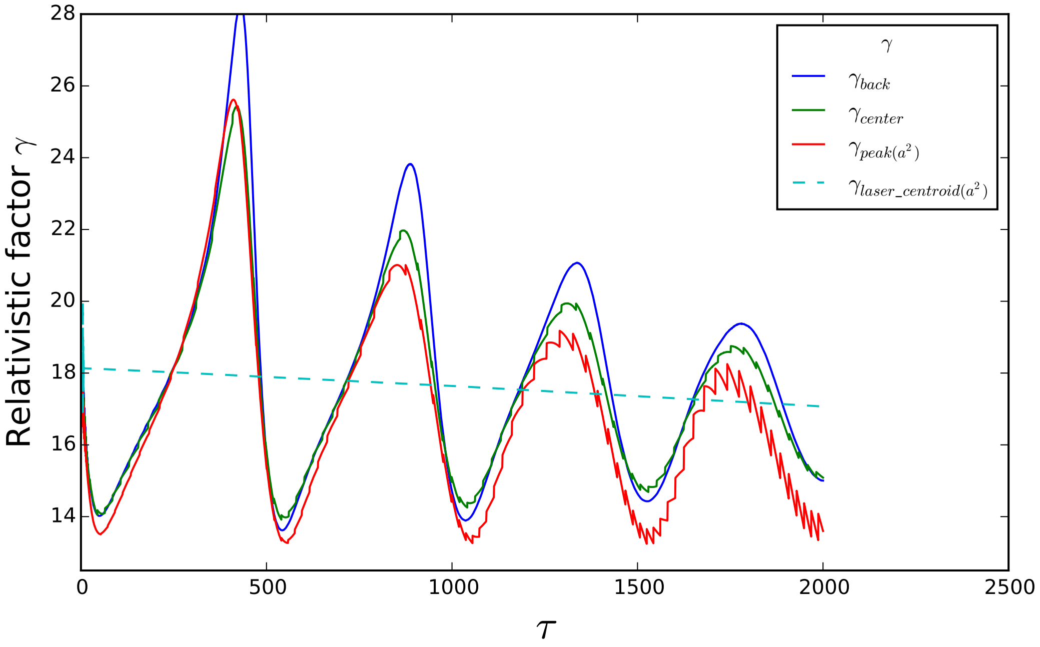


Figure 1.4.2.: Temporal evolution of the relativistic factor of various points of interest for a wake generated by gaussian laser pulse propagating in a matched plasma channel, with intensity $a_0 = 0.5$, waist $W = 4$ and $k_0/k_p = 20$. The relativistic factor of the center and back of the wake follow the evolution of the laser intensity peak (maximum value of the intensity), while the laser intensity centroid (weighted average of a^2 over the ζ coordinate) evolves on a slower timescale.

In figure 1.4.3, we separately analyze the effects that either the transversal or the longitudinal evolution of the pulse have on the wake velocity. Red lines show a virtual experiment in which longitudinal evolution was suppressed by integrating the paraxial equation instead of the full envelope equation for the laser pulse, while in the case indicated by the green lines a laser pulse that does not undergo any transverse evolution is considered, in a similar fashion as in Benedetti et al. [57] and as described in [22]. In the latter case, only the slower longitudinal effects determine the laser evolution and hence the wake velocity.

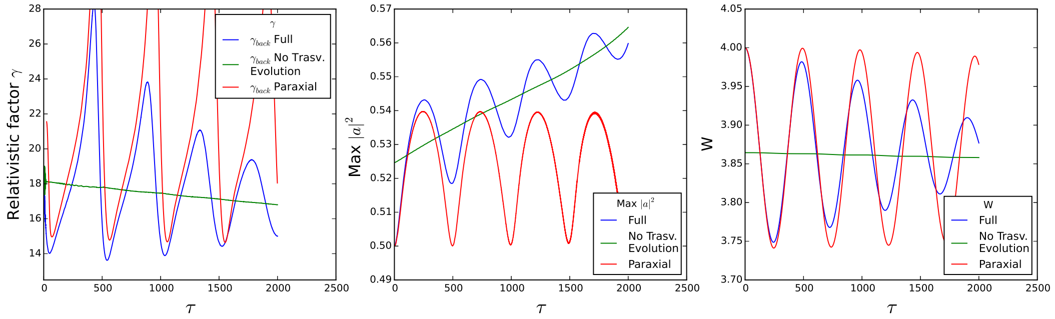


Figure 1.4.3.: Temporal evolution of the relativistic factor of the wake for a gaussian laser pulse propagating in a matched plasma channel, with intensity $a_0 = 0.5$, waist $W = 4$ and $k_0/k_p = 20$. The effects on the wake velocity caused by the transverse and longitudinal evolution of the pulse shape are studied separately by considering either the paraxial evolution of the laser envelope (neglecting the longitudinal evolution or hence the energy depletion/self-steepening, red lines) and suppressing transverse evolution (green lines) and compared to the full dynamics (blue lines).

1.5. Depletion Rate and Laser Intensity centroid velocity in the $a_0 > 1$ regime

For higher intensities ($a_0 > 1$), the perturbative approximation 1.1.2 is no longer accurate. In Krall's theory [93], the 3D proper density follows from the wake potential as:

$$\rho = \frac{1}{\Psi + 1}\left(1 - \nabla_{\perp}^2 \Psi\right) \tag{1.5.1}$$

For computing the depletion rate, we make the assumption that, slice by slice, transversally, the one dimensional equation for the potential [1] applies:

$$\frac{\partial^2 \Psi\left(v^2, \zeta\right)}{\partial \zeta^2} = \frac{1}{2}\left[\frac{1 + v^2 f^2(\zeta)/2}{(\Psi + 1)^2} - 1\right]$$

, where we also assumed that rated the laser pulse envelope is separable as $|\hat{a}|^2 = \tilde{a}^2(r) \cdot f^2(\zeta)$, and $\nu = \tilde{a}(r)$. This assumption can be motivated by the importance of longitudinal terms in the depletion formula 1.2.1 and it produces a 1D-potential - 3D-density hybrid model, in which 1D theory is used transversally for the wake potential, but the density distribution is obtained using a 3D law 1.5.1.

For $L < W$, the validity of the assumption was benchmarked using quasi-static numerical simulations. In figure 1.5.1, the proper density field obtained with the full quasi-static model is compared with the one obtained by means of the 1D hybrid model. In the region of interest for the depletion integral (i.e. the laser pulse) no significant difference is observed for laser plasma parameters $L = 2$, $W = 4$, $a_0 = 3.0$. In particular, the longitudinal integrand $I = \int dr\, r\, a^2 \partial_\zeta \rho$ in the depletion rate formula $\partial_\tau \varepsilon \simeq -\frac{k_p^2}{2k_0^2} \int d\zeta I$ computed with the hybrid model (red solid line in figure) is in very good agreement with the full solution of the quasi static equations (blue solid line). On the contrary, computing the same integral using a 1D formula for the proper density in place of 1.5.1 (neglecting the transverse laplacian) introduces a noticeable discrepancy (green solid line).

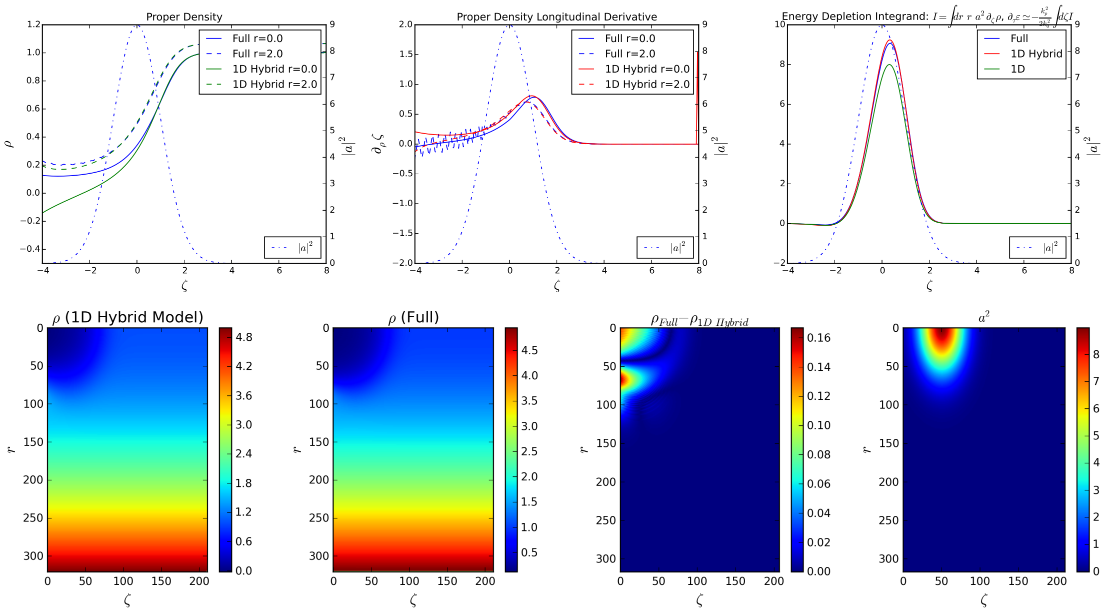


Figure 1.5.1.: Proper density ρ, proper density longitudinal derivative $\partial_\zeta \rho$ and longitudinal integrand $I = \int dr\, r\, a^2 \partial_\zeta \rho$ of the depletion rate $\partial_\tau \varepsilon \simeq -\frac{k_p^2}{2k_0^2} \int d\zeta I$, numerically computed using fully dimensional, hybrid (1D potential, 3D proper density) and 1D models. For laser plasma parameters are $L = 2$, $W = 4$ $a_0 = 3.0$, the proper density computed with the hybrid model, and hence depletion rate, is in very good agreement with the solutions of the full quasi-static equations.

Substituting the proper density in Eq 1.2.1 with Krall's formula 1.5.1, we get an expression for the energy depletion rate

$$\partial_\tau \varepsilon \simeq \frac{k_p^2}{k_0^2}\int dr\ \left\{I_0\left(\tilde{a}^2(r)\right) r + I_1\left(\tilde{a}^2(r)\right)\left[\tilde{a}^2(r)' + r\tilde{a}^2(r)''\right] + I_2\left(\tilde{a}^2(r)\right)\left[\tilde{a}^2(r)'\right]^2 r\right\} \tag{1.5.2}$$

, where I_0, I_1, I_2 are quantities that depend only on the 1D theory potentials $\Psi(v^2, \zeta)$, and they can be fully computed numerically, as shown in figure 1.5:

$$\begin{cases} I_0(v^2) = \frac{1}{2}v^2 \int d\zeta \quad \frac{1}{\Psi(v^2,\zeta)+1}\,\partial_\zeta f^2(\zeta) \\ I_1(v^2) = \frac{1}{2}v^2 \int d\zeta \quad \frac{1}{\Psi(v^2,\zeta)+1}\partial_{v^2}\Psi(v^2,\zeta)\,\partial_\zeta f^2(\zeta) \\ I_2(v^2) = \frac{1}{2}v^2 \int d\zeta \quad \frac{1}{\Psi(v^2,\zeta)+1}\partial^2_{v^2}\Psi(v^2,\zeta)\,\partial_\zeta f^2(\zeta) \end{cases}$$

This way, we are able to compute the depletion rate in the nonlinear case, in 3D and for any waist W, transversally integrating quantities that are simple, pre-computable, "universal" 1D theory results, that depend only on the radial value profile of the intensity.

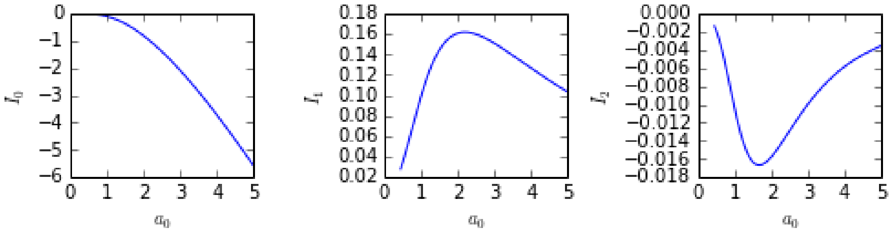


Figure 1.5.2.: I_0, I_1, I_2 as a function of a_0. These quantities depend only on the 1D theory potentials $\Psi(v^2, \zeta)$ and the laser longitudinal shape $f(\zeta)$.

In figure 1.5 we show the agreement of Eq. 1.5.2 with full numerical solutions, obtained with INF&RNO. Excellent agreement is verified both for the pulse depletion rate and the laser intensity centroid initial velocity.

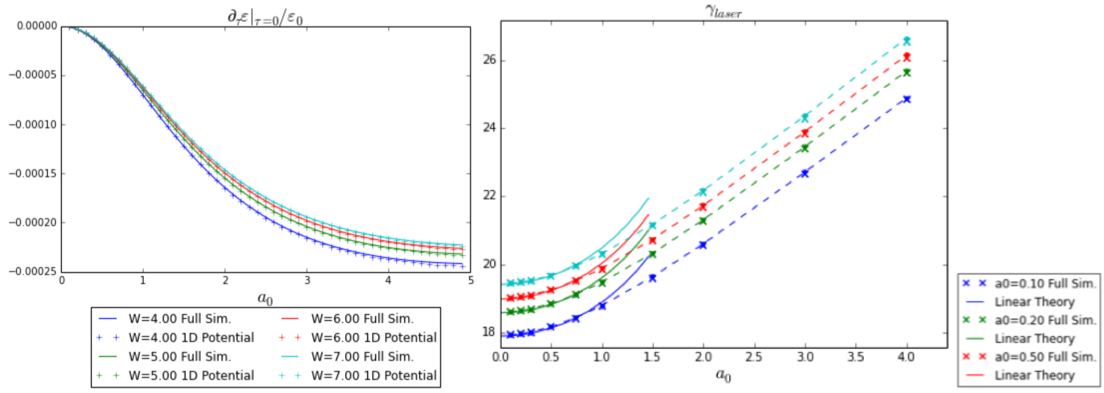


Figure 1.5.3.: Initial depletion rate and laser intensity centroid velocity for a resonant ($L = 2$) gaussian pulse. The 1D potential, 3D proper density model (crosses) shows excellent agreement with the full solution of the quasi static equations (solid lines for depletion rate, dashed lines for γ_{laser}) for the range of parameters $4 < W < 8$ and $a_0 < 5$. Solid lines in the laser intensity centroid box are the analytical results Eq 1.3.3 in the weakly relativistic limit.

2. Wake Velocity and Self-Injection in the Nonlinear Bubble Regime: Numerical Investigation

As of today, most of the laser plasma acceleration experiments have been performed in the so called bubble regime, in which the ponderomotive force of a short and intense laser pulse propagating in an underdense plasma expels background electrons, leading to the formation of a ellipsoidal plasma cavity moving at relativistic velocity, the "bubble" wake.

The bubble regime can be accessed if the laser peak normalized potential of the laser is $a_0 \gtrsim 2$ and $k_p L \sim 1$. The linearly varying longitudinal and transverse fields of the bubble wake have almost ideal accelerating and focusing properties for particles placed in the proper phase.

In some cases, it has been observed, both in experiments and 3D particle-in-cell (PIC) simulations, that electrons from the background plasma can be "self-"injected and accelerated in the bubble. The self-injection process provides the simplest acceleration scheme from the experimental point of view, and understanding its properties is of fundamental importance in order to control and possibly optimize the performance of laser plasma accelerators. Despite its importance, a complete theory of self-injection is still lacking.

In this chapter we systematically analyze the self-injection process by means of fully consistent PIC simulations run under controlled conditions.

The geometrical properties that characterize the bubble wake are discussed in section 2.1. In section 2.2, a laser pulse intensity threshold for self injection is empirically derived as a function of the bubble wake velocity, for the case of non-evolving laser driver (and hence wake). In section 2.3 we study how the laser driver (consistent) evolution affects the temporal evolution of the bubble wake velocity, and hence the self-injection properties.

2.1. Wake Geometrical Properties in the Bubble Regime

The characterization of the geometry of the wake in the bubble regime was carried out by means of numerical simulations.

We considered the bubble wake generated by a non-evolving Gaussian laser pulse, propagating along the $\hat{z}$ direction in an underdense, uniform, cold plasma. The laser envelope is described by:

$$a(z, r, t) = a_0 \exp\left(-r^2/w_0^2\right) \exp\left(-\left(z - z_0\left(t\right)\right)^2/L^2\right)$$

, where $z_0(t)$ is the laser centroid, moving at a constant speed $dz_0/dt = \beta_0$ and relativistic factor γ_0.

The pulse length was set to the linearly resonant length $L = 2$. Since the self-guided propagation of the pulse is of crucial importance for accelerator applications, the value of the pulse waist was taken to match the condition for self-guided propagation of a short and intense laser pulse $w_0 = 2\sqrt{a_0}$ in [56].

In the case of a non-evolving pulse shape, the plasma response, and hence the wake, reaches a stationary state and its geometrical properties depend only on the laser model parameters.

The wake shape and size can be characterized the longitudinal ($R_{\|}$) and transverse ($R_{\perp}$) radii. The longitudinal radius $R_{\|}$is defined as the longitudinal length of the accelerating part of the wakefield and the transverse radius $R_{\perp}$ is defined as the radial distance of the bubble center (defined as the position where all the fields are zero) from the position where the plasma density reaches the background value, growing from being almost zero on the axis.

The geometry of the bubble shows weak dependence on the laser propagation relativistic factor γ_0 for $\gamma_0 > 10$, and, in this regime, the bubble shape can be simply characterized by the normalized laser peak intensity a_0.

We performed a numerical scan to characterize the functional dependence of the bubble shape parameters on the intensity a_0. Simulations were performed with the 2D cylindrical, ponderomotive particle in cell code INF&RNO in the range $2 < a_0 < 7$.

To ensure that the wake properties reach their stationary values as smoothly as possible, the laser-plasma interaction was initialized adiabatically, slowly ramping up both the plasma density and the laser intensity. At early times, we also limited particle velocity to prevent early injection/beam loading, as it could affect the wake shape and make interpretation at later times more difficult.

The stationary plasma response and bubble shape can also be computed using a quasi-static model (see Section 6.2) in this regime.

Simulations show that $R_{\|}$ and $R_{\perp}$ depend linearly on a_0. Figure 2.1 shows the scaling of the radii with the normalized laser intensity and a numerical fit gives, for $R_{parallel}$:

$$R_{\|}(a_0) \simeq 2.9 + 0.305 \cdot a_0 \tag{2.1.1}$$

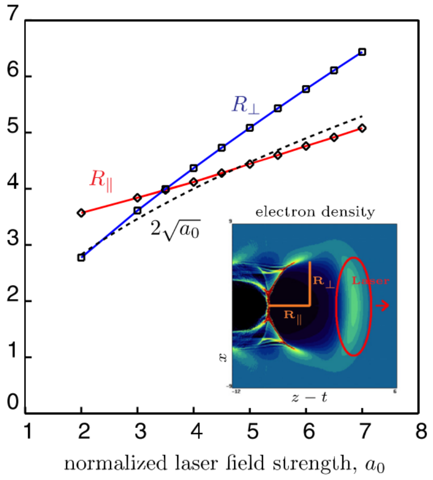


Figure 2.1.1.: Scaling of the bubble radii, $R_{\|}$[red curve] $R_{\perp}$[blue curve], with normalized laser field strength a_0. The black dashed line is the theoretical bubble size proposed in [56], $R = 2\sqrt{a_0}$. The simulation parameters are $L = 2$, $w_0 = 2\sqrt{a_0}$ and $\gamma_0 = 100$. Phys. Plasmas 20, 103108 (2013);

The previous models in Refs. [96, 94, 95] all assume a bubble of spherical shape. In our numerical study, in matched conditions $w_0 = 2\sqrt{a_0}$, we observed spherical symmetry only around $a_0 \simeq 3.5$ (and $w_0 = 2\sqrt{a_0} \simeq 3.75$), and significant deviations for $a_0 > 5$.

In general, besides the intensity a_0, all other parameters in our non-evolving gaussian pulse model affect the ponderomotive push on the plasma electrons and hence the shape of the bubble wake. In order to obtain a condition for the bubble sphericity for any gaussian pulse, we performed a systematic study in the full pulse parameter space, varying the pulse length, the waist and the intensity (L, w_0, a_0). In figure 2.1 the bubble radii difference $R_{\|} - R_{\perp}$ is shown versus a_0 and w_0, for gaussian pulses of different lengths $L = 1.5$, 2.0, 2.5. Each circle is a simulation result in the parameter space (L, w_0, a_0).

For any laser intensity value a_0, spherical bubble wakes ($R_\perp \simeq R_\|$, white circles) were observed only around $w_0 \simeq 3.75$, as for the matched, resonant gaussian pulse case.

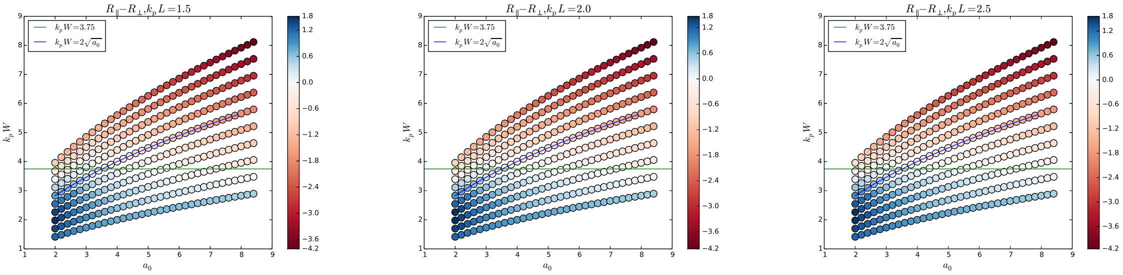


Figure 2.1.2.: Bubble radii difference $R_\| - R_\perp$ versus a_0 and w_0, for a gaussian laser pulse with length $L = 1.5,\ 2.0,\ 2.5$. Each circle is a simulation result in the parameter space (L, w_0, a_0). Blue circles represent a longitudinally elongated bubble shape $R_\| > R_\perp$, while red ones a transversally elongated one $R_\perp > R_\|$. For $w_0 < 3$ and $w_0 > 4$, both increasing the pulse waist w_0 and intensity a_0 result in a relative transversal expansion of the bubble. For any a_0, spherical bubble wakes ($R_\perp \simeq R_\|$, white circles) were observed only around $w_0 \simeq 3.75$.

2.2. Threshold for self-injection for a non-evolving laser

A conclusive theory of particle self-injection and trapping in the 3D nonlinear bubble regime is still missing. In several contributions, the dependence of self-injection threshold on the wake phase velocity is considered to play a major role in the self-injection physics, as discussed in Ref. [94] and [95]. A critical discussion of these models and the admissibility of their hypotheses can be found in Ref [97] and Ref [98].

For a non-evolving wake (that we generate using non-evolving laser), self-injection in the bubble regime can be modeled studying the motion of a generic test particle in the stationary 3D wake. In a reference frame comoving with the laser pulse and the wake, the trajectories of the test particles are governed by the equations:

$$\begin{cases} \frac{\partial\zeta}{\partial t} & = \frac{p_z}{\gamma} - \beta_0 \\ \frac{\partial x}{\partial t} & = \frac{p_x}{\gamma} \\ & \\ \frac{\partial p_z}{\partial t} & = -\frac{1}{2\gamma}\frac{\partial(a^2/2)}{\partial\zeta} + \frac{\partial\Psi}{\partial\zeta} - \frac{p_x}{\gamma}B_y \\ \frac{\partial p_x}{\partial t} & = -\frac{1}{2\gamma}\frac{\partial(a^2/2)}{\partial x} + \frac{\partial\Psi}{\partial x} - \left(\beta_0 - \frac{p_z}{\gamma}\right)B_y \end{cases} \tag{2.2.1}$$

, where β_0 is the wake phase velocity, ζ, x are the longitudinal (comoving with the pulse and the wakefield) and transverse coordinates, $p_z$and $p_x$are the longitudinal/transverse momenta, $\gamma = 1 + a^2/2 + p_z^2 + p_x^2$, a is the laser field amplitude, and Ψ is the wake potential, such that $E_z = -\partial\Psi/\partial\zeta$, $E_x - \beta_0 B_y = \partial\Psi/\partial x$.

The Hamiltonian of the test particle is:

$$H = \gamma - \beta_0 p_z - \Psi$$

If the wake is non-evolving, then H is a constant of motion. In particular, for a test electron initially at rest (a background cold plasma electron), $H = 1$ holds.

A particle is trapped in the wake if its longitudinal velocity is equal or greater than the wake phase velocity β_0 and its location resides in the accelerating/focusing domain of the wakefield. Hence, the for the phase space phase at the moment of injection/trapping $\left(\tilde{\zeta}\,\tilde{x}\,\tilde{p}_z,\,\tilde{p}_x\right)$:

$$\tilde{p}_z/\tilde{\gamma} = \beta_0$$

holds.

Taking $H = 1$ for an electron initially at rest and assuming that self-injection occurs far behind the laser pulse, where the laser field amplitude is neglectable $\tilde{a} \simeq 0$, the necessary condition for trapping is

$$\Psi = \Psi(\tilde{\zeta}, \tilde{x}) = -1 + \frac{\sqrt{1 + (\tilde{p_x})^2}}{\gamma_0} \tag{2.2.2}$$

, in terms of the wake potential at the moment of trapping.

In Eq 2.2.2 we see that trapping is facilitated (i.e. it requires a less negative potential to happen) at low wake phase velocities. Furthermore, equation 2.2.2 suggest that the wake velocity should play a very important role in determining the injection threshold, while for a non-evolving gaussian laser the potential map $\Psi(\zeta, x)$ depends only on a_0.

Taking a non-evolving laser pulse as the driver, the laser pulse shape evolution (which is a combination of depletion, self-steepening and focusing effects) is decoupled from the analysis of the injection mechanism. Under these controlled conditions we can determine when self-injection occurs and relate its appearance to the wake velocity and laser intensity (i.e., wake size and amplitude).

We studied the threshold for self injection performing several INF&RNO PIC simulations in the (γ_0, a_0) parameter space. The stationary wakefield was initialized as described in section 2.1. After the wake was cleanly initialized, we measured the about of self-injected charge, after a fixed laser propagation length. In figure 2.2, two regions are clearly separated: an injection domain and a no-injection one (denoted by black circles). We obtained an empirical expression for the threshold of injection, given by the expression

$$a_0^* \left(\gamma_0\right) \gtrsim 2.75 \left[1 + \left(\gamma_0/22.\right)^2\right]^{1/2} . \tag{2.2.3}$$

Our simulations show that an injection threshold exists for a cold plasma even for a non-evolving pulse and bubble wake. In particular, for any given bubble phase velocity γ_0, we find that self-injection takes place above a certain the laser intensity a_0^* (i.e., if the bubble size is large enough). The threshold is significantly lower than the one presented in [94] (that predicts the existence of a threshold for self-injection, assuming a simplified analytical expression for the bubble fields) and it is in qualitative agreement with the one presented in [95] at low wake velocities ($\gamma_0 < 60$). For large γ_0, we found that a_0^* grows linearly as $\sim \gamma_0/8$, so, as expected, self-injection does not occur in the ultra-relativistic limit $\gamma_0 \to \infty$ [99].

The threshold condition for self-injection can also be rewritten as a condition on the laser power:

$$P^* \left(\gamma_0\right) / P_c \simeq 2.6 \left[1 + \left(\frac{\gamma_0}{22.}\right)^2\right]^{3/2}$$

For instance, for a plasma with density $n_0 = 3 \cdot 10^{18} cm^{-3}$, the minimum power required for self-injection is such that $P = P_c \cdot (4 \sim 6)$. The calculation is in agreement with experimental results.

The injection threshold was also investigated using the integration of the test particles

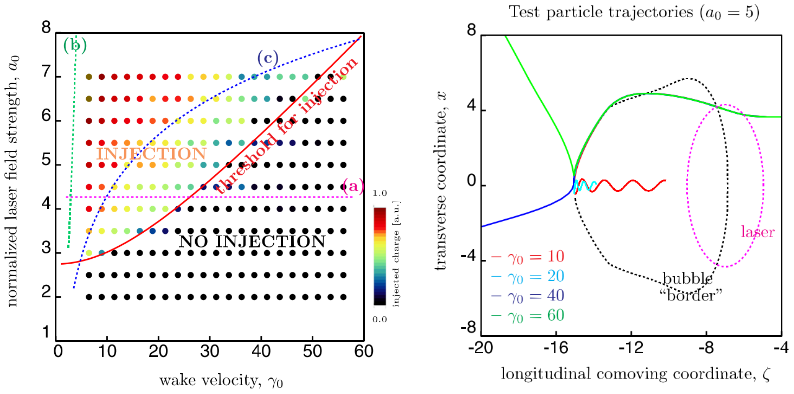


Figure 2.2.1.: Left box: Injection threshold and amount of self-injected charge (represented by color) for different values of the wake velocity γ_0 and laser field strength a_0. The solid red line is the empirical condition for self injection Eq 2.2.3. (a),(b),(c) are the threshold conditions respectively given in references [56], [94] and [95]. Right box: Test particle trajectories for different values of the wake velocity relativistic factor, with $a_0 = 5$, $L = 2$ and $w_0 = 2\sqrt{a_0}$. Phys. Plasmas 20, 103108 (2013)

equations of motion, Eq 2.2.1, using the wakefield map computed with INF&RNO. Figure 2.2 (right) shows different particle trajectories for $a_0 = 5$ and wake velocities $\gamma_0 = 10, 20, 40, 60$. No injection is observed for $\gamma_0 = 40, 60$, while the trajectories for $\gamma_0 = 10, 20$ feature trapping and betatron motion. The injection threshold obtained with this method is consistent with fully self-consistent INF&RNO simulations.

The analysis of the transverse phase space (Fig 2.2) at injection shows an inverse correlation for $\tilde{x}$ and $\tilde{p}_x$: the injection momentum tends to be higher (lower) if injection happens on-axis (off-axis). Furthermore, the phase space area (spread) of the phase space at injection grows with the inverse of the wake phase velocity (for fixed a_0) and with a_0 (for fixed γ_0). The condition 2.2.2 was also be verified to hold at injection using test-particle simulations.

2.3. Evolution of the Phase Velocity of a Bubble Wake generated by an evolving laser driver

In the previous section, the threshold for self-injection has been derived in the case of a non-evolving wake propagating at a constant speed, comoving with a non-evolving driver.

If the driver evolves (due to diffraction, self-focusing, plasma wave guiding, self-

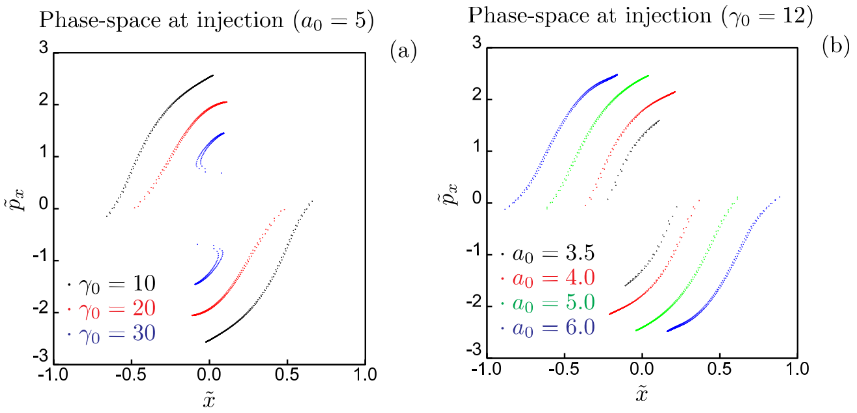


Figure 2.2.2.: Test particles transverse phase space at injection, for fixed wake velocity ($\gamma_0 = 12$ left) and laser intensity ($a_0 = 5$ right). The other laser-plasma parameters are $L = 2$ and $w_0 = 2\sqrt{a_0}$. Phys. Plasmas 20, 103108 (2013)

steepening, depletion, etc.) the bubble wake velocity is determined by the driver evolution, and is no longer equal to the driver velocity. In particular, it can be different from the driver group velocity (as in the weakly relativistic regime in the previous chapter).

In figure 2.3, we show the temporal evolution of the laser group velocity (red line) and the wake phase velocity, measured at the center (blue line) and at the back of the wake, for laser-plasma parameters $a_0 = 4.5$, $w_0 = 2\sqrt{a_0}$, $k_0/k_p = 90$, $L = 2$, with the laser focused at the entrance of the plasma slab. The wake phase velocities have been measured by tracking the position of the longitudinal field zero crossing points throughout the simulation.

The magenta line in figure 2.3(a) is the 1D theory prediction for the phase velocity of the back of the wake in the limit $a_0 \gg 1$ and, in the early stage of laser-plasma interaction. The 1D theory includes the effects of pulse steepening and redshifting (depletion) and predicts $\gamma_{0b}^{(1D)} = 0.45\omega_0/\omega_p = 40.5$ [7], but we expect the actual velocity of the wake to be lower than the 1D result, as slice-dependent plasma wave guiding and the transverse evolution of the laser driver due to self-focusing affect the laser intensity profile and hence the shape of the wake.

In 3D, an analytical theory of the nonlinear wake phase velocity is lacking. The linear theory, valid for $a_0 \ll 1$, predicts a constant value $\gamma_b^{(linear)} = \gamma_{laser}^{(linear)} = \omega_0/\omega_p$ (black dashed line in figure 2.3 (a)). The analytical result (eq 1.4.5) for the weakly relativistic regime $a_0 < 0.5$ fails to provide a good approximation because the peak laser normalized potential $a_0 = 4.5$ considered for this example is far beyond the scope of the model. In Ref. [56], the constant value $\gamma_b^{(3D)} = \omega_0/\sqrt{3}\omega_p \simeq 52$ (green dashed line in figure 2.3 (a)) is proposed by using PIC simulations in the bubble regime.

From our simulation, we observe that the wake velocity is, as expected, lower than the one of the driver and lower than the linear theory prediction. During the bubble formation $\tau < 100$, the wake phase velocity exhibits large fluctuation. Even once the wake is formed, its velocity continues to evolve and it is determined mainly by the laser evolution resulting from the competition between laser self-focusing/diffraction, plasma wave guiding, self-steepening, and frequency redshifting. More specifically, the wake velocity, measured at the center or at the back of the bubble, during its evolution reaches a minimum value of $\gamma_b \simeq 18 - 25$, much lower than the laser driver $\gamma_{laser} \simeq 123$ (red line in figure 2.3 (a)).

If the wake velocity evolution is slow enough (i.e., the velocity does not change too much over the time a plasma particle interacts with the bubble wake) we can, at any time, use Eq. 2.2.3 to determine if self-injection will occur, evaluating it using the instantaneous values of wake velocity and peak normalized field strength. The cyan dashed line in figure 2.3 (a) is the minus wake velocity $\gamma_b^*(\tau)$ compatible with self-injection, computed using Eq. 2.2.3 and the dynamic peak normalized field strength $a_0(\tau)$ (shown figure 2.3 (b)).

We expect self-injection to happen if the actual bubble phase velocity $\gamma_b(\tau)$ measured at the back (where injection takes place) of the bubble is lower than the threshold value $\gamma_b^*(\tau)$.

According to figure 2.3 (a), the actual phase velocity is lower than the threshold value for $150 \lesssim \tau \lesssim 500$ and self-injection mainly occurs, as predicted, within this interval., as can be seen in figure 2.3 (c), showing the distribution of self-injected electrons as a function of their initial longitudinal coordinate.

2.4. Empirical law for the Minimum Value of the Bubble Wake Phase Velocity

Since self-injection occurs when the phase velocity is low, the scaling of the minimum value of bubble phase velocity at the center of the wake, γ_0^{min}, appears to be a critical parameter for estimating the self-injection threshold for an evolving driver. In order

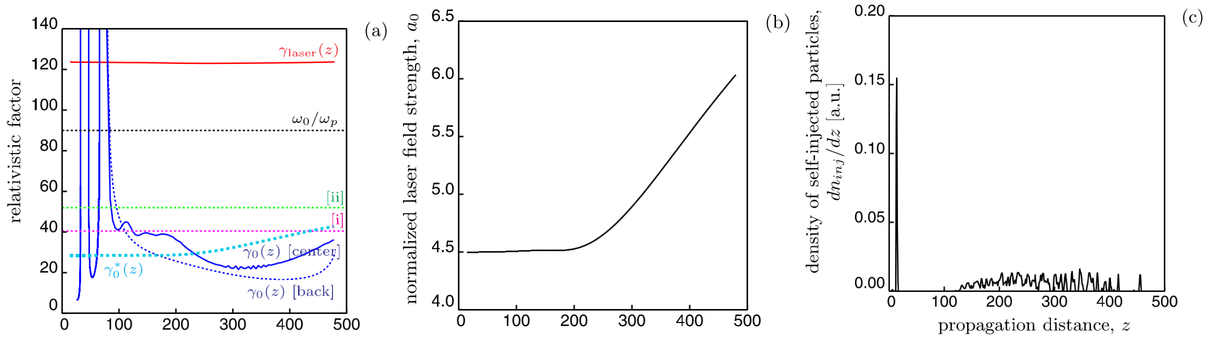


Figure 2.3.1.: (a)(b)(c). Temporal evolution of the wake velocities and other observable resulting from a fully consistent simulation with laser-plasma parameters $a_0 = 4.5$, $w_0 = 2\sqrt{a_0}$, $k_0/k_p = 90$, $L = 2$.
(a) Evolution of the laser group velocity (red line); Wake phase velocity measured at the center and at the back of the bubble wake (blue solid and blue dashed lines); Linear theory prediction for wake velocity (black dashed line); 1D wake velocity [7]([i], magenta dashed line); 3D wake velocity proposed in Ref. [56]; Maximum wake velocity compatible with injection Eq. 2.2.3 (cyan dashed line).
(b) Normalized laser field strength $a_0(\tau)$.
(c) Distribution of self-injected electrons as a function of their initial longitudinal coordinate. As expected, self-injection occurs when the wake velocity measured at the back is lower than the injection threshold Eq. 2.2.3 computed with $a_0(\tau)$.
Phys. Plasmas 20, 103108 (2013);

to characterize γ_0^{min}, we run fully consistent numerical simulations, changing the background plasma density and the laser intensity.

We found that, if $a_0 > 2$, the minimum value of the phase velocity is independent from a_0, even though the details of the phase velocity evolution depend on laser intensity.

The scaling of γ_0^{min} with the plasma background density is shown in figure 2.4, where we plot the values of the minimum wake velocity, measured in a set of simulations with different plasma densities ($10 < k_0/k_p < 150$) . An empirical fit of the minimum bubble velocity is given by the simple formula (red dashed line in figure):

$$\gamma_0^{min} \simeq 2.4 \cdot \sqrt{\frac{k_0}{k_p}} \tag{2.4.1}$$

So far, our study assumes a Gaussian laser driver. As a consequence of transverse

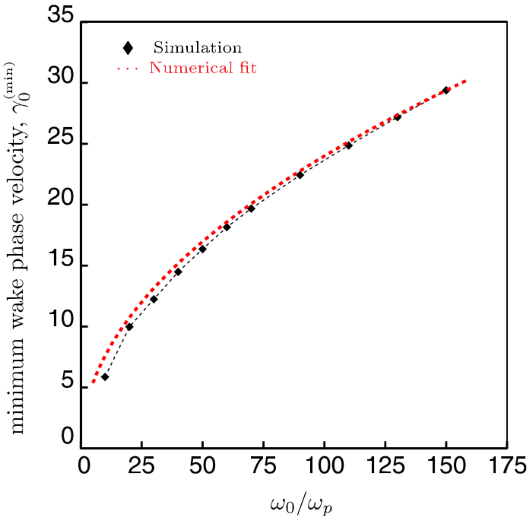


Figure 2.4.1.: Scaling of the minimum of the wake velocity measured at the center of the bubble wake γ_0^{min}, as a function of plasma frequency ω_0/ω_p. The laser is an initially gaussian pulse with $L = 2$, $w_0 = 2\sqrt{a_0}$ and $a_0 = 4.5$ focused at the beginning of the plasma, and its evolution is simulated self-consistently using the envelope equation. The dashed line is the empirical fit Eq 2.4.1. It is found in simulations that the minimum value of the phase velocity is independent from a_0. Phys. Plasmas 20, 103108 (2013)

laser dynamics, its intensity profile evolves towards a "conical" shape (narrower towards the back). Furthermore, the laser shape is modified by the depletion and self-steepening processes. Any pulse shape modification affects the particle orbits and the shape of the wake, since they are determined by the ponderomotive force. The effect of the laser shape on the self-injection physics will be subject of future investigations.

Part II.

Computational methods

3. The particle-in-cell method

3.1. Phase space representation

The most complete ("full") physical model for studying laser-plasma interactions is the Vlasov equation, providing a 6D phase space kinetic description of the plasma, coupled with the Lorentz force, the relativistic equations of motion and Maxwell equations for Electrodynamics.

The most straightforward and complete approach to model numerically a system described by Maxwell-Vlasov equations consists in computing, for each time, the phase space distribution $f_j(\mathbf{x}, \mathbf{p}, t)$, discretized on a grid. In a full three-dimensional model, the plasma phase space is six-dimensional. Thus, the number of grid points (the memory required for the execution of the simulation), scales as n^6, n being the linear dimension of the discrete grid.

This memory requirements are far beyond the actual technology limits, for example taking a meaningful grid size, let's say $n = 1024$, the memory required would be $\sim 10^{18} Gb$.

It is therefore necessary to use a "compressed" representation of the discretized $f_j(\mathbf{x}, \mathbf{p}, t)$. A method that use a very sparse phase space representation is the so-called *particle-in-cell method* [39]. It decomposes the f_j distribution into the sum of contributions coming from a finite N_{p_j} set of computational macro-particles, or quasiparticles. Their trajectories are followed in the phase space in a lagrangian manner, while the electromagnetic fields are discretized on a spatial grid, with grid spacing $\Delta\mathbf{x}$.

The macro particles are not point like charges, they are represented by a density function which is extended in space so that they can be considered as a smooth cloud of charge , in order to smooth out the numerical noise. The support of these function has a size of the order of the grid cell size. Whereas in the configuration space the numerical particles are defined by a finite extension, in the momentum space they are point-like (they have definite momentum).

The interaction of the particles with the field grids, which complete the description of the dynamics, is achieved by processes of *interpolation* and *deposition*. The interpolation and deposition processes, being the support of the quasi-particle density function compact involve only a small number of grid *cells*, the ones overlapping with the particle's finite shape.

The Vlasov equation and the equations of motion read:

$$(\partial_t + \dot{\mathbf{x}}\cdot\partial_{\mathbf{x}} + \dot{\mathbf{p}} \cdot \partial_{\mathbf{p}})\, f_j(\mathbf{x}, \mathbf{p}, t) = 0 \tag{3.1.1}$$

$$\dot{\mathbf{x}} = \frac{\mathbf{p}}{\gamma m}\ ,\ \ \dot{\mathbf{p}} = \mathbf{F}(\mathbf{x}, \mathbf{p}, t) \tag{3.1.2}$$

The PIC approach consists in discretizing the phase space density function using a finite, approximated, sum:

$$f_j(\mathbf{x}, \mathbf{p}, t) = f_{0j} \sum_{n=0}^{N_{p_j}-1} g\left(\mathbf{x} - \mathbf{x}_n(t)\right) \delta\left(\mathbf{p} - \mathbf{p}_n(t)\right),$$

in which, f_0 is a normalization factor, $\mathbf{x}_n(t)$ is the trajectory of the *n-th* macro-particle and $\mathbf{p}_n(t)$ is its momentum.

The function $g(\mathbf{x})$ is the macro-particle *shape function.*

The shape function is used as a convolution kernel and it is assumed to have δ-like properties (from which follows $f_0 = \frac{1}{N_p}$):

$$\begin{cases} \int g(\mathbf{x} - \mathbf{x}_n) d\mathbf{x} = 1 \\ \int \partial_{\mathbf{x}} g(\mathbf{x} - \mathbf{x}_n) d\mathbf{x} = \mathbf{0} \end{cases} \tag{3.1.3}$$

$g(\mathbf{x})$ describes the macro-particle spatial extension in space and it is useful for reducing the numerical noise arising from interpolation and deposition processes, which would arise if a δ-function was used instead. The meaning of $g(\mathbf{x})$ is evident considering the expression for the charge density, which becomes:

$$\rho(\mathbf{x}, t) = \textstyle\sum_j Q_j \int f_j(\mathbf{x}, \mathbf{p}, t) d\boldsymbol{p}$$
$$\rho(\mathbf{x}, t) = \textstyle\sum_{j,n} q_j g(\mathbf{x} - \mathbf{x}_n)$$

, whereas the electrical current can be defined as:

$$\mathbf{j}(\mathbf{x},t)=\sum_{j,n}\mathbf{v}_n q_j g(\mathbf{x}-\mathbf{x}_n)$$

Rewriting the Vlasov equation 3.1.1, using this discretized phase space discretization and equation 3.1.2, one gets:

$$\begin{cases}\partial_t f & = -f_0\sum_n^{N_p}\left\{\left[\partial_{\mathbf{x}}g\left(\mathbf{x}-\mathbf{x}_n(t)\right)\cdot\dot{\mathbf{x}}_n(t)\right]\delta\left(\mathbf{p}-\mathbf{p}_n(t)\right)\right. \\ & \left. +g\left(\mathbf{x}-\mathbf{x}_n(t)\right)\left[\partial_{\mathbf{p}}\delta\left(\mathbf{p}-\mathbf{p}_n(t)\right)\cdot\dot{\mathbf{p}}_n(t)\right]\right\} \\ \dot{\mathbf{x}}\cdot\partial_{\mathbf{x}}f & = f_0\sum_n^{N_p}\left\{\frac{p_n(t)}{\gamma m}\qquad\cdot\left[\partial_{\mathbf{x}}g\left(\mathbf{x}-\mathbf{x}_n(t)\right)\right]\delta\left(\mathbf{p}-\mathbf{p}_n(t)\right)\right\} \\ \dot{\mathbf{p}}\cdot\partial_{\mathbf{p}}f & = f_0\sum_n^{N_p}\left\{\mathbf{F}\left(z,p_n(t),t\right)\cdot g\left(\mathbf{x}-\mathbf{x}_n(t)\right)\left[\partial_{\mathbf{p}}\delta\left(\mathbf{p}-\mathbf{p}_n(t)\right)\right]\right\}\end{cases} \tag{3.1.4}$$
$$f_0\sum_n^{N_p}\left(-\dot{\mathbf{x}_n}\cdot g_n'\delta_n-\dot{\mathbf{p}}_n g_n\cdot\delta_n'+\frac{\mathbf{p}_n}{\gamma m}\cdot g_n'\delta_n+\mathbf{F}\left(z,p_n(t),t\right)g_n\cdot\delta_n'\right)=0$$

Integrating in the momentum space and using the delta function properties, one has:

$$\sum_n^{N_p}\left(-\dot{\mathbf{x}}_n+\frac{\mathbf{p}_n}{\gamma m}\right)\partial_{\mathbf{x}}g\left(\mathbf{x}-\mathbf{x}_n(t)\right)=0,\ \forall\mathbf{x}\rightarrow\dot{\mathbf{x}}_n=\frac{\mathbf{p}_n}{\gamma m}$$

Being $\overline{\mathbf{F}}_n$ the spatial average of the external force field acting on the $n-th$ macroparticle $\mathbf{F}(\mathbf{x},\mathbf{p}_n,t)$ evaluated over the shape function $g(\mathbf{x})$:

$$\overline{\mathbf{F}}_n(\mathbf{x},\mathbf{p},t)=\int g(\mathbf{x}-\mathbf{x}_n)\mathbf{F}(\mathbf{x},\mathbf{p}_n,t)dx$$

, integrating on dz, and using the delta-like properties 3.1.3 of $g(\mathbf{x})$, one gets:

$$\sum_n^{N_p}\left(-\dot{\mathbf{p}}_n+\overline{\mathbf{F}}_n\right)\partial_{\mathbf{p}}\delta(\mathbf{p}-\mathbf{p}_n)=0,\ \forall\mathbf{p}=0\rightarrow\dot{\mathbf{p}}_n=\overline{\mathbf{F}}_n$$

The *particle-in-cell* method, therefore, reduces the computational complexity required for the evolution of a six-dimensional phase space grid to a system of $2N_p$ (for each species) equations of motion, coupled with the proper equations (in our case for the e.m. fields) that close the system giving an expression for the external force field $\mathbf{F}$.

3.1.1. Passes of an electromagnetic PIC code and numerical parameters of a laser plasma interaction simulation

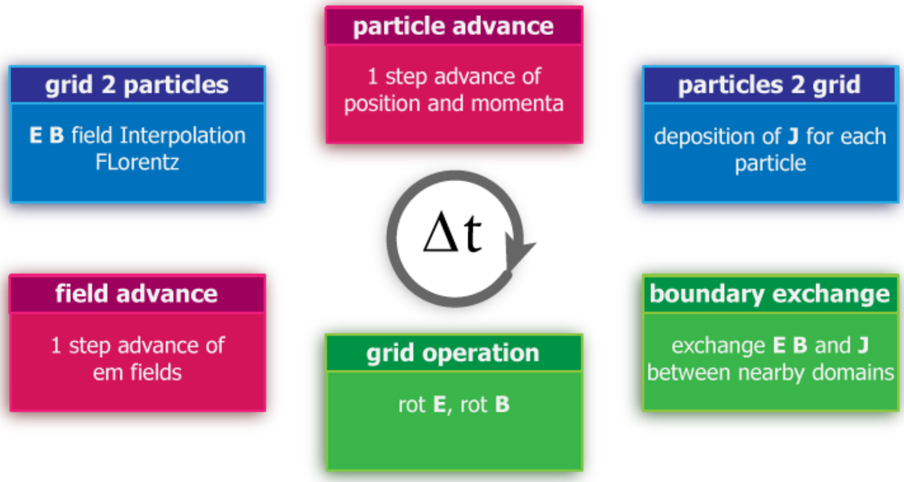
Dealing with charged particles, the physical description of the problem is closed by the Maxwell equations for the electromagnetic fields, which are coupled with the particle motion in a bidirectional way (by the Lorentz force and by the evaluation of charge and current densities).

The passes of an integration cycle of an electromagnetic PIC code are the following:

1. *Time advancement of macro-particles momentum and position* $\mathbf{p}, \mathbf{x}$, using the obtained equations of motion and the Lorentz force. The fields are interpolated from the **E**,**B** grids.

2. *Deposition* (spatial average on a discrete grid) of the external field quantities needed in Maxwell equations, ρ and $\mathbf{j}$.

3. *Time advancement of electromagnetic fields* **E**,**B**, discretized on spatial grids, (see subsection 3.2.1), using Maxwell equations and the quantities computed in step 2 as external sources.


particle advance
1 step advance of position and momenta
grid 2 particles
E B field Interpolation FLorentz
particles 2 grid
deposition of J for each particle
Δt
field advance
1 step advance of em fields
boundary exchange
exchange E B and J between nearby domains
grid operation
rot E, rot B

The critical parameter of a simulation is the grid cell size Δx. The integration timestep Δt is related to Δx by the Courant condition. It is a condition required for the stability of the explicit integration schemes for the Maxwell PDEs, reading $\Delta t \leq c\Delta x$, where the constant c depends on the set of algorithms used.

Furthermore, the size Δx must be small enough to resolve with enough grid points the typical lengths of the considered system.

In the case of a system of electromagnetic waves interacting with a plasma, these are:

- λ_{em}, the wavelength of the electromagnetic waves
- $\lambda_{sd} = c/\omega_{pe}$, the plasma skin depth

The smallest of the two length scales must be resolved with enough grid points. The two length scales correspond to two mutually-exclusive regimes:

- Sovracritical regime: $\omega \ll \omega_{pe} \to \lambda_{sd} \ll \lambda_{em}$, the λ_{sd} must be resolved, having the other one resolved as well.
- Underdense plasma regime: $\lambda_{em} \ll \lambda_{sd}$, the λ_{em} must be resolved, having the other one resolved as well.

  The *laser envelope* approximation (see section) may come to help in this case, requiring only the much larger scale λ_{sd} to be resolved, allowing the use much smaller grid sizes.

Another critical parameter of a PIC simulation the number of macro-particles per cell, sampling the local phase space. Approximating the phase space distribution as a finite decomposition of a too small number of spatially extended macro-particles, can cause some regions of the phase space to be represented with not enough detail and the introduction of a statistical noise effect. The amplitude of the latter effect scales approximately with $\sqrt{N_{part\,per\,cell}}$ ([39]).

3.1.2. Interpolation and deposition using shape functions

The spatial averaging needed for interpolation and deposition processes is defined using the particles' shape function g.

3.1.2.1. Force interpolation

The average (interpolated) force acting on a particle is defined as, being $\mathbf{F}(\mathbf{x}, \mathbf{p}_n, t) = q\left(\mathbf{E}(\mathbf{x}) + \frac{\mathbf{p_n} \times \mathbf{B}(\mathbf{x})}{m\gamma c}\right)$:

$$\overline{\mathbf{F}}_{\mathbf{n}}(\mathbf{x}_n, \mathbf{p}_n, t) = \int g(\mathbf{x} - \mathbf{x}_n)\mathbf{F}(\mathbf{x}, \mathbf{p}_n, t) d\mathbf{x}$$

Indexing the grid cells, with characteristic function $\chi_\imath$, with the multidimensional index $\imath = (i, , j, k)$ it is possible to *decompose* the above integral average into a finite sum of single cell averages:

$$\overline{\mathbf{F}}_\mathbf{n}(\mathbf{x}_n, \mathbf{p}_n, t) = \sum_{\iota \in G} \int_{X_\iota} g(\mathbf{x} - \mathbf{x}_n)\mathbf{F}(\mathbf{x}, \mathbf{p}_n, t) d\mathbf{x}$$

The force is given by the fields which are discretized in such a way that they take a single, constant, value per cell $\mathbf{E}_\mathbf{i}, \mathbf{B}_\imath$. It is therefore possible to write the cell-integrals as function of the particle position only:

$$\overline{\mathbf{F}}_n(\mathbf{x}_n, \mathbf{p}_n, t) = \sum_{\iota \in G} F_\imath(\mathbf{p}_n, t) \int_{X_\iota} g(\mathbf{x} - \mathbf{x}_n) d\mathbf{x}$$

, or, introducing the *shape factors* (for a particle whose position is $\mathbf{x}_n$) $S_\imath(\mathbf{x_n}) = \int_{X_\iota} g(\mathbf{x} - \mathbf{x}_n) d\mathbf{x}$:

$$\overline{\mathbf{F}}_n(\mathbf{x}_n, \mathbf{p}_n, t) = \sum_{\iota \in G} \mathbf{F}_\imath(\mathbf{p}_n, t) S_\imath(\mathbf{x}_n)$$

For the shape factors the property $\sum_\imath S_\imath(\mathbf{x}) = 1$ hold true.

3.1.2.2. Charge and current deposition

In order to evaluate the current and electrical charge density, discretized on a grid, it is necessary to "deposit" the macro-particle charge on the grid nodes. Being $\rho(\mathbf{x})$ defined as $\rho(\mathbf{x}) = \sum_n q \, g(\mathbf{x} - \mathbf{x}_n)$, then:

$$\begin{aligned} \rho_\imath &= \frac{\int_{\chi_\imath} \rho(\mathbf{x}) d\mathbf{x}}{\int d\mathbf{x}_{\chi_\imath} = V_\imath} = \int_{\chi_\imath} \left[\sum_n q g(\mathbf{x} - \mathbf{x}_n) \right] d\mathbf{x}/V_\imath \\ &= \sum_n q \left[\int_{\chi_\imath} g(\mathbf{x} - \mathbf{x}_n) d\mathbf{x} \right] /V_\imath = \frac{1}{V_\imath} \sum_n q S_\imath(\mathbf{x}_n) \end{aligned}$$

3.1.2.3. Common shapefunctions and shapefactors

For a regular 3D cartesian grid, with grid cells sized $\Delta\mathbf{x} = (\Delta x, \Delta y, \Delta z)$ centered in the point $\mathbf{x}_{\imath=(i,j,k)} = (x_i, y_j, z_k)$, $x_i = x^0 + \Delta x \cdot i$, it is useful to introduce the *centered and normalized* shapefunctions and shapefactors $\tilde{g}(\tilde{\mathbf{x}})$ and $\tilde{S}(\tilde{\mathbf{x}}_\imath)$, defined starting from the cell-centered coordinate system (denoted by $\tilde{}$):

$$\tilde{\mathbf{x}}_\imath = ((x-x_i)/\Delta x, (y-y_j)/\Delta y,\ (z-z_k)/\Delta z)\,,$$

, in which, the shape factors becomes:

$$S(\tilde{\mathbf{x}}_\imath) = S_\imath(\mathbf{x}) = \tilde{S}\left((x-x_i)/\Delta x, (y-y_j)/\Delta y,\ (z-z_k)/\Delta z\right)$$

The equation $S_\imath(\mathbf{x_n}) = \int_{X_\imath} g(\mathbf{x}-\mathbf{x}_n)d\mathbf{x}$, can be recast using the linear change of variables $\mathbf{y} = \Delta\mathbf{y} * \tilde{\mathbf{y}}_\imath + \mathbf{y}_\imath$:

$$\tilde{S}(\tilde{\mathbf{x}}_\imath) = S_\imath(\mathbf{x}) = \int_{\chi i} g\left(\mathbf{y} - (\tilde{\mathbf{x}}_\imath * d\Delta\mathbf{x} + \mathbf{x}_\imath)\right) d\mathbf{y}$$

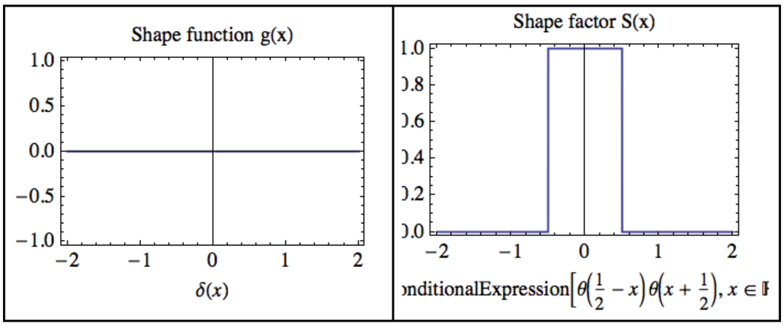
$$\tilde{S}(\tilde{\mathbf{x}}_\imath) = V\int_{\chi} g(\Delta\mathbf{x} * (\tilde{\mathbf{y}}_\imath - \tilde{\mathbf{x}}_\imath))d\tilde{\mathbf{y}}_\imath = \int_{\chi} \tilde{g}(\tilde{\mathbf{y}}_\imath - \tilde{\mathbf{x}}_\imath)d\tilde{\mathbf{y}}_\imath \qquad (3.1.5)$$

where χ is the volume of the box defined by $|\widetilde{x}| < \frac{1}{2}$, $|\widetilde{y}| < \frac{1}{2}$, $|\widetilde{z}| < \frac{1}{2}$ and $\tilde{g}(\tilde{\mathbf{y}}_\imath) \equiv Vg(\Delta\mathbf{x} * \tilde{\mathbf{y}}_\imath)$.

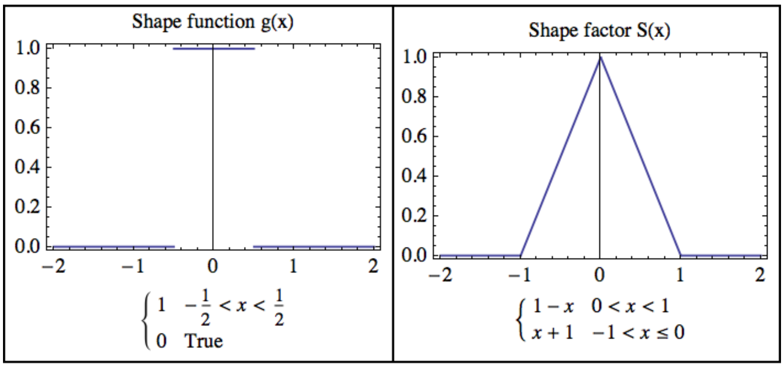
It is natural for the shapefunction to be *separable* in one dimensional components, i.e. $g(\mathbf{x}) = g(x)g(y)g(z)$. By simple integration properties, one has also:

$$S(\mathbf{x}) = S^x(x)S^y(y)S^z(z)$$

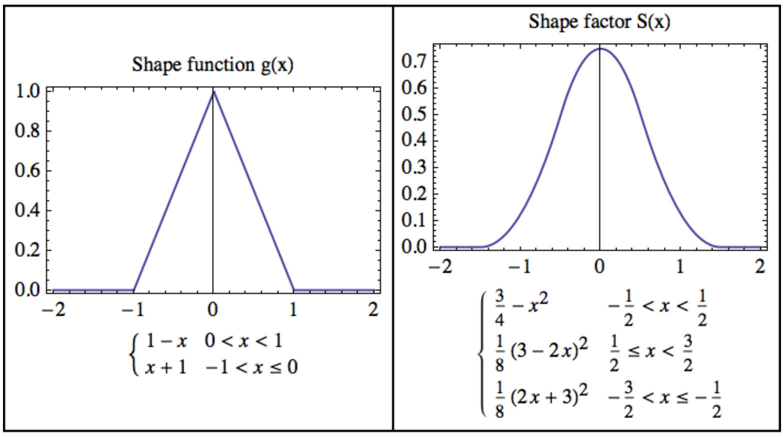
This last relation and equation 3.1.5 allow to compute easily the shape factors for any separable shape function. Some examples of normalized, one-dimensional shape *functions/factors* are (dropping the ~ in the figures) are:



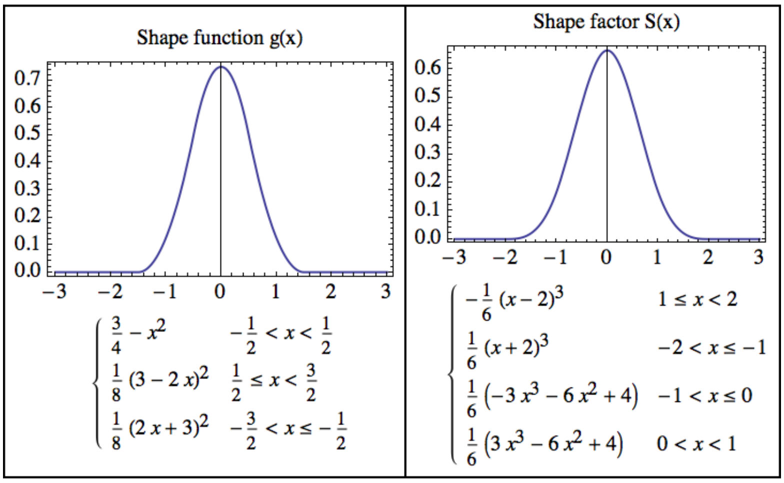
Point-Like



Linear



Quadratic



Cubic

These classical shape functions are defined piecewise on intervals of length Δx (1 in the normalized coordinates system). By definition, the shape factor functions $S(x)$ have the same properties. The intervals $\alpha \in \text{Å}$ of piecewise definition of these $S(\widetilde{x}_i)$ are always of the kind $\alpha = [a_\alpha, a_\alpha + 1]$, and can be identified by their parameter $a \in A$, integral or half-integral.

$$a \leq \widetilde{x}_i \leq a+1$$

Replacing some definitions in the relation above, and applying the floor and ceil function ($\lfloor x \rfloor$and $\lceil x \rceil$) properties, one obtains directly the cell index corresponding to a given piece of function definition (for performance reasons, it is useful to know it in advance):

$$\begin{gathered} a\Delta x \leq x - x^0 - i_a \Delta x \leq (a+1)\Delta x \\ \lceil (x-x_0)/\Delta x - a - 1 \rceil \leq i_a \leq \lfloor (x-x_0)/\Delta x - a \rfloor \end{gathered}$$

$$\forall a : i_a = \lfloor (x-x_0)/\Delta x - a \rfloor = \begin{cases} \lfloor \tilde{x}_0 \rfloor - a \equiv i_0 - a; & a \text{ integer} \\ \lfloor \tilde{x}_0 + \frac{1}{2} \rfloor - a' \equiv i'_0 - a'; & a = a' - \frac{1}{2}\text{halfinteger} \end{cases}$$

The optimized chain of computation reads (’ for the case in which a are half integer):

$$\begin{tikzcd} \forall a \in A \arrow[d] \\ i_a = i_0^{(')} - a^{(')} \arrow[d] \\ \widetilde{x}_{i_0 - a} = \widetilde{x}_{i_0} - a^{(')} \arrow[d] \\ S(\widetilde{x}_{i_0} - a^{(')}) \equiv S_a(\widetilde{x}_{i_0}) \quad \text{optimized} S_a \end{tikzcd}$$

Defining $b \in B$ and $c \in C$ as the analogous, for the y and z directions, of the intervals $a \in A$, one can finally recompose the full 3D interpolation algorithm for a particle in position $\mathbf{x} = (x, y, z)$ (dropping ’):

$$\begin{cases} (i_0, j_0, k_0) & = (\lfloor \tilde{x}_0 \rfloor, \lfloor \tilde{y}_0 \rfloor, \lfloor \tilde{z}_0 \rfloor) = \left(\left\lfloor (x - x^0)/\Delta x \right\rfloor, ..., ... \right) \\ \overline{F} & = \sum_{(abc)} S^x_a(\widetilde{x}_{i_0}) \cdot S^y_b(\widetilde{y}_{j_0}) \cdot S^z_c(\widetilde{z}_{k_0}) \cdot F_{i_0+a,\, j_0+b,\, k_0+c} \end{cases}$$

and the deposition algorithm (of the single particle quantity$F$) :

$$\begin{cases} (i_0, j_0, k_0) & = (\lfloor \tilde{x}_0 \rfloor, \lfloor \tilde{y}_0 \rfloor, \lfloor \tilde{z}_0 \rfloor) = \left(\left\lfloor (x - x^0)/\Delta x \right\rfloor, ..., ... \right) \\ \forall a, b, c : & = S^x_a(\widetilde{x}_{i_0}) \cdot S^y_b(\widetilde{y}_{j_0}) \cdot S^z_c(\widetilde{z}_{k_0}) \cdot F \to \oplus \to F_{i_0+a,\, j_0+b,\, k_0+c} \end{cases}$$

More generally, considering symmetric shape factors $S(\tilde{x})$ with support $supp(S)$ in the interval $[-l, l = \tilde{b} + \Delta x/2]$, the interpolation is computed only on the grid cells for which $S_i(x) \neq 0$, or, equivalently, $S(\tilde{x}_\imath) \neq 0$ holds true:

$$supp(S) = \{-l \leq \tilde{x}_i \leq l\}$$

$$-l \leq (x - x_i)/\Delta x \leq l$$

$$-l\Delta x \leq \left(x - x^0 - i \cdot \Delta x\right) \leq l\Delta x$$

$$(x - x_0)/\Delta x - l \leq i \leq (x - x_0)/\Delta x + l$$

$$\lceil \tilde{x}_0 - l \rceil \leq i \leq \lfloor \tilde{x}_0 + l \rfloor$$

So, the cells interacting with our particle, are the one with i-index in the set I:

$$i \in I = \{\lfloor \tilde{x}_0 - l \rfloor + 1,\ \lfloor \tilde{x}_0 - l \rfloor + 2,\ \lfloor \tilde{x}_0 - l \rfloor + 3, ...,\ \lfloor \tilde{x}_0 + l \rfloor\}$$

$$\#(I) = 2l;$$

Furthermore,

- if l is an integer: $i \in I = \{\lfloor \tilde{x}_0 \rfloor - l + 1,\ ..., \lfloor \tilde{x}_0 \rfloor - l + (l - 1), \lfloor \tilde{x}_0 \rfloor,\ ...,\ \lfloor \tilde{x}_0 \rfloor + l\}$, e.g. $l = 1 :\ i \in I = \{\lfloor \tilde{x}_0 \rfloor, \lfloor \tilde{x}_0 \rfloor + 1\}$
- if $l = m - \frac{1}{2}$ is an half-integer: $i \in I = \{\left\lfloor \tilde{x}_0 + \frac{1}{2} \right\rfloor - m + 1, ..., \left\lfloor \tilde{x}_0 + \frac{1}{2} \right\rfloor, ..., \left\lfloor \tilde{x}_0 + \frac{1}{2} \right\rfloor + m - 1\}$ e.g. $l = \frac{3}{2} : m = 2,\ i \in I = \{\left\lfloor \tilde{x}_0 + \frac{1}{2} \right\rfloor - 1, \left\lfloor \tilde{x}_0 + \frac{1}{2} \right\rfloor, \left\lfloor \tilde{x}_0 + \frac{1}{2} \right\rfloor + 1\}$

3.2. The “standard” second-order PIC: leapfrog and FDTD

The “standard”, second-order PIC scheme, uses the leapfrog scheme for advancing particles’ momentum and position and the FDTD method for solving Maxwell equations.

3.2.1. Solving the Maxwell equations numerically using the Yee Lattice

The temporal evolution of the electromagnetic fields, in presence of an electrical current $\mathbf{j}$ (generated by the plasma particles’ velocities in our case), is completely determined, given the proper initial and boundary conditions, by the Maxwell–Faraday and Ampère-Maxwell laws:

$$\begin{cases} \frac{\partial \mathbf{E}}{\partial t} &= +c\nabla \times \mathbf{B} - 4\pi \mathbf{j} \\ \frac{\partial \mathbf{B}}{\partial t} &= -c\nabla \times \mathbf{E} \end{cases} \tag{3.2.1}$$

These equations form a system of two first order partial differential equations, which can be integrated numerically in time using finite difference methods. First of all, the fields are discretized on a finite grid and the spatial differentiation operators are approximated by finite differences.

The finite difference approximation $\Delta_{\mathbf{h}}[f](\mathbf{x})$ of a linear differential operator Δ is a linear function of the field values at the locations $\mathbf{x} + \mathbf{h} * \mathbf{k}$, being $\mathbf{k}$ a vector of signed integers, $\mathbf{h}$ the discretization step size vector, and * the component-by-component product:

$$\Delta_{\mathbf{h}}[f](\mathbf{x}) = \sum_{k \in \kappa} \mu_k^{\Delta} f(\mathbf{x} + \mathbf{h} * \mathbf{k}) \cong \Delta f(\mathbf{x})$$

Indexing the grid points G with a multi-dimensional index $\imath = (i, j, k, ...) \in G$, the finite difference, calculated at the grid point $\imath$, is:

$$\begin{cases} (\Delta_h[f])_\imath = \sum_{\jmath \in G} \mu_{\imath - \jmath} f_\jmath \equiv \mu_{\imath\jmath}^{\Delta} f_\jmath \\ \mu_{\imath\jmath}^{\Delta} \equiv \mu_{\imath-\jmath}^{\Delta} \end{cases}$$

For the curl, one has:

$$\nabla \times \mathbf{E} = \begin{pmatrix} \partial_y E_z - \partial_z E_y \\ \partial_z E_x - \partial_x E_z \\ \partial_x E_y - \partial_y E_x \end{pmatrix} \to (\nabla \times \mathbf{E}_\imath) = \begin{pmatrix} \mu_{\imath\jmath}^{\partial_y} E_{z_\jmath} - \mu_{\imath\jmath}^{\partial_z} E_{y_\jmath} \\ ... \\ ... \end{pmatrix} \equiv \overline{\mu}_{\imath\jmath}^{(\nabla\times)} \mathbf{E}_j$$

This way, the PDE becomes a system of *ordinary* differential equations involving the values of the fields defined at the grid points locations $\imath$.

$$\begin{cases} \left(\frac{\partial \mathbf{E}}{\partial t}\right)_\imath & = +c\left(\overline{\mu}_{\imath\jmath}^{(\nabla\times)} \mathbf{B}_j\right)_\imath - 4\pi \mathbf{j}_\imath \\ \left(\frac{\partial \mathbf{B}}{\partial t}\right)_\imath & = -c\left(\overline{\mu}_{\imath\jmath}^{(\nabla\times)} \mathbf{E}_j\right)_\imath \end{cases}$$

It is now possible to integrate iteratively and numerically the equations, using methods like Runge-Kutta, or directly expressing the temporal differentiation operators as finite differences, and then solving the resulting system for the values at later times.

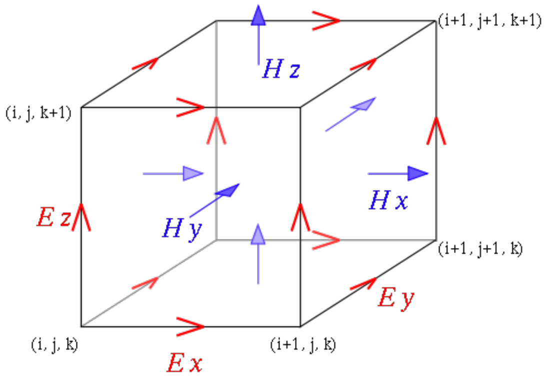
Depending on the fields' discretization geometry and the physical quantities that is more important to conserve, various choices can be made between numerical derivative and integration schemes:

- The *time* integration is usually computed explicitly (the appropriate Courant–Friedrich's–Lewy on the integration step must be therefore satisfied for having stability), using Runge-Kutta, forward (Euler) or centered (leapfrog) schemes.
- The *spatial* derivatives (curls) can be discretized using centered (second-order accurate in space), higher order (five point stencil), upwind (for certain geometries and for certain purposes) or even compact (see [3]) schemes.

A very popular integration scheme, which is second-order accurate in *space and time*, is the Finite-Difference Time-Domain method, or FDTD (see [100]). It manages to discretize both time and all of the space derivatives using centered differences of the kind $\Delta x \cdot f'(x) \simeq f(x + \frac{1}{2}\Delta x) - f(x - \frac{1}{2}\Delta x)$, which are practically more accurate than the standard $2\Delta x \cdot f'(x) \simeq f(x + \Delta x) - f(x - \Delta x)$, being $f'(x) = \partial_x f(x)$:

- The electric and magnetic fields must be defined at staggered *time* positions ($\mathbf{E}^n, \mathbf{B}^{n+1/2}$), and the time-advance iteration is obtained with a leapfrog step: the PDE system has a symplectic structure. This integration scheme allows for numerical-energy-dissipation-free wave propagation.

- *Spatially*: the different field components are stored for different grid locations, on a *Yee Lattice*. A 3D computational domain is split into cubical voxels. The components of **E** are stored for the edges (in the corresponding directions), while the components of **B** for the face centers of the cube. The numerical derivatives composing the curl operations, can all be expressed using central differences, as we can see clearly in the following figure (from wikipedia):



For an uniform 3D cartesian discretization of the fields (the grid spacings are dx, dy, dz), the FDTD scheme can be therefore written explicitly as (the notation used is $F^n_{i,j,k} \equiv F(idx, jdy, kdz, ndt)$):

$$\begin{cases} \frac{E_{x_{i+1/2,j,k}}^{n+1}-E_{x_{i+1/2,j,k}}^{n}}{dt} & = c\frac{B_{z_{i+1/2,j+1/2,k}}^{n+1/2}-B_{z_{i+1/2,j-1/2,k}}^{n+1/2}}{dy} - c\frac{B_{y_{i+1/2,j,k+1/2}}^{n+1/2}-B_{y_{i+1/2,j,k-1/2}}^{n+1/2}}{dz} \\ & -4\pi J_{x_{i+1/2,j,k}}^{n+1/2} \\ \frac{E_{y_{i,j+1/2,k}}^{n+1}-E_{y_{i,j+1/2,k}}^{n}}{dt} & = c\frac{B_{x_{i,j+1/2,k+1/2}}^{n+1/2}-B_{x_{i,j+1/2,k-1/2}}^{n+1/2}}{dz} - c\frac{B_{z_{i+1/2,j+1/2,k}}^{n+1/2}-B_{z_{i-1/2,j+1/2,k}}^{n+1/2}}{dx} \\ & -4\pi J_{y_{i,j+1/2,k}}^{n+1/2} \\ \frac{E_{z_{i,j,k+1/2}}^{n+1}-E_{z_{i,j,k+1/2}}^{n}}{dt} & = c\frac{B_{y_{i+1/2,j,k+1/2}}^{n+1/2}-B_{y_{i-1/2,j,k+1/2}}^{n+1/2}}{dx} - c\frac{B_{x_{i,j+1/2,k+1/2}}^{n+1/2}-B_{x_{i,j-1/2,k+1/2}}^{n+1/2}}{dy} \\ & -4\pi J_{z_{i,j+1/2,k}}^{n+1/2} \\ & \\ \frac{B_{x_{i,j+1/2,k+1/2}}^{n+1/2}-B_{x_{i,j+1/2,k+1/2}}^{n-1/2}}{dt} & = -c\frac{E_{z_{i,j+1,k+1/2}}^{n}-E_{z_{i,j,k+1/2}}^{n}}{dy} + c\frac{E_{y_{i,j+1/2,k+1}}^{n}-E_{y_{i,j+1/2,k}}^{n}}{dz} \\ \frac{B_{y_{i+1/2,j,k+1/2}}^{n+1/2}-B_{y_{i+1/2,j,k+1/2}}^{n-1/2}}{dt} & = -c\frac{E_{x_{i+1/2,j,k+1}}^{n}-E_{x_{i+1/2,j,k}}^{n}}{dz} + c\frac{E_{z_{i+1,j,k+1/2}}^{n}-E_{z_{i,j,k+1/2}}^{n}}{dx} \\ \frac{B_{z_{i+1/2,j+1/2,k}}^{n+1/2}-B_{z_{i+1/2,j+1/2,k}}^{n-1/2}}{dt} & = -c\frac{E_{y_{i+1,j+1/2,k}}^{n}-E_{y_{i,j+1/2,k}}^{n}}{dx} + c\frac{E_{x_{i+1/2,j+1,k}}^{n}-E_{x_{i+1/2,j,k}}^{n}}{dy} \end{cases}, \tag{3.2.2}$$

or, defining the operators:

$$\begin{cases} \nabla^{+} f_{ijk} & = \left(\frac{f_{i+1,j,k}-f_{i,j,k}}{dx}, \frac{f_{i,j+1,k}-f_{i,j,k}}{dy}, \frac{f_{i,j,k+1}-f_{i,j,k}}{dz} \right) \\ \nabla^{-} f_{ijk} & = \left(\frac{f_{i,j,k}-f_{i-1,j,k}}{dx}, \frac{f_{i,j,k}-f_{i,j-1,k}}{dy}, \frac{f_{i,j,k}-f_{i,j,k-1}}{dz} \right) \end{cases}, \tag{3.2.3}$$

as:

$$\begin{cases} \nabla^{-} \cdot \mathbf{E}^{n} & = 4\pi\rho^{n} \\ \nabla^{+} \cdot \mathbf{B}^{n+1/2} & = 0 \\ & \\ \frac{\mathbf{E}^{n+1}-\mathbf{E}^{n}}{dt} & = c\nabla^{-} \times \mathbf{B}^{n+1/2} - 4\pi\mathbf{J}^{n+1/2} \\ \frac{\mathbf{B}^{n+1/2}-\mathbf{B}^{n-1/2}}{dt} & = -c\nabla^{-} \times \mathbf{E}^{n} \end{cases}, \tag{3.2.4}$$

with:

$$\begin{cases} \mathbf{E}^n &= \left(E^n_{x_{i+1/2,j,k}}, E^n_{y_{i,j+1/2,k}}, E^n_{z_{i,j,k+1/2}} \right) \\ \mathbf{B}^n &= \left(B^n_{x_{i,j+1/2,k+1/2}}, B^n_{y_{i+1/2,j,k+1/2}}, B^n_{z_{i+1/2,j+1/2,k}} \right) \\ \\ \rho^n &= \rho^n_{i,j,k} \\ \mathbf{J}^n &= \left(J^n_{x_{i+1/2,j,k}}, J^n_{y_{i,j+1/2,k}}, J^n_{z_{i,j,k+1/2}} \right) \end{cases}$$

The Courant condition of this numerical scheme is ([39]):

$$c \cdot dt < 1/\sqrt{\left(\frac{1}{dx}\right)^2 + \left(\frac{1}{dy}\right)^2 + \left(\frac{1}{dz}\right)^2}$$

Physics requires the first two equations (Gauss laws) in 3.2.4 to hold at every time step .

Analytically, if the *electrical charge continuity equation* $\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0$ holds, the EM field evolution PDEs 3.2.1 do automatically enforce Gauss law (if the law was satisfied at the initial time). In fact, applying the divergence to the Maxwell–Faraday Law $\frac{\partial \mathbf{E}}{\partial t} = +c\nabla \times \mathbf{B} - 4\pi \mathbf{j}$, and using the Gauss law itself, one obtains the *electrical charge continuity equation*: if the latter is satisfied so it is the Gauss law.

In a simple PIC simulation, instead, the $\mathbf{J}$ deposition process, which uses finite sized shape functions, introduces and accumulates numerical errors due to cell boundary crossing and charge conservation must be enforced in other ways, as shown in section 3.3.

3.2.2. Boris pusher

For what concerns the quasi-particles' motion, the solutions are computed using *Boris method*, which is a second-order, leapfrog-like, method that perfectly matches with the previously discussed FDTD and can be used, in general, for integrating the equations of motion of relativistic particles in an external electromagnetic field.

For a relativistic particle of mass m, the second Newton's equation reads,

$$\mathbf{p} = \gamma m \mathbf{v}, \; \mathbf{F} = \frac{d\mathbf{p}}{dt}$$

Introducing $\mathbf{u} = \gamma \mathbf{v}/c = \mathbf{p}/mc$, one has:

$$\mathbf{F} = m\frac{d\left(\gamma\mathbf{v}\right)}{dt} = m\frac{d\mathbf{u}}{dt}$$

In the electromagnetic case, the acting force is the Lorentz force, $\mathbf{F} = q\left(\bar{\mathbf{E}} + \frac{\mathbf{v}\times\bar{\mathbf{B}}}{c}\right)$, in which the fields are the result of the *shape function* interpolation on the grid, at the particle's position $\mathbf{x}$. One has:

$$\frac{d\mathbf{u}}{dt} = \frac{q}{m}\left(\mathbf{E} + \frac{\mathbf{v}\times\mathbf{B}}{c}\right)$$

The so-called leapfrog method can be used to resolve numerically a symplectic system of differential equations (with the appropriate initial condition), of the kind:

$$\begin{cases} \dot{\mathbf{u}} & = \frac{1}{m}\mathbf{F} \\ \dot{\mathbf{x}} & = \mathbf{u}/\gamma \end{cases}$$

Defining $\mathbf{x}$ and $\mathbf{u}$ at staggered discrete times $\mathbf{x}^n = \mathbf{x}(n\Delta t)$, $\mathbf{u}^{n+1/2} = \mathbf{u}(n\Delta t + \Delta t/2)$, the method integrates iteratively the equations of motion according to the replacement rules:

$$\begin{cases} \frac{\mathbf{u}^{n+1/2}-\mathbf{u}^{n-1/2}}{\Delta t} & = \frac{1}{m}\mathbf{F}^n \\ \frac{\mathbf{x}^{n+1}-\mathbf{x}^n}{\Delta t} & = \mathbf{u}^{n+1/2}/\gamma^{n+1/2} \end{cases}$$

In the non-relativistic case $\gamma = 1$ and if the force can be defined at integer times $\mathbf{F} = \mathbf{F}(x) \rightarrow \mathbf{F}^n = \mathbf{F}(x^n)$, all the quantities are discretized and centered correctly in time.

In our case, instead, Lorentz force requires to know the momentum $\mathbf{u}$ and the relativistic factor γ at integer times n:

$$\frac{\mathbf{u}^{n+1/2} - \mathbf{u}^{n-1/2}}{\Delta t} = \frac{q}{mc}\left(\mathbf{E}^n + \frac{\mathbf{u}^n}{2\gamma^n}\times\mathbf{B}^n\right)$$

It is therefore necessary to center in time (obtaining the values at times $n\Delta t$) $\mathbf{u}^n$ and γ^n. The momentum can be centered using a temporal average (which formally maintains the second-order accuracy):

$$\mathbf{u}^n = \mathbf{u^{n+1/2}} + \mathbf{u^{n-1/2}},$$

and leads to an implicit equation:

$$\frac{\mathbf{u}^{n+1/2} - \mathbf{u}^{n-1/2}}{\Delta t} = \frac{q}{mc}\left(\mathbf{E}^n + \frac{\mathbf{u}^{n+1/2} + \mathbf{u}^{n-1/2}}{2\gamma^n} \times \mathbf{B}^n\right),$$

whereas, for the centering of γ^n, it is convenient to use the *Boris method.* Defining $\mathbf{u}^+$ and $\mathbf{u}^-$ by the relations:

$$\mathbf{u}^{n-1/2} = \mathbf{u}^- - \frac{q\mathbf{E}^n \Delta t}{2mc}$$

$$\mathbf{u}^{n+1/2} = \mathbf{u}^+ + \frac{q\mathbf{E}^n \Delta t}{2mc}$$

and replacing in the above equation, one has:

$$\frac{\mathbf{u}^+ - \mathbf{u}^-}{\Delta t} = (\mathbf{u}^+ + \mathbf{u}^-) \times \frac{q}{2\gamma^n mc}\mathbf{B}$$

$$\mathbf{u}^+ - \mathbf{u}^- = (\mathbf{u}^+ + \mathbf{u}^-) \times \frac{q\mathbf{B}\Delta t}{2\gamma^n mc}$$

The very last equation represent a rotation of the vector $\mathbf{u}$ around an axis parallel to $\mathbf{B}$ of an angle $\theta = -2\arctan(qB\Delta t/2\gamma mc)$. Therefore the relation $\gamma^n = \sqrt{1 + \left(\frac{u^-}{c}\right)^2} = \sqrt{1 + \left(\frac{u^+}{c}\right)^2}$ holds.

Introducing $\mathbf{t} = \frac{q\mathbf{B}\Delta t}{2\gamma^n mc}$, one has:

$$\mathbf{u}^+ - \mathbf{u}^- = (\mathbf{u}^+ + \mathbf{u}^-) \times \mathbf{t}$$

But this is a linear system in the unknowns u_x^+, u_y^+, u_z^+:

$$\begin{cases} t_y\,(u_z^+ + u_z^-) - t_z\,(u_y^+ + u_y^-) + u_x^+ - u_x^- & = 0 \\ -t_x\,(u_z^+ + u_z^-) + u_y^+ + t_z\,(u_x^+ + u_x^-) - u_y^- & = 0 \\ u_z^+ + t_x\,(u_y^+ + u_y^-) - t_y\,(u_x^+ + u_x^-) - u_z^- & = 0 \end{cases},$$

having, as solution:

$$\mathbf{u}^{'} = \mathbf{u}^{-} + \mathbf{u}^{-} \times \mathbf{t}$$
$$\mathbf{u}^{+} = \mathbf{u}^{-} + \mathbf{u}^{'} \times \tfrac{2\mathbf{t}}{1+t^2}$$
$$\mathbf{u}^{+} = \mathbf{u}^{-} + (\mathbf{u}^{-} + \mathbf{u}^{-} \times \mathbf{t}) \times \tfrac{2\mathbf{t}}{1+t^2}$$

The position of the particle is obtained centering respect to the values of $\mathbf{u}, \gamma$:

$$\mathbf{x}^{n+1} = \mathbf{x}^{n} + \mathbf{v}^{n+1/2}\Delta t = \mathbf{x}^{n} + \frac{\mathbf{u}^{n+1/2}}{\gamma^{n+1/2}}\Delta t$$

The scheme is time-reversible and introduces a second-order error in the particle trajectory.

3.2.3. Bringing all together

For writing a PIC code using the Boris scheme, one has to match the time indices in the evolution equations derived in the previous subsections.

The chain of computation for the $n-th$ simulation cycle, assuming that the quantities are know at times $\mathbf{E^n}, \mathbf{B}^{n-1/2}, \mathbf{x}^n, \mathbf{u}^{n-1/2}$, reads:

1. Advance $\mathbf{B}^{n-1/2} \to \mathbf{B}^{n}$ using $\mathbf{E}^n$.

2. Advance $\mathbf{u}^{n-1/2} \to \mathbf{u}^{n+1/2}$ using $\mathbf{E}^n$ and $\mathbf{B}^n$.

3. Advance $\mathbf{x}^{n+1}$ using $\mathbf{x}^n$ and $\mathbf{u}^{n+1/2}$.

4. Compute $\mathbf{J}^{n+1/2}$ using $\mathbf{u}^{n+1/2}$ and $\mathbf{x}^{n+1/2} = \mathbf{x}^n + \mathbf{x}^{n+1}$.

5. Advance $\mathbf{B}^{n} \to \mathbf{B}^{n+1/2}$ using $\mathbf{E}^n$.

6. Advance $\mathbf{E}^{n} \to \mathbf{E}^{n+1}$ using $\mathbf{B}^{n+1/2}$ and $\mathbf{J}^{n+1/2}$.

3.3. Charge conservation using Esirkepov Shape functions

The Esirkepov method ([72]) for electrical current deposition enforces charge conservation directly inside the deposition algorithm, and it can be applied for quasi-particles of arbitrary shape factors.

As, analytically, the charge continuity equation $\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0$, guarantees the Gauss law to be compatible with the dynamics (determined by Maxwell Ampere equation), an analogue equation must hold (with the appropriate discretized operators) for numerical discrete quantities.

This equation can be obtained using the following properties of the discretized operators defined in 3.2.3 (Δ is here the discrete Poisson operator in central differences):

$$\begin{array}{lll} \nabla^{\pm} \cdot \nabla^{\pm} \times & = 0 & \\ \nabla^{\pm} \cdot \nabla^{\mp} & = \Delta & = \frac{f_{i+1,j,k} - 2f_{i,j,k} + f_{i-1,j,k}}{dx^2} + \frac{f_{i,j+1,k} - 2f_{i,j,k} + f_{i,j-1,k}}{dy^2} + \frac{f_{i,j,k+1} - 2f_{i,j,k} + f_{i,j,k-1}}{dz^2} \end{array}$$

, and applying the $\nabla^{-} \cdot$ divergence to the third equation in 3.2.4, one has:

$$\frac{\nabla^{-} \cdot \mathbf{E}^{n+1} - \nabla^{-} \cdot \mathbf{E}^{n}}{dt} = c\nabla^{-} \cdot \nabla^{-} \times \mathbf{B}^{n+1/2} - 4\pi \cdot \nabla^{-}\mathbf{J}^{n+1/2}$$

, or recalling Gauss law, $\nabla^{-} \cdot \mathbf{E}^{n} = 4\pi\rho^{n}$, :

$$\frac{\rho^{n+1} - \rho^{n}}{dt} + \nabla^{-} \cdot \mathbf{J}^{n+1/2} = 0 \tag{3.3.1}$$

Gauss Law is satisfied at every time step only if the obtained *discretized continuity equation* 3.3.1 holds at every time step, explicitly:

$$\frac{\rho^{n+1}_{i,j,k} - \rho^{n}_{i,j,k}}{dt} + \frac{J^{n+1/2}_{x_{i+1/2,j,k}} - J^{n+1/2}_{x_{i-1/2,j,k}}}{dx} + \frac{J^{n+1/2}_{y_{i,j+1/2,k}} - J^{n+1/2}_{y_{i,j-1/2,k}}}{dy} + \frac{J^{n+1/2}_{z_{i,j,k+1/2}} - J^{n+1/2}_{z_{i,j,k-1/2}}}{dz} = 0 \tag{3.3.2}$$

The charge density, in a PIC, is constructed from the form factors S (the cell-integrated shape functions) of the quasiparticles, indexed by α, with position $\mathbf{x}^{n}_{\alpha}$:

$$\begin{array}{rcl} \rho^{n}_{i,j,k} & = & \sum\limits_{\alpha \in particles} q_{\alpha} S_{i,j,k}(\mathbf{x}^{n}_{\alpha}) \,, \\ S_{i,j,k}(\mathbf{x}_{\alpha}) & = & S\left(\tilde{\mathbf{x}}_{\alpha_{\mathbf{ijk}}} = (x_i - x_{\alpha})/dx, (y_j - y_{\alpha})/dy, (z_k - z_{\alpha})/dz\right), \\ \sum\limits_{ijk \in grid} S_{i,j,k}(\mathbf{x}_{alpha}) & = & 1 \,\forall\alpha \end{array}$$

Considering a single particle of charge q one therefore has $\rho^{n}_{i,j,k} = qS_{i,j,k}(\mathbf{x})$. During its motion, in a timestep, it shifts by $(\delta x, \delta y, \delta z)$ from the position $\mathbf{x}^{n}$ to $\mathbf{x}^{n+1}$. Using the vector $\mathbf{W}$, defined as

$$\begin{cases} W_{x_{ijk}} & = -\frac{dt}{q}\frac{J_{x_{i+1,j,k}}-J_{x_{i,j,k}}}{dx} \\ W_{y_{ijk}} & = -\frac{dt}{q}\frac{J_{y_{i,j+1,k}}-J_{y_{i,j,k}}}{dy} \\ W_{z_{ijk}} & = -\frac{dt}{q}\frac{J_{z_{i,j,k+1}}-J_{z_{i,j,k}}}{dz} \end{cases} \tag{3.3.3}$$

, into the discretized continuity equation 3.3.2 one is lead to the equation:

$$\begin{aligned} S_{ijk}(\mathbf{x}^{n+1}) - S_{ijk}(\mathbf{x}^{n}) - W_{x_{ijk}} + W_{y_{ijk}} + W_{z_{ijk}} = 0 \\ \rightarrow \quad S_{ijk}(x+\delta x, y+\delta y, z+\delta z) - S(x,y,z) = W_{x_{ijk}} + W_{y_{ijk}} + W_{z_{ijk}} \end{aligned}$$

As shown in Ref. [72], if the particle shifts by $(\delta x, \delta y, \delta z)$ along a *straight line*, the $\mathbf{W}$ components are linear combinations of the eight shape factors ($\imath = (i,j,k)$):

$$\begin{gathered} S_\imath(x,y,z),\ S_\imath(x+\delta x,y,z),\ S_\imath(x,y+\delta y,z),\ S_\imath(x,y,z+\delta z), \\ S_\imath(x,y+\delta y,z+\delta z),\ S_\imath(x+\delta x,y,z+\delta z),\ S_\imath(x+\delta x,y+\delta y,z),\ S_\imath(x+\delta x,y+\delta y,z+\delta z) \end{gathered}$$

, and they are given by:

$$\begin{cases} W_{x;\imath} & = \left(\frac{1}{3},\ \frac{1}{3},\ -\frac{1}{6},\ -\frac{1}{6},\ -\frac{1}{3},\ \frac{1}{6},\ \frac{1}{6},\ \frac{1}{3}\right)\cdot \bar{S}_{ijk} \\ W_{y;\imath} & = \left(\frac{1}{3},\ -\frac{1}{6},\ \frac{1}{3},\ -\frac{1}{6},\ \frac{1}{6},\ -\frac{1}{3},\ \frac{1}{6},\ \frac{1}{3}\right)\cdot \bar{S}_{ijk} \\ W_{z;\imath} & = \left(\frac{1}{3},\ -\frac{1}{6},\ -\frac{1}{6},\ \frac{1}{3},\ \frac{1}{6},\ \frac{1}{6},\ -\frac{1}{3},\ \frac{1}{3}\right)\cdot \bar{S}_{ijk} \\ \\ \overline{S}_{ijk} & = \begin{pmatrix} S_\imath(x,y,z) \\ S_\imath(x+\delta x,y,z) \\ S_\imath(x,y+\delta y,z) \\ S_\imath(x,y,z+\delta z) \\ S_\imath(x,y+\delta y,z+\delta z) \\ S_\imath(x+\delta x,y,z+\delta z) \\ S_\imath(x+\delta x,y+\delta y,z) \\ S_\imath(x+\delta x,y+\delta y,z+\delta z) \end{pmatrix} \end{cases}$$

Once **W** is computed for a particle of arbitrary shape factors S, it is possible to iterate (in space) the equations 3.3.3 to calculate the correct contribution to the current **J**. The boundary conditions for the integration (the iteration starting and ending points), are determined by the fact that, outside the particle's finite-size shape function, the contribution of the particle to the current is zero.

4. Jasmine: PIC implementation on GPUs

Particle-in-Cell codes (PIC) are well-established tools for modeling plasma based particle accelerators, providing an accurate kinetic description for plasma physics. PIC simulations are extremely demanding in terms of computational power, hence the development of efficient parallel codes is of great interest [3, 70, 79].

New opportunities towards computational efficiency are given by the recent development of new many-core architectures: Graphics Processing Units (GPUs) have evolved, with the NVIDIA G80 architecture [71], into completely programmable general-purpose massively parallel processors. The new generations of NVIDIA GPU micro-architectures, Fermi (2010), Kepler (2012) and Maxwell (2014), retain and extend the same CUDA programming model, introducing better performance, new features and new low-level instructions.

These architectures offer a new kind of massive, on-chip, parallelism, in which DRAM memory is shared by thousands of concurrent threads.

Due to the massively parallel nature of the architecture, the adaptation of PIC algorithms requires some rethinking. In particular, the particle-to-grid operations (e.g. the evaluation of the current density) need special care to avoid memory inconsistencies and conflicts. Development of GPU PIC codes has become a topic of great interest in the laser-plasma community. The first GPU code scaling efficiently to multiple GPUs, PIConGPU, has been presented by Burau et al. [65]. Kong et al. proposed a method [66] for implementing deposition in a consistent manner while using the fast manual caches available on GPUs. Osiris [68] and Vorpal [67] are also working for having their frameworks running on these architectures.

In this chapter we present our GPU particle-in-cell code framework, named *jasmine*. *jasmine* implements a second order, FDTD-based explicit 3D PIC scheme, supporting double precision and arbitrary-order particle shapes. The optimized four-point derivative/integration scheme OSE2-Lpf, proposed in Ref. [4], helps to reduce dispersion errors for long laser-propagation cases. Charge conservation can be enforced using Esirkepov [72] shape factors and wave-absorbing boundary conditions are implemented using a perfectly matched layer [73].

In section 4.1, we describe a robust algorithm for grid deposition that is efficient for any number of particles per cell and particle shape function order. It exploits the exposed GPU memory hierarchy and avoids the use of shared memory atomic operations, which can hurt performance in pre-Maxwell GPU architectures especially when many particles lay on the same cell.

The code scalability for warm-plasma simulations has been measured up to 88 GPUs and a dynamic load-balancing algorithm increases the efficiency of the code for certain laser-plasma accelerator simulations, as discussed in section 4.2.

The code also supports tracking and dumping quasi-particle trajectories that allow scattering radiation as post-processing, and it can model the laser field tunneling ionization process modeling ionization with the ADK model (section4.4.1).

4.1. GPU parallelization

A modern GPU chip is composed by ˜10-20 Streaming Multiprocessor (SM). Each SM dispatches instructions to groups of ˜32 lightweight threads, named warps. In principle, each thread has its own execution context (registers, etc.), but serialization occurs if threads in a warp take different execution paths. In fact, the SMs hardware paradigm is the SIMD (single instruction, multiple data), as they only can dispatch instructions to a warp of threads (˜lanes). On the other hand, the (software/) programming model provides an abstract execution model ("SIMT", single instruction multiple thread), in which threads have their own independent execution context (program counter), and, in this view, vectorization/serialization is performed dynamically by the GPU hardware.

Threads are grouped in application-configurable "blocks", sharing a low latency, on-chip, memory space, named shared memory (16-64 KB in size per SM, depending on the architecture).

The SMs hide instruction execution latencies (memory access, arithmetic, ...) by keeping many warps at in execution at once, exploiting the applications' massive parallelism. While waiting for a latency, other operations, on different warps, are started in order to "absorb" it. The SM is designed to minimize the cost of these execution switches; for example, the registers for all the threads in the block(s) being executed are stored in the SM's on-chip register file.

All the (off-chip) DRAM (global memory, in CUDA terminology) of the GPU board is visible to all threads. The DRAM provides much higher bandwidth ($>$ 80 GB/s) than present CPUs and a latency of ~800 cycles. To fully utilize the bandwidth, enough concurrent accesses should be in flight, in order to hide the memory access latency. Furthermore, threads in a warp should access global memory using ordered, "coalesced" access patterns (e.g. all the threads in a warp should access contiguous memory locations).

The massive parallelism, the strictness of global memory access patterns, the latency hiding philosophy, the SIMD-like execution of warps and the different automatic caching philosophy (the number of bytes per "thread" is several orders of magnitude lower compared to CPU models) are constraints that must be taken into account for writing efficient GPU codes.

Parallelizing a Particle-in-Cell code for this architecture presents some challenges. Grid-only operations are straightforward to implement, while efficiency for mesh-to-particle interpolation is achieved when the most efficient memory spaces are used for caching. The critical part is the particle-to-grid current deposition, as it requires a rethinking of the algorithm, because of the memory collisions arising from massive parallelism.

4.1.1. Deposition Algorithm

The most naïve approach for parallelizing the PIC density deposition algorithm for shared-memory, massively parallel architectures is processing each particle in parallel, summing density values to the grids stored in the global shared memory. This leads to race conditions due to the fact that two threads, processing two different particles, need to perform a non-atomic sum to the same memory location if the two particles are in the same cell.

Therefore, atomic operations or synchronized algorithms are required. In addition to that, considering a simulation with an average number of particles per cell N_{ppc} and total shape function order K (2D linear shapes $K = 4$, 3D quadratic$K = 27$), one has that each density grid cell is accessed $K \times N_{ppc}$ times. As stated in [66], it is therefore worthwhile to cache the density grid in the fastest memory space available, such as the shared-memory (the on-chip, multiprocessor, manual cache shared by a block of threads) on the CUDA architecture, which provides several orders of magnitude lower latency and higher throughput compared to the CUDA device global memory.

Our density deposition algorithm consists of the following steps:

1. Particles are kept organized by grid "neighborhood" first, and by grid cell index next. One CUDA thread block processes particles that have their center inside a grid neighborhood. The density data associated to the neighborhood is cached in the block's shared memory. This caching saves up to $K \times N_{ppc}$ global memory accesses.

2. For each cell overlapped by the particle's shape, the sums are performed using an in-block segmented reduction in shared memory. The segments (brackets) are determined by the particle cell index.

3. The results of the sums are scattered to the density grid cache and finally to the global memory grid.

The benefits of using parallel reduction instead of using shared-memory atomic operations (prior to the Maxwell architecture shared memory atomic operations are implemented with lock/update/unlock patterns) include better performance, lower accumulated round-off error [78] and full determinism of the order of the sum operations.

In subsection 5.2.3, another approach for implementing densities deposition is discussed; for a specific numerical scheme, modeling LWFA using the ponderomotive approximation in cylindrical symmetry r-z (Sec. 5.1), an algorithm using global memory atomics, on density contributions "grouped" using Kepler architecture "shuffle" instructions (used for efficiently communicating data across registers in a warp) performs well and can be implemented in a very straightforward manner.

Enforcing Charge Conservation using Esirkepov's Method

This scan-based approach can present an obstacle when optimizing the number of grid deposition operations in the Esirkepov's charge conservation method [72], as the number of parallel sums is fixed in the sorting/bracketing step.

The conservation method can be directly implemented without branching, at the cost of increasing by two in each direction the size of the block of cells in which a particle deposits its density contribution. This introduces a large overhead. For example, for the 3D quadratic shape case the number of cells involved increases from 27 to 125. Such

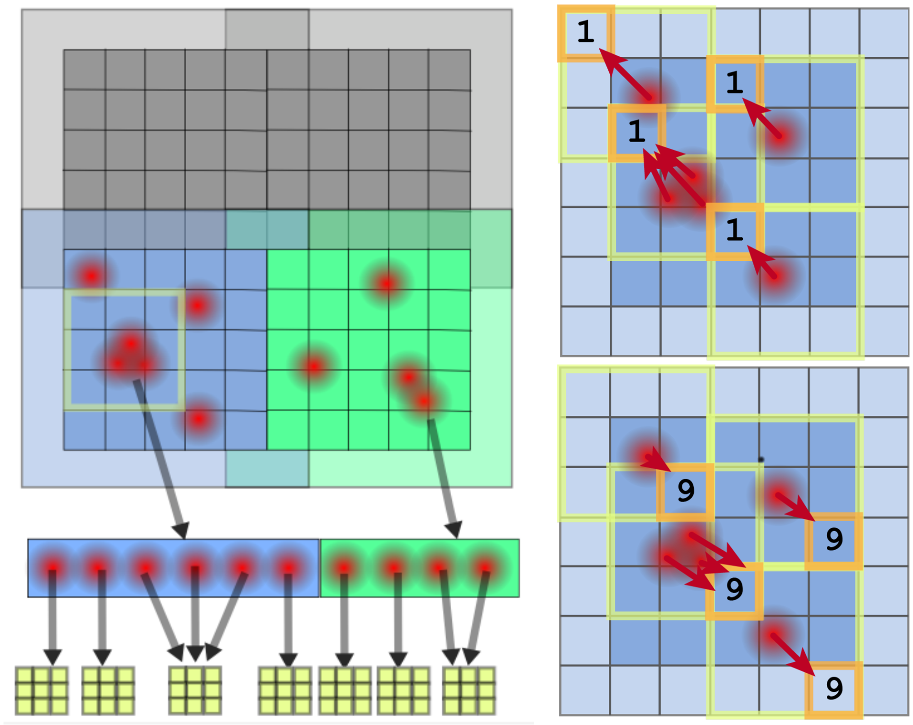


Figure 4.1.1.: Shared-memory-atomics free, reduction-based deposition algorithm. (left) Thread blocks are assigned the particles contained in their density subdomain (blue and green subdomains), and they cache the density field in shared memory. Particles are kept organized inside the neighborhood and, for each each cell, for each shape factor (in this case, 2d with quadratic shape factors) a (segmented) reduction in shared memory is performed. The figure on the (right) shows where the reductions for each cell and the first and last shape factor are summed in the shared cache.

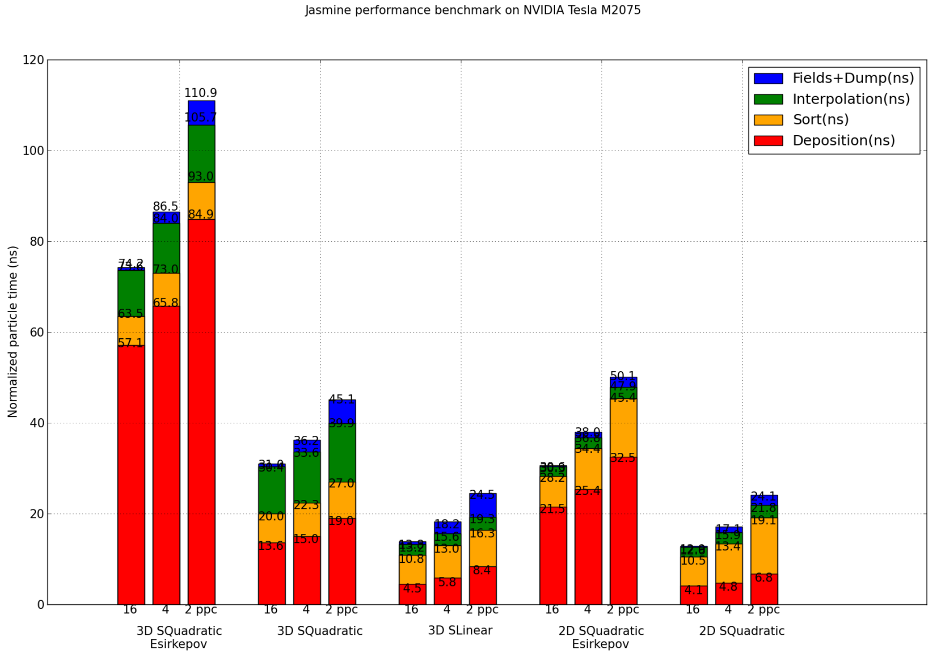


Figure 4.1.2.: jasmine performance benchmark results for varying numerical configurations. A test case bubble regime laser plasma acceleration simulation with physical parameters “a0=7.7, w0=9.0 μm, n0=1e19 1/cm3” was run for $ct = 60\ \mu m$ in double precision.

overhead can be reduced since the actual particle shape is increased in size only in the directions along which it crosses its cell boundary in the last time step.

A solution for optimizing the number of operations in a scan-based deposition algorithm could be to organize the particles according by cell-cross directions. Unfortunately, such method would result in high fragmentation and inefficiency, as the number of cell crossing configurations is 27 for the 3D case.

However, for grids in which the transverse cell size is larger than the longitudinal one (more than a factor of two), it is convenient to split particles that are crossing cells transversely from the ones that are not, which are the majority because of the CFL condition. In the case mentioned before, this reduces the total shape size from 125 to 45 for the vast majority of the particles.

4.2. Multi-GPU Parallelization

Large 3D simulations do not fit memory limitations of a single GPU board (˜6-20GB depending on the GPU). CPU codes have shown to be able to scale very efficiently up to hundreds of thousands cores and GPU PICs must achieve efficient multi-GPU parallelization/scalability in order to be able to solve the same problems and be competitive.

Whereas the implementation for a single GPU, being shared-memory based, required no explicit subdomain boundary exchange, multiple-GPU parallelization does. The classical domain decomposition (in 3D) using MPI approach was used in jasmine. Zero-copy, host mapped memory is used when possible for moving buffers from GPU RAM to the CPU one, through which data is passed before network transfers. Recent developments in the GPU HPC technology allow remote GPU remote direct memory access (GPUDirect™ RDMA), and their use will be the subject of future developments.

In order to achieve good scalability, overlapping network communication with computation is even more important for GPU codes than for CPU ones, as the computational power over network bandwidth ratio might be higher. Therefore, in jasmine, particles are exchanged across GPU nodes concurrently with current deposition. All file output (saving snapshots of simulations and online diagnostics) operations are executed in parallel with computation as well.

A scalability measurement was run for a warm plasma test case on the clusters APE QUonG and CINECA PLX and the results are shown in figure 2. Thanks to communication-computation overlap, our code shows close-to-ideal weak scaling up to the measured 88 GPU and strong scaling shows that it is possible to increase the performance adding new nodes up to the same limits.

The relative performance of jasmine, versus our CPU PIC implementation ALaDyn [3], has been measured running full scale, 3D, benchmark simulations in the same numerical conditions. The performance of jasmine per single, Fermi-generation, GPU board has shown to be roughly equivalent to the one of ALaDyn run on 40-50 modern CPU cores.

4.2.1. Simple Load-Balancing Algorithm for Laser Plasma Simulations

In laser plasma simulations, the dynamics tend to create large-scale particle density inhomogeneity. In parallel particle codes, this translates into a load imbalance of the processors, which severely limits scalability.

Preparing to scale to a considerably higher number of GPUs, in jasmine we have

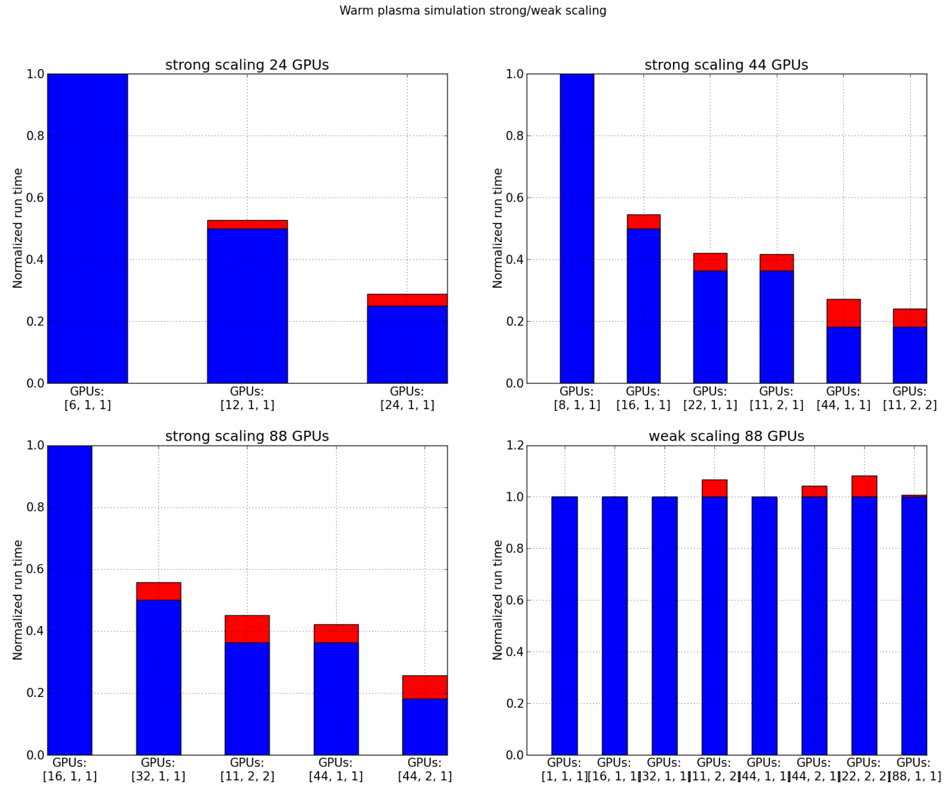


Figure 4.2.1.: jasmine strong/weak scaling benchmarks on INFN APE QUonG (24 GPUs) and CINECA PLX (used 88 GPUs) machines

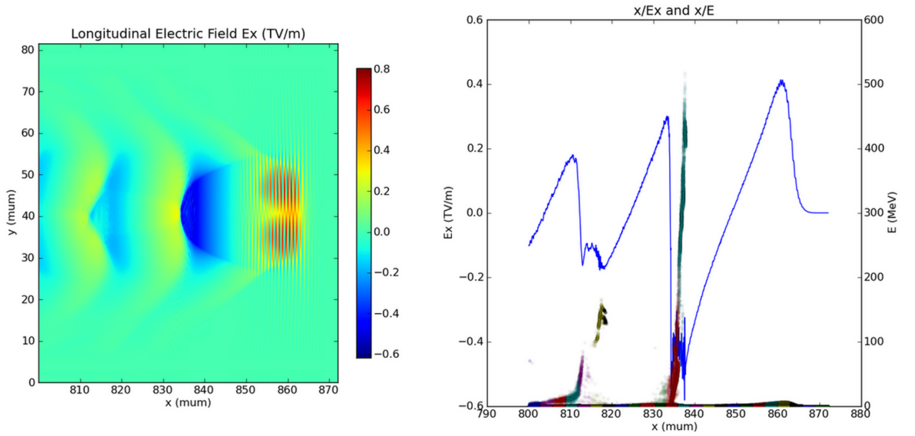


Figure 4.2.2.: LPA bubble regime simulation test case used for benchmarking the load-balancing algorithm. Physical parameters are “a0=5.8, w0=13.2 μm, n0=3.8e18 1/cm3”. Snapshot at ct = 800.0 μm. To stress the load imbalance effect , the benchmark was run in a smaller simulation box (50μm x 50μm x 50μm).

implemented a dynamic load-balancing algorithm. We have chosen to let the subdomain topology (and therefore the neighbors number) intact, but we resize the subdomain boxes by moving their boundaries.

Rather than seeking a global solution for each iteration of the algorithm, we proceed in a hill-descending fashion: each few simulation time-steps we choose the chain of resizing “moves” (each move consists in resizing a subdomain along a particular direction) that minimizes a fitness function. The fitness function seeks the states which minimize both the load variance across the nodes and the maximum loaded node / average node ratio.

The load redistribution process has shown to converge very rapidly for both one-dimensional domain decomposition and higher dimensional ones, up to the number of subdomains we could test on. We tested the algorithm’s capability to redistribute the computational load for a typical 3D bubble regime laser plasma acceleration simulation. The results of these tests show high efficiency gains and are shown in figure 4.2.2, 4.2.3, 4.2.4 and 4.2.5.

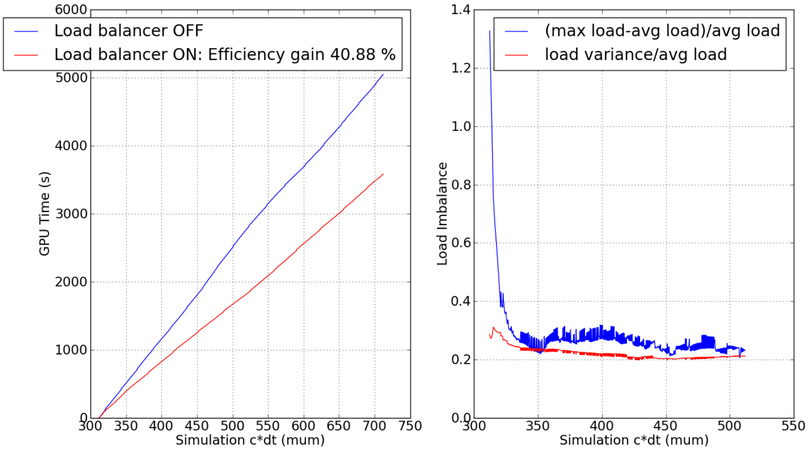


Figure 4.2.3.: Load balancing algorithm test run on 12 GPUs. Simulation used for testing discussed in figure 4.2.2.

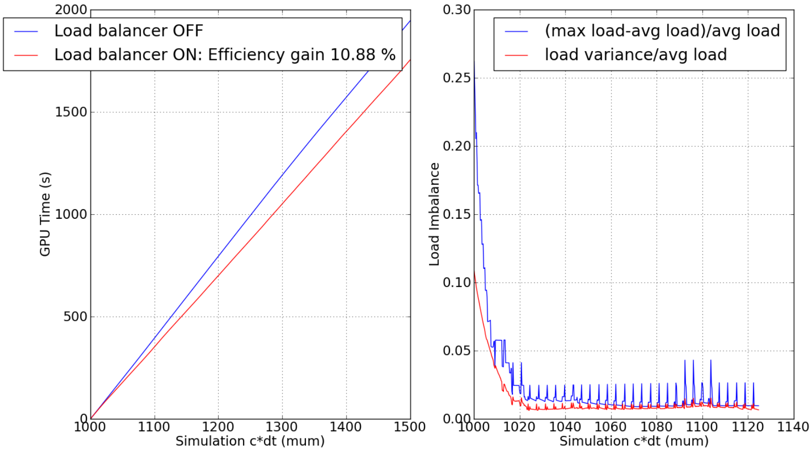


Figure 4.2.4.: Load balancing algorithm test run on 36 GPUs and 2D domain decomposition (12x3). Simulation used for testing discussed in figure 4.2.2.

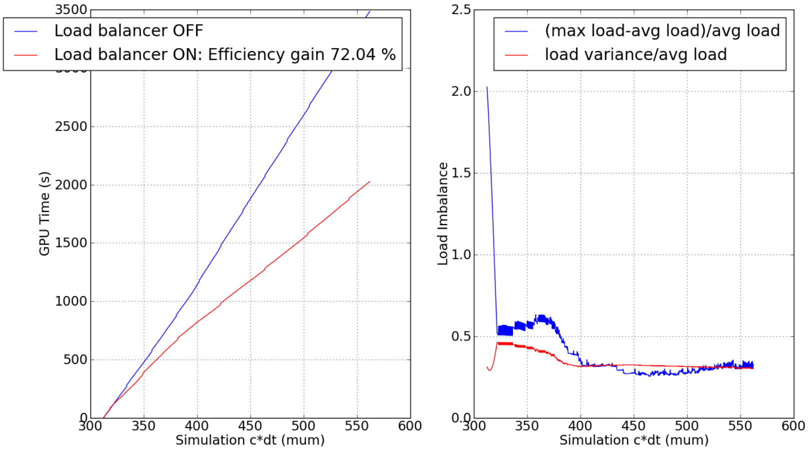


Figure 4.2.5.: Load balancing algorithm test run on 72 GPUs and 3D domain decomposition (8x3x3). Simulation used for testing discussed in figure 4.2.2.

4.3. Meta-programming Technique

We have chosen to rely on meta-programming techniques to write maintainable and possibly unique code for all quasi-particle weighting orders, numerical schemes and parallel architectures (GPUs but also a baseline implementation for multicore CPUs). In fact, the core PIC algorithms (deposition, interpolation, ...) follow a generic structure but they need to be configured my many parameters: dimensionality, particle shapes, physical laws and numerical schemes.

The core PIC algorithms (deposition, interpolation, ...), have been written as final-configuration-agnostic templates, which can be used to generate code for the simulation parameters at compile time, as a pre-compilation step. The templates take as input also the data structure used to store simulation status, which is part of the simulation configuration (for example, particle data can be organized with a parameter switch in a GPU-friendly manner, as struct-of-arrays or array-of-structs). This opens the possibility to integrate the jasmine GPU kernels into other PIC codes.

We have created a python framework, using the general-purpose template engine mako [77] as text-rewriting system, to generate the final, specialized, C++ code. This way, a full and extremely flexible programming language, python, can be used as a macro language, providing more freedom and transparency compared to standard C++ template meta-programming (at obvious costs). In addition to help writing generic deposition algorithms (using standalone components) and generating a baseline multicore CPU implementation based on the CUDA kernels, the framework has been used for building numerical features. As an example, jasmine supports physical coordinates stretching, providing a larger simulation box, useful for delaying effect at the boundaries due to quasi-particles leaving the domain, without increasing the computational cost. The inverse of the stretching function is required for integrating the quasi-particles motion, and such inverse function (with all the coefficients and pieces if piecewise) is symbolically pre-computed at compile time (via the package sympy), and the required, efficient, C++ code is generated automatically by the template framework, once the stretching function has been chosen and configured by the user.

4.4. GPU implementation of PIC auxiliary features

Besides the core PIC algorithms, a code framework for simulating laser plasma interactions needs some additional features, such as particle trajectory tracking, diagnostics and dynamic particle allocation (for the moving window, tunneling ionization and particle splitting).

Obviously, for these operations not to become a bottleneck, they also must be parallelized and implemented for the GPU architecture. Algorithmically, most of them can be built using combinations of “scan primitives” [75], a family of parallel binary operations which return, for an input vector, the vector of the incremental operation results, requiring global knowledge of the inputs for each output. In jasmine, we have used scan primitives provided by the “thrust” library [76]. Sorting, stream compaction and global sums can be implemented in parallel for shared memory architectures using these primitives.

For example, stream compaction is used to filter and select in parallel the relevant particles (i.e. belonging to the accelerated bunches) each timestep when simulating radiation emission processes (e.g. bubble betatron motion or Thomson scattering). The quasi-particles’ trajectories are then used for simulating radiation emission, as a post-processing step and using the algorithm presented in Ref. [92]. Similarly, a scan operation orchestrates the parallel work of the threads assigned to the different ions and computes, in parallel, the output indices of the quasi-particles generated by the ionization state changes of the ions (see subsection 4.4.1).

4.4.1. Tunneling ionization modeling with the ADK model

For high-field laser-gas interaction, the ionization process, producing the plasma, is dominated by tunneling ionization. In numerical studies, the ionization process is often omitted, as ionization occurs before the main body of the pulse hits the target. For example, low-Z gases (H or He) are ionizable by laser picosecond prepulses and ionization effects are usually neglectable.

When a higher-Z gas target is used, or for laser intensities close to the ionization threshold, the modeling the ionization process is necessary as it influences the generation of the plasma wake.

Tunneling ionization effects have been exploited for inventing methods of radiation and particle acceleration generation. As an example, the self-injection threshold in LWFA can be lowered using high-Z gas targets, in which electrons can be generated by ionization at the right phase of the wakefield for injection [88, 91, 90, 89], allowing better control on the beam quality.

For modeling these applications, PIC codes must include consistent ionization simulation modules. A detailed analysis of the numerical implementation of the ADK (Ammosov-Delone-Krainov) tunneling ionization model [84] in PIC codes and a benchmark of existing implementations can be found in Ref. [81]. The ADK model predicts the ionization rate W_{lm} of a complex atom in a static external electric field E_{dc}, for the electron with quantum numbers l, m, using the semi-classical approximation [84, 81].

In PIC simulators, variables storing the ionization status are added to the ions (or atoms) quasi-particle species. The ionization rate W_{lm} is then computed using the local field and the ionization status via the ADK formulas, at each time step. For each simulation timestep (Δt long), once the ionization rate has been computed, the ions quasi-particles are ionized with probability $P = 1 - \exp(W(t)\Delta t)$, using a uniform random number generator and the rejection sampling method for the decision. At high laser intensities, multi-level ionization may happen in a single PIC timestep, and, in order to model this effect accurately, a system of equations involving all the ionization rates has to be integrated [81].

This discrete sampling method introduces numerical noise, and the number of PIC quasi-particles per cell can be used to control it [81]. Furthermore, the spatiotemporal resolution should be chosen appropriately in order to ensure that the field peaks (that can be close to a level ionization threshold) are appropriately sampled in time [81]. Figure

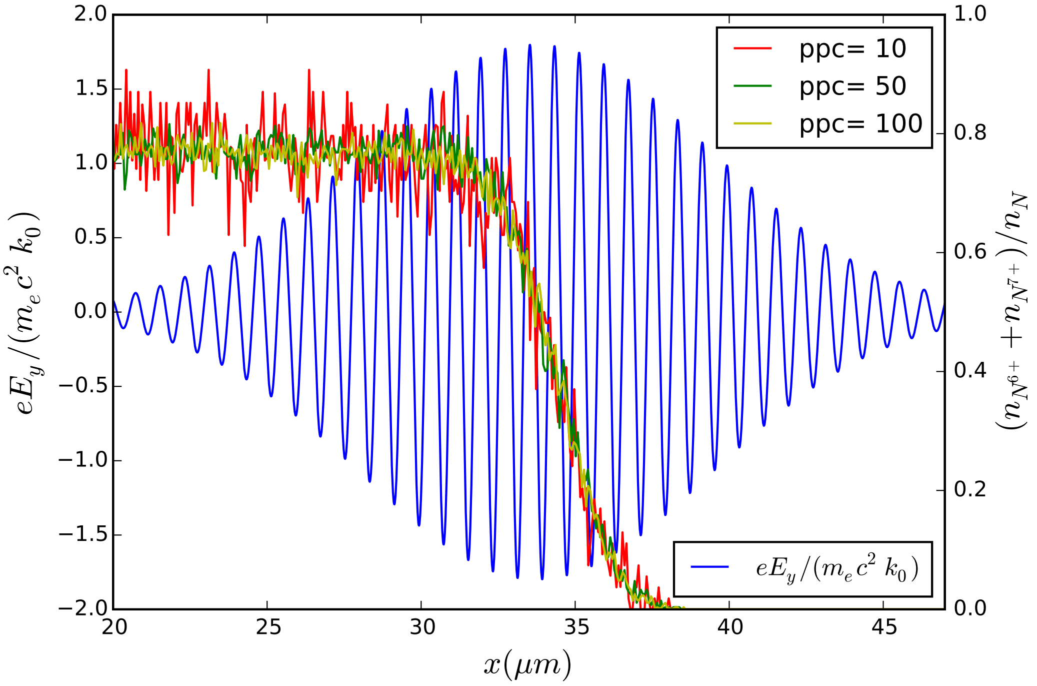


Figure 4.4.1.: A gaussian laser pulse with peak $a_0 = 1.8$, $L = 10\lambda_0$ and wavelength $\lambda_0 = 0.8\mu m$ propagates through and ionizes a uniform Nitrogen gas 1D slab, with total electron density $n = 10^{-5} n_c$. The spatial resolution used is $\Delta_x = \lambda_0/66$. The laser electric field E_y (blue line, left y axis), is overlayed by the locally averaged ratio (right axis) of nitrogen ions in statuses N^{6+} and N^{7+} $((n_{N^{6+}} + n_{N^{7+}})/n_N)$, for *jasmine* numerical parameters: 10 particles per cell (red line), 50 particles per cell (green line), 100 particles per cell (yellow line).

4.4.1 shows a 1D benchmark run with *jasmine* (similar to the one run in Ref. [81]) in which the effect of the number of particles per cell on the ionization noise is visible.

In jasmine, the GPU implementation of the ADK module was almost straightforward. A first kernel computes the ions' ionization rates, samples the probability rate (using the library curand for uniform random number generation), updates the ionization status and outputs the number of ionized electrons for each ion. For each ion being ionized, a second kernel then initializes the status of the newly created electrons, using a parallel scan [75] to compute (in parallel) computational electrons' indices in the particles array.

A 2D benchmark simulation with ionization induced injection was run with *jasmine* using similar parameters as in the 2D benchmark in Ref. [81]. A laser pulse with peak

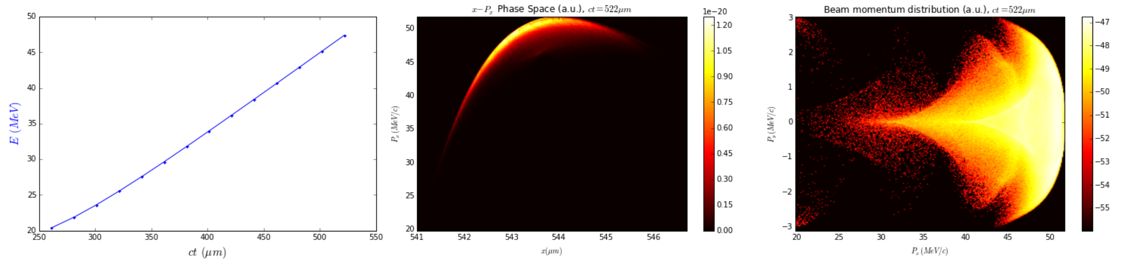


Figure 4.4.2.: 2D ionization induced benchmark simulation run with the code *jasmine.* (Left) Beam average energy evolution. (Center) Longitudinal beam phase space. (Right) Beam $P_x - P_y$ distribution. See parameters and interpretation in the text.

normalized potential $a_0 = 2.0$ and $\tau_{fwhm} = 56fs$ (cos^2 longitudinal profile) is focused to a focal spot with waist $w_0 = 14.1\mu m$ and propagates through a preionized plasma with electron density $n_p = 1.64 \cdot 10^{18}$. A $8\mu m$ thick (with $8\mu m$ ramps) layer of Nitrogen (atom density $n_N = 1.65 \cdot 10^{18}$) at the entrance of the plasma is used for localized ionization induced injection, allowing the acceleration of a low energy spread ($< 3\%$) electron bunch.

5. INF&RNO

INF&RNO (Ref. [26, 27, 28]) is a 2D cylindrical (r-z) code, developed at LBNL, that adopts an envelope model for the laser pulse and uses the ponderomotive force approximation to describe the interaction of the pulse with the plasma. Its purpose is simulating laser-plasma electron accelerators, where a *short* and intense laser pulse interacts with an *underdense* plasma over distances that can reach the meter.

A 3D "full" , i.e. resolving the smallest spatial/temporal scale which is the laser wavelength, PIC simulation requires $10^4 - 10^5$CPU core hours in present supercomputers for a millimeter scale plasma. The computational complexity required for scaling to a meter-scale plasma with fixed accuracy makes simulating meter-scale propagations unfeasible with standard computational tools and today's supercomputers.

Anyways, 3D simulations are crucial for understanding phenomena and for designing LPAs working at these physical scales. Two solutions have been proposed to overcome these computational limitations. The first solution consists in running the full PIC simulation in an optimal in a optimal Lorentz "boosted" frame (*the Boosted Lorentz frame* (*BLF*)) moving along the laser propagation direction at relativistic velocities. If one can neglect the effect of backward propagating waves (e.g. Raman backscattering), the unbalance between the maximum and minimum physical scales involved in a simulation can be reduced, leading to a shortening of the simulation length (also by several orders of magnitude). Applications and limitations of this technique are discussed in Ref [30, 33, 32, 31].

The second solution is the use of reduced models. Dimensional reduction, for example assuming cylindrical symmetry instead of the full cartesian geometry, can provide $> 10x$ simulation speedups. Furthermore, the envelope approximation for the laser pulse and the ponderomotive approximation for describing the laser-plasma interaction remove the necessity to resolve the laser wavelength as the minimum spatial/temporal scale. In fact, applying these approximations, the smallest scale to resolve is the plasma wavelength, which makes the computational work scale much better in long propagation distances/high energy/low density regimes (for laser wakefield accelerators with short pulses $\lambda_{laser} \ll \lambda_p \sim L_{envelope}$).

INF&RNO can run the simulations using the fluid or PIC model (or both, for staged simulations). The fluid simulations can be run in a Boosted Lorentz Frame. The parallelized version of INF&RNO is capable of running LPA simulations, including meter-scale, 10 GeV acceleration stages (e.g. BELLA, Ref. [25]), in practically feasible times on modern supercomputers. Another, stronger, model reduction is the quasi-static approximation, discussed in the next chapter.

5.1. Numerical scheme

The laser envelope model describes the laser pulse propagation in a fully ionized plasma. It leads to a slow-varying laser envelope equation which is obtained representing the fields as modulations of *fast and slow* parts. The model assumes the ions to be stationary and the collision time to be much greater than the other timescales.

Both the hypothesis are valid for laser plasma interactions involving a short laser pulse propagating in an underdense plasma ($\omega_p^2/\omega^2 \ll 1$): the electrons' thermal speed is small compared to their quiver speed in the laser field and their collision time is much greater than the laser pulse length.

5.1.1. Laser envelope equation derivation

The wave equation and Poisson equation for the normalized potentials in the Coloumb gauge, being e and m are the charge and the mass of the electron:

$$\begin{cases} \mathbf{E} & = -\nabla\Phi - \frac{\partial \mathbf{A}}{\partial ct} \\ \mathbf{B} & = \nabla \times \mathbf{A} \\ 0 & = \nabla \cdot \mathbf{A} \end{cases} \rightarrow \begin{cases} \phi & = \frac{e\Phi}{mc^2} \\ \mathbf{a} & = \frac{e\mathbf{A}}{mc^2} \end{cases},$$

read:

$$\begin{cases} \left(\nabla^2 - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\right)\mathbf{a} & = k_p^2 \dfrac{n}{n_0}\dfrac{\mathbf{u}}{\gamma} + \dfrac{\partial}{\partial ct}\nabla\phi = -k_p^2 \dfrac{1}{n_0 ce}\mathbf{J} + \dfrac{\partial}{\partial ct}\nabla\phi \\ \nabla^2\phi & = k_p^2\left(n - n_i\right)/n_0 \end{cases},$$

in which $\mathbf{u} = \gamma\mathbf{v}/c = \mathbf{p}/(mc)$ is the fluid average of the normalized electron momentum and $\gamma = (1+u^2)^{1/2}$, $k_p = \left(\frac{4\pi n_0 e^2}{mc^2}\right)^{1/2}$, $\mathbf{J} = -enc\mathbf{u}/\gamma$.

Introducing the comoving coordinates $\xi = z - ct, \tau = t$, and neglecting the term $\frac{\partial}{\partial ct}\nabla\phi$ (the fast part of the electrostatic potential $\phi_{fast} \sim \exp(ik\xi)$ is typically small compared to the fast part of the plasma current), the wave equation becomes:

$$\left(\nabla_\perp^2 + \frac{2}{c}\frac{\partial^2}{\partial\xi\partial\tau} - \frac{1}{c^2}\frac{\partial^2}{\partial\tau^2}\right)\mathbf{a} \backsimeq k_p^2 \frac{n}{n_0}\frac{\mathbf{u}}{\gamma}$$

Further possible simplifications are:

- For forward-going light waves, the term $\frac{1}{c^2}\frac{\partial^2}{\partial\tau^2}\mathbf{a}$ can be neglected. On the other hand, this term is important for backward-propagating radiation (i.e. Raman backscattering) and it is necessary when running simulation in a boosted Lorentz frame (see Ref. [26]), as it is required for Lorentz invariance.

- The leading-order transverse electron motion is the laser quiver motion: on the right hand side we can approximate $\mathbf{u} = \mathbf{a}$.

- Assuming a linearly polarized laser field of frequency $\omega = ck$, propagating in the positive z direction, having a transverse component expressible as a modulation of a fast oscillating field and a slow-varying $|\partial_\xi \hat{a}_{slow}| \ll |k\,\hat{a}_{slow}|$ envelope $\hat{a}_{slow}$: $\mathbf{a} = \hat{\mathbf{a}}_{slow}(r,\xi,t)\exp(ik\xi)/2 + c.c.$

Applying these simplifications, one has:

$$\left(\nabla_\perp^2 + 2ick\frac{\partial}{\partial\tau} + \frac{2}{c}\frac{\partial^2}{\partial\xi\partial\tau}\right)\hat{\mathbf{a}}_{slow} \backsimeq k_p^2\frac{n}{n_0}\frac{\hat{\mathbf{a}}_{slow}}{\gamma}$$

or,

$$\left(i\frac{k_0}{k_p} + \partial_\xi\right)\partial_\tau \hat{a} = \frac{1}{2}\frac{n}{n_0}\frac{\hat{a}}{\gamma} - \frac{1}{2}\Delta_\perp \hat{a}$$

5.1.2. Laser envelope equation numerical solution and parallelization

In INF&RNO, the laser envelope is evolved using the envelope equation complete with the second order time derivative term ∂_τ^2

$$\left(\nabla_\perp^2 + 2i\frac{k_0}{k_p}\frac{\partial}{\partial\tau} + 2\frac{\partial^2}{\partial\zeta\partial\tau} - \frac{\partial^2}{\partial\tau^2}\right)\hat{a} = \frac{\delta}{\gamma_{fluid}}\hat{a} \tag{5.1.1}$$

, in which $a_\perp = eA_\perp/mc^2 = \frac{\hat{a}}{2}e^{i(k_0/k_p)\xi} + c.c.$, $2\pi/k_0$ is the laser wavelength, γ_{fluid} is the relativistic factor associated to the local average of plasma velocity and δ is the normalized density n/n_0. Eq. 5.1.1 is then discretized in time using an implicit Crank-Nicholson scheme:

$$-\frac{\hat{a}^{n+1} - 2\hat{a}^n + \hat{a}^{n-1}}{\Delta_\tau^2} + 2\left(i\frac{k_0}{k_p} + \frac{\partial}{\partial\zeta}\right)\left(\frac{\hat{a}^{n+1} - \hat{a}^{n-1}}{2\Delta_\tau}\right) = -\nabla_\perp^2\left(\frac{\hat{a}^{n+1} + \hat{a}^{n-1}}{2}\right) + \frac{\delta^n}{\gamma_{fluid}^n}\left(\frac{\hat{a}^{n+1} + \hat{a}^{n-1}}{2}\right) \tag{5.1.2}$$

For a resonant pulse ($L_{rms} \sim 1$), the characteristic length L_{rms} of the pulse is the smallest relevant scale of interest (and comparable to the plasma skin depth), and so, in principle, the smallest length to resolve. However, during propagation in the plasma, as a consequence of laser-pulse redshifting and depletion, structures smaller than L arise in the laser envelope [35, 34, 36, 28]. This effect has to be taken into account when designing the numerical scheme (Ref [28]) for the laser envelope equation, and always when choosing the resolution to be used in simulations.

In particular, the discrete representation of the longitudinal derivative ∂_ζ requires extra care, as it might introduce significant numerical errors (preventing a correct description of the laser evolution) when the operator is not discretized in its optimal form and/or the small structures forming in $\hat{a}$ are not well resolved. In INF&RNO, the polar form of the complex field $\hat{a}$ is used instead of the Cartesian representation [28]:

$$\hat{a}(\zeta) = a(\zeta)e^{i\theta(\zeta)}$$

where $a(\zeta) = |\hat{a}$ and $\theta = \arg(\hat{a})$. Evaluating the longitudinal derivative of the laser envelope using the polar form proved to have some numerical advantage, as the polar amplitude and phase are less prone to show an oscillatory behavior or significant variability over small scales compared to the real or imaginary parts. Using the polar form, the longitudinal derivative operator becomes $\partial_\zeta\hat{a} = (\partial_\zeta a) \cdot e^{i\theta} + i\hat{a}\,(\partial_\zeta\theta)$.

In figure 5.1.1 the performance of the INF&RNO polar form envelope solver is shown in detail. For a 1D simulation with laser-plasma parameters $a_0 = 1$, $L_{rms} = 1$ and $k_0/k_p = 100$, even at moderately low resolution $L_{rms}/\Delta_\zeta = 30$, the polar form envelope solver result (blue dots) is in excellent agreement with the reference solution obtained with the full PIC code ALaDyn (red solid line). The standard, Cartesian form, solver converges to the correct solution only at high resolution $L_{rms}/\Delta_\zeta = 1000$.

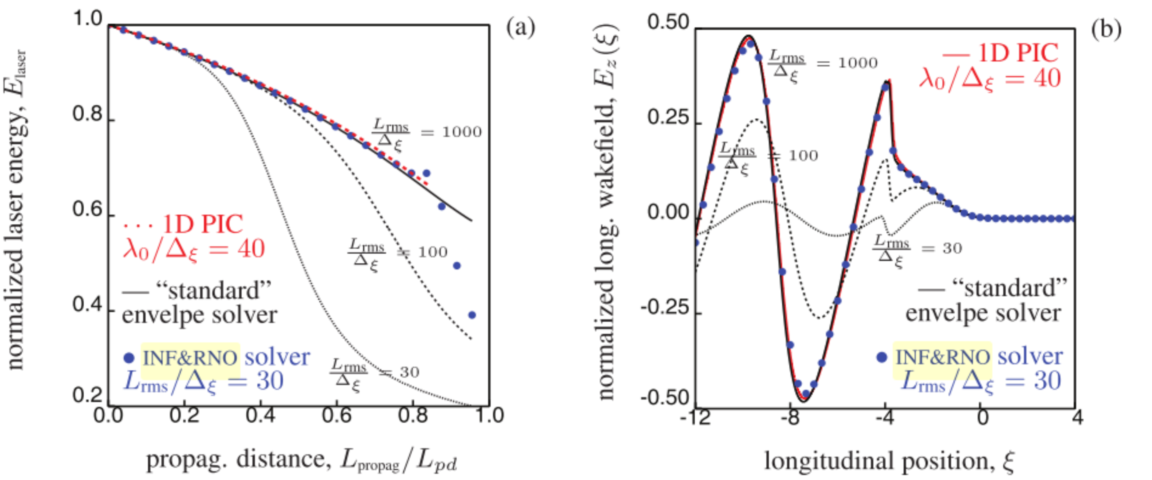


Figure 5.1.1.: Benchmark simulation for the different envelope solvers, in 1D. The laser-plasma parameters are $a_0 = 1$, $L_{rms} = 1$ and $k_0/k_p = 100$.
(a) Laser energy evolution, as a function of the propagation distance, normalized to the laser pulse pump depletion length $l_{pd} \sim \lambda_p \left(\frac{k_0}{k_p}\right)^2$.
(b) Lineout of the longitudinal wakefield, E_z, after a propagation distance corresponding to $0.8 \cdot l_{pd}$
In both the boxes the red dashed line is the reference result obtained with the full PIC code ALaDyn (red solid line). The black lines are the results obtained with the "standard", Cartesian form, envelope solver at different resolutions, $L_{rms}/\Delta_\zeta = 30$, 100, 1000 (black dashed lines). The blue points are the result obtained with the INF&RNO polar form envelope solver for $L_{rms}/\Delta_\zeta = 30$. Figure courtesy of C. Benedetti [28].

In INF&RNO, new temporal steps of the envelope $\hat{a}^{n+1}$ are computed by using the Crank-Nicholson discretization Eq. 5.1.2, knowing $\hat{a}^{n-1}$ and $\hat{a}^{n}$.

The right-to-left dependency is introduced by the presence of the longitudinal derivative ∂_ζ. The resolutive algorithm we use for obtaining $\hat{a}^{n+1}$ is simply to integrate from right boundary (where the envelope field is zero) to the left. The right to left integration allows to treat the longitudinal derivative term as a known term, breaking the integration into a chain of 1D linear systems in the transverse coordinate that must be solved serially. Since the longitudinal derivative acts in in Eq 5.1.1 only mixed with the temporal derivative ∂_τ, the envelope field $\hat{a}^{n+1}_{i-1,j}$ at time $n+1$ and at the longitudinal grid point $i-1$ depends on the previous local statuses of $\hat{a}^{n}_{i-1,j}$ and $\hat{a}^{n-1}_{i-1,j}$ and the only dependency on the points on the right is due to the temporal derivative $\partial_\tau \hat{a}$, discretized as $\left(\hat{a}_{ij}^{n+1} - \hat{a}_{ij}^{n-1}\right)/2\Delta_\tau$ (we omit for simplicity $\hat{a}_{i+1,j}^{n+1}$ following from the discretization using a second order upwind scheme)

$$\hat{a}_{i-1,j}^{n+1} = F\left(\hat{a}_{ij}^{n+1} - \hat{a}_{ij}^{n-1},\, \hat{a}_{i-1,j}^{n},\, \hat{a}_{i-1,j}^{n-1}\right) \tag{5.1.3}$$

This right-value dependency makes the laser integration part impossible to parallelize longitudinally, as it is impossible to break the dependency chain and split the computation among processing units. Leaving it un-parallelized would result in a very poor scalability (of fundamental importance for a modern code, see section 5.2), and the maximum reachable speedup would have been $< 10x$ (a simple estimate using Ahmdal's law).

Slice by slice, the form of F that follows from Eq 5.1.3 requires to solve tridiagonal linear systems *transversally.* As a result, the code has dependencies that parallelization hard along both the transversal and the longitudinal directions.

An iterative scheme for solving Eq. 5.1.1, via Eq. 5.1.3, has been implemented in INF&RNO in order to solve the recurrence and enable parallelism.

Since the envelope field is slowly varying in time, it is reasonable to assume that its temporal derivative varies slowly in time $\hat{a}_{ij}^{n+1} - \hat{a}_{ij}^{n-1} \sim \hat{a}_{ij}^{n-1} - \hat{a}_{ij}^{n-3}$. In fact, the difference between these two terms is a finite difference approximation of the second derivative in time of $\hat{a}$, which is a small factor in our conditions, important only for describing backward-propagating waves (as seen in section 5.1.1). The iterative scheme starts by using a previous temporal derivative at the right points in place of unknown $\hat{a}_{ij}^{n+1} - \hat{a}_{ij}^{n-1}$, as the an initial guess allowing to compute a rough estimate g_1 for $\hat{a}_{ij}^{n+1}$ for all grid points i in parallel, using F. The newly computed $g_1$can then provide another approximation for each point on the right $\hat{a}_{ij}^{n+1} - \hat{a}_{ij}^{n-1}$, and the process can be iterated until the succession g_k converges.

$$\begin{cases} g_{i-1,j;1}^{n+1} = F\left(\hat{a}_{ij}^{n-1} - \hat{a}_{ij}^{n-3},\, ...\right) & \textit{initial guess} \\ \\ g_{i-1,j;k}^{n+1} = F(g_{i,j;k-1}^{n+1} - \hat{a}^{n-1},\, ...) & k-th\ iteration \\ \\ \hat{a}_{ij}^{n+1} = g_{ij;P}^{n+1} & final \end{cases}$$

The iterations are stopped when a convergence criterion comparing successive g_k is

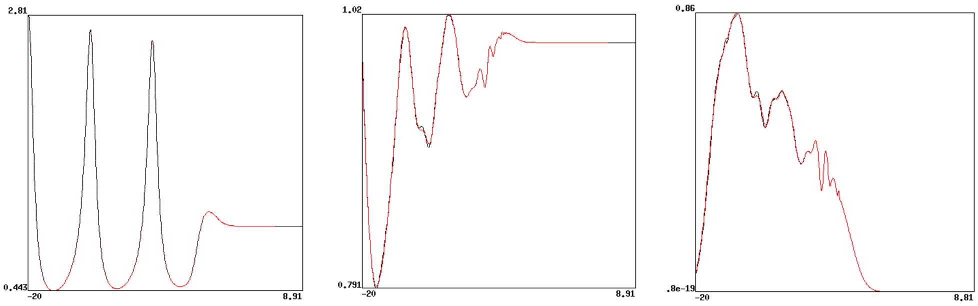


Figure 5.1.2.: Validation of the envelope solver parallel/iterative algorithm: longitudinal line-outs of the proper density before laser depletion (left), density with the depleted laser (center), depleted laser envelope modulus (right). Laser plasma parameters are $a_0 = 1, w_0 = 5, L = 2, k_0/k_p = 20$

met. Due to the fact that the envelope is slow-varying, even at low resolutions and with the laser deep into depletion, a low number of iterations (~ 10) manages to find a very accurate solution, as seen in the lineouts in figure 5.1.2. Since any successive approximation g_k can be computed in parallel for each longitudinal grid point i (the recurrence $\hat{a}^{n+1}_{i-1,j} = F(\hat{a}^{n+1}_{ij}...)$ becomes $g^{n+1}_{i-1,j;k} = F(g^{n+1}_{i,j;k-1}...)$, in which the iteration index k solves the dependency), we were able to get rid of any parallelization/scalability issue at the price of introducing some cheap iterations that scale well (in the PIC, the particle interpolation/deposition dominates the CPU time by far).

5.1.3. Plasma motion and wakefield equations in cylindrical comoving coordinates

In INF&RNO, cylindrical symmetry is assumed around the z axis for plasma phase space distribution and wakefield fields (the direction of laser propagation), as for the envelope field.

Let us define these non-dimensional, plasma wavelength normalized, comoving r-z cylindrical coordinates and variables:

$$\begin{cases} \xi = k_p\,(z-ct) & ,\; k_p = \omega_p/c \\ r = k_p r & \\ \tau = \omega_p t = k_p ct & \\ \delta = \frac{n}{n_0} & (normalized\, density) \end{cases}$$

The laser envelope evolves at timescales much longer than a single laser oscillations. The force acting on a plasma electron can be therefore averaged in time in this scheme, using the *ponderomotive* approximation for the force [1]. In this model, the plasma responds to the driver ponderomotive force generating an electromagnetic wakefield and consistently perturbing the driver propagation modifying the proper density in Eq. 5.1.1.

The relativistic electromagnetic *wakefield* is described by the electromagnetic fields $\mathbf{E}_{wake}$, $\mathbf{B}_{wake}$ (the suffix wake is further omitted for brevity), evolving according to Ampere-Maxwell laws:

$$\begin{cases} \frac{\partial \mathbf{E}_{wake}}{\partial t} = c\nabla \times \mathbf{B}_{wake} - 4\pi \mathbf{J} \\ \frac{\partial \mathbf{B}_{wake}}{\partial t} = -c\nabla \times \mathbf{E}_{wake} \end{cases} \rightarrow_{cyl} \begin{cases} \frac{\partial E_z}{c\partial t} & = \frac{1}{r}\frac{\partial r B_\phi}{\partial r} - \frac{1}{r}\frac{\partial B_r}{\partial \phi} - \frac{4\pi}{c} J_z \\ \frac{\partial E_r}{c\partial t} & = \frac{1}{r}\frac{\partial B_z}{\partial \phi} - \frac{\partial B_\phi}{\partial z} - \frac{4\pi}{c} J_r \\ \frac{\partial E_\phi}{c\partial t} & = \frac{\partial B_r}{\partial z} - \frac{\partial B_z}{\partial r} - \frac{4\pi}{c} J_\phi \\ \ldots & \\ \ldots & \\ \ldots & \end{cases}$$

, where J are normalized current densities. To maintain the radial symmetry of the plasma distribution (the laser envelope acts only with the radially symmetric ponderomotive force), the wakefield electromagnetic fields are constrained to be $E_\phi = 0, B_z = 0, B_r = 0$, in every point of the domain. This simplifies our system, which becomes:

$$\begin{cases} \frac{\partial E_z}{c\partial t} &= \frac{1}{r}\frac{\partial (rB_\phi)}{\partial r} - \frac{4\pi}{c}J_z \\ \frac{\partial E_r}{c\partial t} &= -\frac{\partial B_\phi}{\partial z} - \frac{4\pi}{c}J_r \\ \frac{\partial E_\phi}{c\partial t} &\equiv 0 \\ \\ \frac{\partial B_z}{c\partial t} &= \frac{\partial E_\phi}{\partial r} \equiv 0 \\ \frac{\partial B_r}{c\partial t} &= -\frac{\partial E_\phi}{\partial z} \equiv 0 \\ \frac{\partial B_\phi}{c\partial t} &= -\frac{\partial E_r}{\partial z} + \frac{\partial E_z}{\partial r} \end{cases} \rightarrow_{\xi = k_p(z+ct)} \begin{cases} \frac{\partial E_z}{\partial \tau} &= \frac{\partial E_z}{\partial \xi} + \frac{1}{r}\frac{\partial (\rho B_\phi)}{\partial r} - j_z \\ \frac{\partial E_r}{\partial \tau} &= \frac{\partial (E_r - B_\phi)}{\partial \xi} - j_r \\ \frac{\partial B_\phi}{\partial \tau} &= -\frac{\partial (E_r - B_\phi)}{\partial \xi} + \frac{\partial E_z}{\partial r} \end{cases}$$

In INF&RNO, the background plasma can be modeled using either a fully kinetic PIC or a cold fluid description (that does not describe wave-breaking and self-injection) and the laser-plasma coupling is described via the ponderomotive approximation. The relativistic equations of motion of an electron of momentum $\mathbf{p}$, in the *envelope/ponderomotive* an approximation, read:

$$\begin{cases} \frac{d\mathbf{p}}{dt} = q\left(\mathbf{E}_{wake} + \frac{v\times \mathbf{B_{wake}}}{c}\right) - \frac{mc^2}{2\gamma}\nabla\left|\frac{q\hat{A}}{mc^2}\right|^2 \\ \gamma = \sqrt{1 + \frac{p^2}{m^2c^2} + \left|\frac{q\hat{A}}{mc^2}\right|^2} \end{cases} \tag{5.1.4}$$

In normalized cylindrical coordinates, the equations of motion for electrons become:

$$\begin{cases} \gamma &= \sqrt{1 + \left|\hat{a}\right|^2/2 + u_z^2 + u_r^2} \\ \\ \frac{d\xi}{d\tau} &= \frac{u_z}{\gamma} - 1 \\ \frac{dr}{d\tau} &= \frac{u_r}{\gamma} \\ \\ \frac{du_z}{d\tau} &= -\frac{\partial\gamma}{\partial\xi} - E_z - \frac{u_r}{\gamma}B_\phi \\ \frac{du_r}{d\tau} &= -\frac{\partial\gamma}{\partial r} - E_r - \frac{u_z}{\gamma}B_\phi \end{cases}.$$

In the PIC model, the quasi-particle dynamics follows from the electrons' equations of motion 5.1.4.

In the fluid model, the equations governing the evolution of the fluid plasma model can be obtained starting from the relativistic plasma cold fluid equations ($\mathbf{u} = \gamma\mathbf{v}/c$):

$$\begin{cases} \frac{\partial n}{c\partial t} + \nabla\cdot(n\mathbf{u}/\gamma) & = 0 \\ \left(\frac{\partial}{c\partial t} + \left(\frac{\mathbf{u}}{\gamma}\right)\cdot\nabla\right)\mathbf{u} & = \mathbf{F} \end{cases}$$

Changing variables to the comoving system $\frac{\partial}{\partial t} = \frac{\partial}{\partial \tau} - \frac{\partial}{\partial \xi}$, and recalling 5.1.4, one obtains the fluid equations used in INF&RNO-fluid:

$$\begin{cases} \gamma_{fluid} & = \sqrt{1 + |\hat{a}|^2/2 + u_z^2 + u_r^2} \\ \\ \frac{\partial\delta}{\partial\tau} & = \frac{\partial\delta}{\partial\xi} - \nabla\cdot\left(\frac{\mathbf{u}}{\gamma_{fluid}}\delta\right) \\ \frac{\partial(\delta u_j)}{\partial\tau} & = \frac{\partial(\delta u_j)}{\partial\xi} - \nabla\cdot\left(\overrightarrow{\beta}\,\delta u_j\right) + \delta\left[-\left(\mathbf{E} + \frac{\mathbf{u}}{\gamma_{fluid}}\times\mathbf{B}\right) - \frac{1}{2\gamma_{fluid}}\nabla\left(\frac{|\hat{a}|^2}{2}\right)\right]\ ,\ j = z, r \end{cases}$$

In INF&RNO, an explicit fourth order Runge-Kutta integrator is used for the Ampere-Maxwell wakefield and the quasi-particle motion.

The fields are discretized on un-staggered, 2D, regular z, r grids. The longitudinal ζ derivatives are computed with a second-order upwind scheme ($\partial_{\zeta;\Delta\zeta} f_{i,j} = (-3f_{i,j} + 4f_{i+1,j} - f_{i+2,j})/(2\Delta\zeta)$), while the transversal one with a second-order centered scheme. The motivation of using upwind derivatives is to damp some backward propagating instabilities. In the PIC mode, force interpolation, charge density and current deposition are performed using quadratic shape functions. Compact low-pass digital filters [37] and standard binomial filters (with compensator) are available for field and current smoothing.

5.2. Parallelization and scalability benchmarks

INF&RNO is a reduced model code, but simulations can still take several years of CPU time in order to be completed at higher resolutions. Efficient parallelization, together with accurate and reduce numerical models, was therefore necessary in order to perform simulations of 10 GeV scale laser-plasma acceleration stages.

The key quality for a parallel code is nowadays the scalability, as, if the performance of single scalar units ceased to increase, the number of processing core per CPU and the number of computing nodes connected in supercomputers still increase with a good scaling over the years. Hence, good scalability gives the possibility to benefit from the

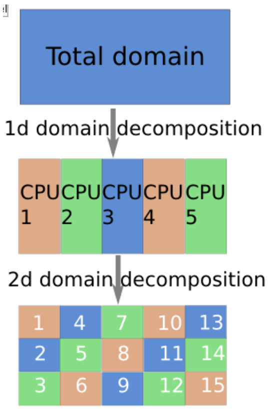


Figure 5.2.1.: Parallelization using domain decomposition in 1D or 2D.

future high performance computing advances.

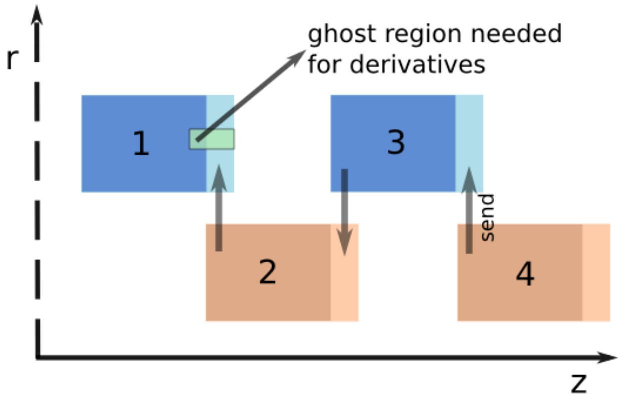
The first requirement for scalability is imposed by Ahmdal's Law: if a code contains a serial part that takes a fraction t_{serial} of the total runtime, the maximum achievable speedup is bound at performance multipliers$\sim 1/t_{serial}$, also for $N_{processors} \to \infty$. In fact, the time for running a simulation scales as $\lim_{N_{procs}\to\infty} t = \frac{t_{parallelizable}}{N} + t_{serial} = t_{serial}$. Hence, every modern code should be designed in order to minimize serial parts. This is the motivation that leaded us to look far an envelope integrator that solved the envelope Crank-Nicholson equation 5.1.2 without using a non-parallelizable recurrence (subsection5.1.2).

5.2.1. 1D domain decomposition

Thanks to the parallel, iterative envelope solver presented in subsection5.1.2, all the longitudinal grid operations in INF&RNO are local stencils, deriving from finite difference discretization. Their locality allows for longitudinal domain decomposition, splitting the computational domain into smaller subdomains, each one evolving almost independently. The only communication required is for exchanging a few boundary grid slices. PIC quasi-particles can be subdivided across the same subdomains, with the difference that they have to be reassigned once they cross a subdomain boundary (they are in principle free to move inside the entire domain, but the Courant timestep for the fields limit the range in a time step to be less than the cell size) and extra ghost cell must be allocated for interpolating fields and deposing densities.

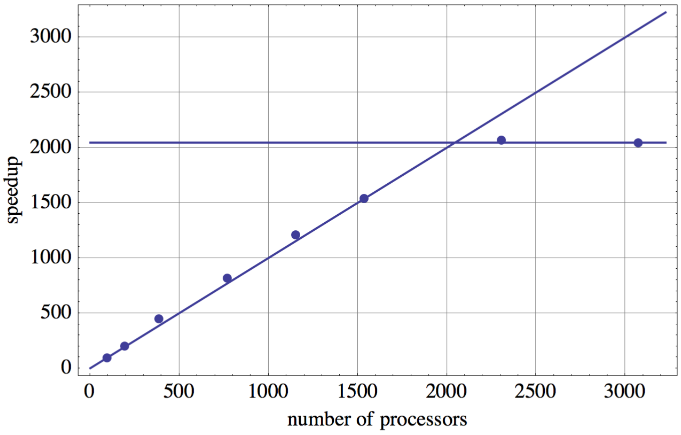
INF&RNO is parallelized with 1D domain decomposition using standard message passing interface to exchange field boundaries and particles crossing domain. Each physical subdomain is computed by a different computational node. For typical laser-plasma simulations, the longitudinal direction is the one with more grid points N_z.

In its longitudinal upwind derivative scheme $\partial_{\xi;\Delta\xi} f_{i,j} = (-3f_{i,j}+4f_{i+1,j}-f_{i+2,j})/(2\Delta\xi)$, the values required for computing f'_{ij} are the two points on the right, $f_{i+1,j}$,$f_{i+1,j}$. Before every derivative computation, one has to be sure that the ghost region contains the right data, which must be copied from the right neighbor node, as illustrated in the following figure. As mentioned before, the PIC scheme may require additional ghost cells if high order shape functions are used. Parallel codes with boundary data exchange scale well until the time for boundary communication (cell points and crossing particles) dominates the computation time. In fact, for a fixed simulation, the total computation scales as $O(1)$ versus the number of parallel units N, the total communication scales as $O(N)$, the computation per node decreases as $\sim 1/N$, but the communication per node stays constant.



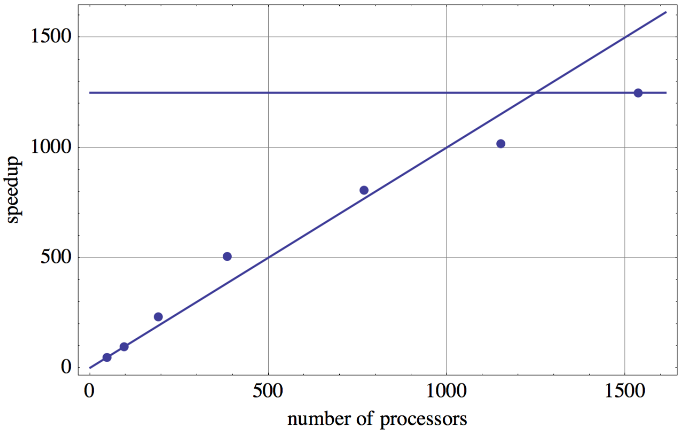
The validation and debugging of the parallelized code was performed comparing results of various simulation runs (see figure 5.2.2 for a simple example), value by value.

The scalability was measured both in terms of weak scaling (runtime scalability increasing number of parallel units keeping the subdomain computational size fixed) and strong scaling (runtime scalability increasing the number of parallel units and keeping the total problem size fixed). On a modern cluster, the fluid only code strong scaling is excellent up to a number of cores $^{\sim}\frac{1}{2}$ of the grid points, as can be seen for the following figure, for $Nz = 4224$.



Fluid scaling (Nz=4224)

Also the PIC code scales well, up to the limit in which only a very few (2-3) grid z-slices are assigned per node. The following figure shows its weak scaling graph for $Nz = 1968$



PIC scaling (Nz=1968)

5.2.2. 2D domain decomposition and shared memory parallelization

Efficient 1D parallelization (subsection 5.2.1) takes INF&RNO scalability close to the maximum 1D limit, i.e. assigning one slice (plus ghost cells) per subdomain.

Besides the maximum number of subdomains, another limitation of the 1D decomposition is the fact that the boundary/domain size ratio scales linearly with the number of domain slices $N_{p;z}$, and this can result in poor scalability. 2D domain decomposition

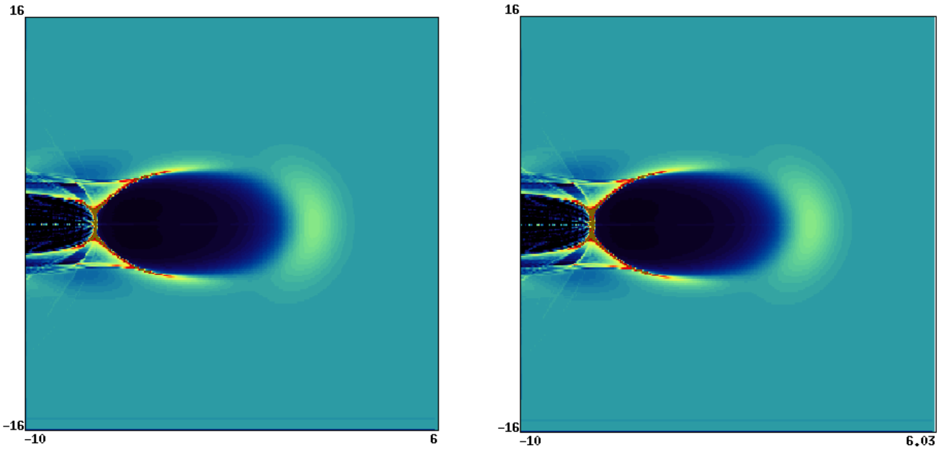
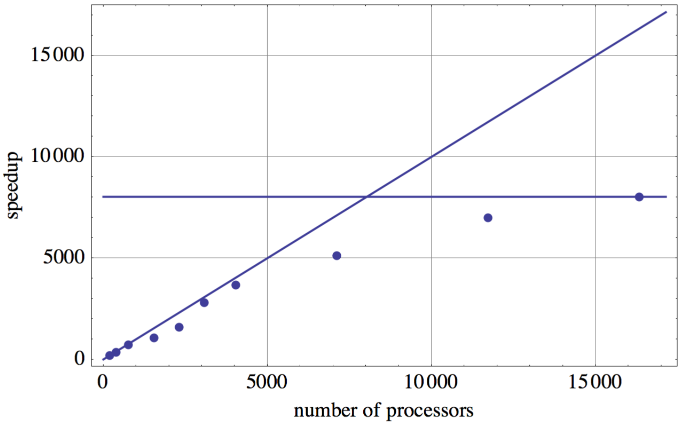


Figure 5.2.2.: Comparison between a parallelized PIC simulation (40 minutes on 120 cores) result and scalar one (38 hours). Laser plasma parameters $a_0 = 3.0$, $L = 2$, $w_0 = 2\sqrt{a_0}$.

(splitting domains also in the radial direction) could be a solution for both the limitations, allowing to scale to a larger number of subdomains, as the grid to split has now $N_z \times N_r$ as size.

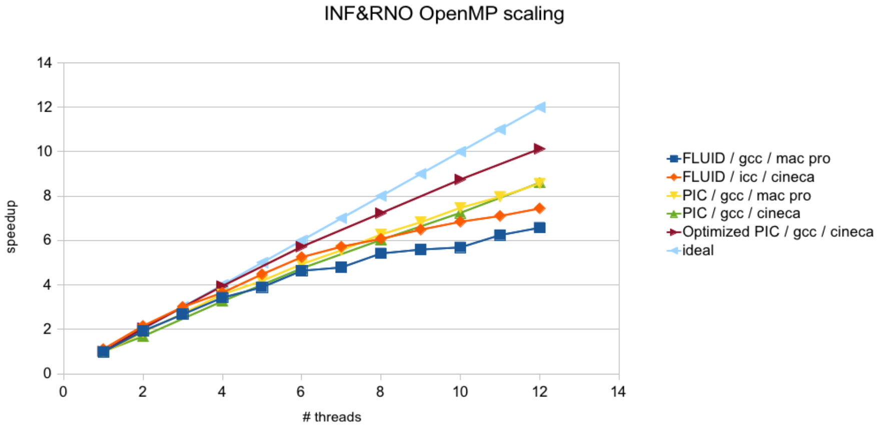
In INF&RNO, the parallelization along the transverse direction presents some difficulties due to the N_z "transverse" tridiagonal systems present in the envelope equation implicit numerical scheme. Tridiagonal system can be solved serially using the very efficient Thomas algorithm, and, while we are still looking for a better way to handle these systems, the best compromise found so far for our case is using a *pipelined* solving algorithm. After the longitudinal domain decomposition, each node has $N_z/N_{p;z}$ tridiagonal systems, each one involving all the N_r transverse points in the slice. Each "system" is then broken in $N_{p;r}$ parts, but the Thomas algorithm introduces a dependency amongst the sequence of parts. The pipelining approach consists in starting all computations as soon as all the dependencies are met (e.g. after solving the first part of the system, for the first z-slice, the first node begins the computation of the first part second z-slice, and so on).

Practically, the strong scalability plot for the PIC code, using the 2D decomposition, shows that it is possible to run on more processors than allowed by the longitudinal-only decomposition (in this case, the maximum theoretical speedup was $N_z = 1968$), still at a reasonable efficiency:



PIC 2d parallelization scaling (Nz=1968)

Another option would be to parallelize each slice using OpenMP shared memory parallelization, without domain decomposition. Some preliminary studies where carried out parallelizing the whole simulation using OpenMP. The density deposition race condition, arising from shared memory parallelization, can be avoided with the use of atomic operations or using a single buffer per core. The OpenMP version of the code reaches good scalability up to 12 cores.



5.2.3. INF&RNO GPU Parallelization

The GPU port of the numerical schemes implemented in INF&RNO is currently under development.

Since the Runge Kutta 4 time integrator uses 4 substeps for the particles, keeping a strictly spatially organized data structure for the particles, as required by the deposition algorithm in jasmine (section 4.1.1), can introduce a significant overhead. Therefore, another, more relaxed, approach for implementing densities deposition has been chosen.

Densities are summed on the global memory density grid with the help of global memory atomic operations. The number of atomic operations is dramatically reduced by performing a warp segmented scan, which groups the global memory access for particles laying in the same cell and at contiguous indices in the particle array. This way, a great part of the global memory accesses can still be "cached", but it is no longer necessary to update the spatial sorting scheme each Runge-Kutta sub-time step. The warp scan is implemented using Kepler architecture "shuffle" instructions.

The densities deposition kernel has been fused with the interpolation/push kernel (i.e. it computes both densities and the particles' status Runge-Kutta temporal derivatives streaming the particle array just once). For typical LWFA simulation parameters, the performance of the fused kernel, on a NVIDIA Kepler K40 GPU, ranges, from $7ns$ on average, per particle (1 particle per cell) to $3ns$ on average, per particle ($>=$ 8 particles per cell).

6. INF&RNO/Quasi-Static

6.1. The Quasi-static Approximation

The quasi-static approximation (QSA) was first applied to nonlinear laser-plasma interaction by Sprangle et al. in [45, 46]. Considering a wakefield driver (either a laser pulse or a ultra relativistic beam) propagating in a tenuous plasma, The QSA separates the timescales of the driver evolution and of the electrons in the wake, assuming that during the time it takes the driver to transit a plasma electron, the driver does not evolve significantly.

If the driver propagates at relativistic speed, the time for crossing a plasma electron is approximately $\tau_L = L$, where L is the normalized driver characteristic length. In order for the plasma electrons to experience a static (independent from τ) driver ponderomotive field (or, in order to the QSA assumption to be satisfied), its evolution timescale must satisfy $\tau_E \ll \tau_L$.

There are two intrinsic time scales associated with Plasma Wakefield Accelerator PWFA and Laser Wakefield Acceleration LWFA. In both regimes, the wakefield driver characteristic length is usually of the order of the plasma wavelength ($L \sim 1$), in order to maximize the amplitude of the wakefield. In PWFA, the drive beam is ultra-relativistic ($\gamma_{beam} > 2000$) and it evolves on the scale of the betatron wavelength, which is $(2\gamma_{beam})^{1/2}$ times longer than the plasma wavelength. In LWFA, the natural scale of evolution of the laser beam is the Rayleigh (diffraction) length $z_R = \frac{1}{2}\frac{k_0}{k_p}w_0^2$, where w_0 is the laser spot size. Since, typically, $z_R \gg 1$, the plasma electrons experience a static laser field.

In the quasi static approximation, the plasma response is calculated by freezing the driver and its ponderomotive fields, neglecting the temporal derivatives ∂_τ in the plasma fluid or kinetic equations, which determine the plasma response to the driver ponderomotive force map. The QSA plasma equations can be used to determine the response as a function of the driver ponderomotive field and the spatial coordinates.

The temporal derivatives are retained in the equations governing the evolution of the driver, and, at each driver evolution “timestep”, the currents and densities are obtained from the quasi-static response for a fixed τ. In other words, numerically, the QSA can be used to simulate the driver-plasma interaction in a iterative, two-step, manner. A QSA iteration is composed by the computation of the wakefield response at fixed τ and the evolution of the driver in τ, for which the response from the first step is used.

Due to the scale separation, the QSA is a very useful approximation for writing efficient numerical codes, as it enables a time-step size which can be much greater than those in full PIC codes. This was successfully demonstrated in the ponderomotive relativistic quasi-static PIC codes WAKE [47, 48] (originally a 2D cylindrical or cartesian code) and QuickPIC [49, 50], a 3D PIC code using pipelining for achieving impressive parallelization results. The 3D quasi-static code HiPACE [51], highly specialized for PWFA, is the first one parallelized in all three dimensions.

In this chapter, we describe the implementation of the QSA in the ponderomotive code INF&RNO [26, 27, 28], covering the numerical scheme in cylindrical symmetry, and its numerical stability issues (6.2), the parallelization using pipelining (6.3) and a load balancing technique that improves the scalability in high intensity cases, for which the iteration load of our quasi-static solver produce a significant load imbalance in across longitudinal domains (6.4).

6.2. Quasi-static numerical scheme in cylindrical symmetry

The QSA is applied starting from the explicit PIC equations in cylindrical, co-moving coordinates used in INF&RNO:

$$\begin{cases} \frac{\partial E_z}{\partial \tau} = & \frac{\partial Ez}{\partial \zeta} + \frac{1}{r}\frac{\partial (rB_\phi)}{\partial r} - J_z \\ \frac{\partial E_r}{\partial \tau} = & \frac{\partial (E_r - B_\phi)}{\partial \zeta} - J_r \\ \frac{\partial B_\phi}{\partial \tau} = & -\frac{\partial (E_r - B_\phi)}{\partial \zeta} + \frac{\partial E_z}{\partial r} \end{cases} \tag{6.2.1}$$

$$\begin{cases} \forall j \in 1,2,...,N_p \\ \frac{d\zeta_j}{d\tau} = \beta_{z,j} - 1 \\ \frac{dr_j}{d\tau} = \beta_{r,j} \\ \frac{du_{z,j}}{d\tau} = -\frac{\partial\gamma_j}{\partial\zeta} - E_z - \beta_r B_\phi \\ \frac{du_{r,j}}{d\tau} = -\frac{\partial\gamma_j}{\partial r} - E_r + \beta_z B_\phi \\ \gamma_j = \sqrt{1 + |\hat{a}|^2/2 + u_{z,j}^2 + u_{r,j}^2} \end{cases} \tag{6.2.2}$$

Fields and currents (generically, any eulerian quantity Q) can be assumed to be frozen, or quasi-static, during plasma evolution in the comoving, speed-of-light frame: $\partial_\tau Q \simeq 0$ (or $\partial Q_t \simeq -\partial_\zeta Q$). Taking $\frac{\partial E_z}{\partial\tau} = 0$, $\frac{\partial E_r}{\partial\tau} = 0$, $\frac{\partial B_\phi}{\partial\tau} = 0$, Eq 6.2.1 becomes:

$$\begin{cases} \frac{\partial E_z}{\partial\zeta} + \frac{1}{r}\frac{d(rB_\phi)}{dr} - J_z = & 0 \\ \frac{\partial(E_r - B_\phi)}{\partial\zeta} - J_r & = 0 \\ \nabla_\perp^2 E_z = \frac{1}{r}\frac{d}{dr}(rJ_r) & = 0 \end{cases} \cdot \tag{6.2.3}$$

The QSA for plasma macro particles introduces the invariant of motion [51]:

$$\gamma - \psi - u_z = 1 \tag{6.2.4}$$

, as follows from applying the QSA on the Hamiltonian for a plasma electron in the wake potential $\psi = e\phi/mc^2 - a_z$.

In accordance with the QSA principle, for all quasi-static particles $\beta_z < 1$, and hence their longitudinal trajectories $\zeta(\tau)$ are strictly monotonic in τ. It is therefore possible to introduce the change of variable $x_j(\tau) = x_j(\zeta)$ for all the particle quantities, changing the temporal variable to the comoving one. Applying the chain rule $\partial_\tau \to \partial_\tau\zeta{\cdot}\partial_\zeta = (\beta_z - 1)\partial_\zeta$ and Eq 6.2.4 (in the form $\beta_z - 1 = (1+\psi)/\gamma$), the particle equations are recast to 6.2.2:

$$\begin{cases} \forall j \in 1,2,...,N_p \\ \frac{dr_j}{d\zeta} = -\frac{u_{r,j}}{1+\psi_j} \\ \frac{du_{r,j}}{d\zeta} = \frac{F_{laser} + \gamma(E_r - B_\phi)}{1+\psi} + B_\phi \\ \frac{d\psi}{d\zeta} = \frac{u_{r.j}}{1+\psi}(E_r - B_\phi) - E_z \\ \gamma_j - u_{z,j} - \psi_j = 1 \end{cases} \tag{6.2.5}$$

On the contrary, beams (driver beams or injected beams) are usually highly relativistic $\beta_z \simeq 1$, and hence the QSA does not apply for their dynamics and they must be treated separately from plasma particles, as they evolve in τ.

In INF&RNO, the quasi-static equations 6.2.3 and 6.2.5 are integrated numerically backwards in ζ, from $\zeta = \zeta_{max}$, where the driver fields are zero and the particles at rest, towards ζ_{min}.

During an integration in ζ (a "swipe"), when computing the ζ_k slice, the state of the system is fully known on the right, for $k' > k$. The integration of the particle equations, performed using a fifth order Adams-Bashforth method, provides $x_j(\zeta_{k-1})$, using $x_j(\zeta_k)$, the fields at ζ_k for all particle quantities x. Currents are then computed at ζ_{k-1} using $x_j(\zeta_{k-1})$. The fields are computed at ζ_{k-1} using $J_z(\zeta_{k-1})$ and $J_r(\zeta_{k-1})$. $E_r - B_\phi$ is integrated in ζ with a simple longitudinal integration and E_z is computed by inverting the transverse laplacian in Eq 6.2.3/3, discretizing the operator a second order numerical derivative resulting in a tridiagonal system. A recurrence equation is then used for solving Eq 6.2.3/1 and obtain B_ϕ.

The integration of this system is numerically unstable, with an instability arising from the presence of both the terms E_z and B_ϕ in Eq 6.2.3/1. In INF&RNO, the instability is resolved using an iterative method, in which a succession of magnetic field solutions $B_\phi^n(\zeta_k)$ "adiabatically" converges to the $B_\phi(\zeta_k)$ solution, for each slice ζ_k. During the first iteration $B_{\phi,0} = 0$. Then, for each iteration n, the particles equations are integrated for obtaining $x_j(\zeta_{k-1})$ using the fields at ζ_k, and, in particular, $B_\phi^n(\zeta_k)$:

$$\begin{cases} \forall j \in 1, 2, ..., N_p \\ \frac{dr_j}{d\zeta} = -\frac{u_{r,j}}{1+\psi}_j \\ \frac{du_{r,j}}{d\zeta} = \frac{F_{laser}+\gamma(E_r-B_\phi)}{1+\psi} + B_\phi^n \\ \frac{d\psi}{d\zeta} = \frac{u_{r.j}}{1+\psi}(E_r - B_\phi) - E_z \\ \gamma_j - u_{z,j} - \psi_j = 1 \end{cases} \tag{6.2.6}$$

The longitudinal field $E_z(\zeta_{k-1})$ is then computed using Eq 6.2.3/3 ($\nabla_\perp^2 E_z = \frac{1}{r}\frac{d}{dr}(rJ_r)$).

A, new, consistent, magnetic field $B_\phi^*(\zeta_k)$ is then computed by solving Eq 6.2.3/1 at ζ_k, discretized using a centered derivative in E_z that uses the newly computed $E_z(\zeta_{k-1})$.

$$\frac{\partial E_z}{\partial \zeta} + \frac{1}{r}\frac{d(rB_\phi^*)}{dr} - J_z = 0. \tag{6.2.7}$$

The next $(n+1)$ iteration magnetic field $B_\phi^{n+1}(\zeta_k)$ is obtained smoothly blending the $n-th$ iteration magnetic field and the newly computed one:

$$B_\phi^{n+1} = \alpha B_\phi^n + (1-\alpha) B_\phi^* \, (\alpha \simeq 0.98). \tag{6.2.8}$$

The iterations are continued until the convergence criterion $\max_r(B_\phi^{n+1} - B_\phi^n) < t$ is satisfied. Once the scheme has converged, the new $(E_r - B_\phi)(\zeta_{k-1})$ is computed integrating Eq 6.2.3/2 .

The convergence speed of the method can be improved by replacing the magnetic field e 6.2.7 and blending equations 6.2.8, by

$$\nabla_r^2 B_\phi^n = \partial_\zeta J_r + \partial_r J_z$$

, in which a smoothing term is added to the laplacian operator:

$$(\nabla_r^2 + k) B_\phi^{n+1} = (\partial_\zeta J_r + \partial_r J_z) + k B_\phi^n \tag{6.2.9}$$

The introduction of the smoothing term can be seen as the introduction of a “support” temporal derivative $k\left(B_\phi^{n+1} - B_\phi^n\right) = -\nabla_r^2 B_\phi^{n+1} + \partial_\zeta J_r + \partial_r J_z$, as in Ref. [49].

This way, high frequency modes (for which $\nabla_r^2 \gg k$) are instantaneously propagated, while the propagation of low frequency modes, the ones that give rise to the instability (being responsible for instantaneous signal propagation) happens slowly.

Figure 6.2.1 shows the convergence of both the iterative schemes to the explicit PIC solution, computed with INF&RNO with adiabatic initialization. The figure shows the longitudinal electric field of a bubble wake generated by a non-evolving gaussian pulse with parameters $a_0 = 3.5,\ 5.0,\ L = 2,\ W = 2\sqrt{a_0}$, computed using the multi-frequency blending and resolution $d\zeta = 1/100$ (blue lines), uniform blending with resolution $d\zeta = 1/100$ (red lines), the explicit PIC solution (green lines) and uniform blending with double resolution $d\zeta = 1/200$. We notice that the quasi-static scheme is able to reproduce the explicit PIC results, even in this very non-linear bubble regime case. Both the quasi-static blending schemes converge to the same solution, but the simulation using the multi-frequency blending scheme was 10x faster than one using uniform blending. In these very non-linear cases, the quasi-static particles may cross very low wake potentials $\psi \simeq -1$ (that can result in trapping if the wake evolves). Since all particle equations contain a term $(1+\psi)^{-1}$ (the factor $\gamma/(1+\psi)$, introduced by the change of variables, can be interpreted as “the weight with which the temporal change contributes to the change along the comoving variable” [51]), singularities might happen both due to numerical errors and inconsistencies with the QSA approximation assumptions. INF&ERNO quasistatic detects these singularities and reruns part of the integration discarding the particles reaching a negative potential below a (dynamic) threshold value. Laser evolution (for short pulses the laser shape is approximately 0 where the singularity happens) and most of the wakefield formation are unaffected by this changes, which are mainly inconsistent only with phenomena that are inconsistent with the QSA in the first place (i.e. self-trapping and beam loading).

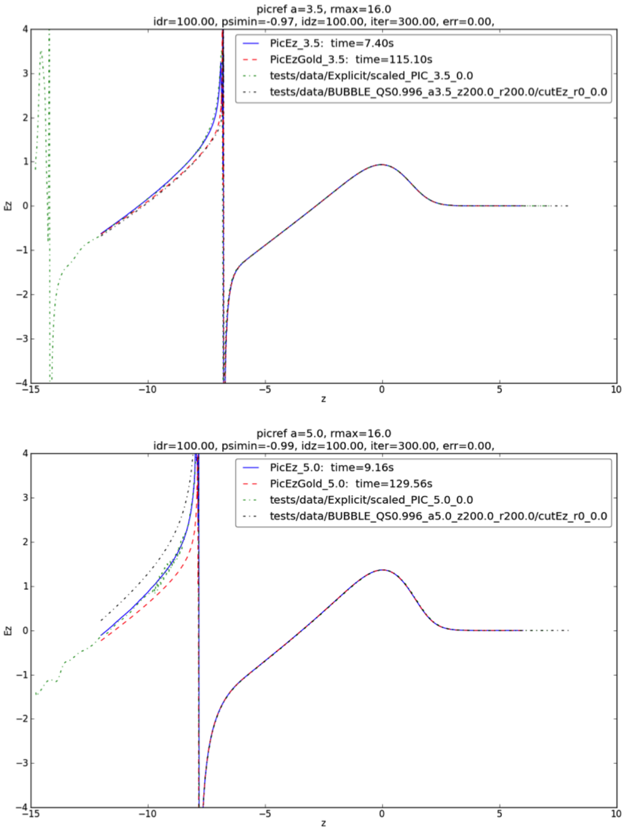


Figure 6.2.1.: Longitudinal electric field of a bubble wake generated by a non-evolving gaussian pulse with parameters $a_0 = 3.5,\ 5.0$, $L = 2$, $W = 2\sqrt{a_0}$, computed using the multi-frequency blending and resolution $d\zeta = 1/100$ (blue lines), uniform blending with resolution $d\zeta = 1/100$ (red lines), the explicit PIC solution (green lines) and uniform blending with double resolution $d\zeta = 1/200$. The quasi-static scheme successfully converge to the explicit reference solution, but is much more efficient. The simulation using the multi-frequency blending scheme was 10x faster than one using uniform blending.

6.3. Parallelization via pipelining

Parallelizing a quasi-static plasma code presents new challenges.

A single longitudinal integration in ζ is numerically time-like, and the computation of each slice in ζ depends on the solution of the ones at its right.

The code 3D HiPACE [51] manages to have efficient parallelization in all the two transverse dimensions, but in INF&RNO quasi-static we are limited by the fact that, having only the radial coordinate as a transverse dimension, the very few particles trajectories and grid points (only a thousand in total) we have to compute are not enough to saturate the parallelism required by highly parallel processor: the technique we tried (shared memory parallelization with OpenMP) has shown to stop to scale past 2-4 cores.

For simulations with driver evolution, one longitudinal quasi static integration is required after each driver evolution timestep (an "epoch"). The key idea is then to use the "pipelining" technique [49, 50].

The the longitudinal domain is divided in N (the number of processors) subdomains D_i, numbered by the index i from right to left. Each processor P_i is responsible for a subdomain D_i, but processors process different driver epochs (physical times).

Since the ζ integration happens from right to left, when the processor P_i finishes the computation of the domain D_i at time t_j, it can send its result (for fields, the boundary is enough) to the processor P_{i+1} and start the computation of $(D_i,\ t_{j+1})$. The processor P_{i+1} can then start the computation of $(D_{i+1},\ t_j)$. Hence, at each CPU "clock" time s_k, the processor P_i computes the domain/time $(D_i,\ t_k - i)$ and stays idle if $t_k < i$ (the time for filling the pipeline).

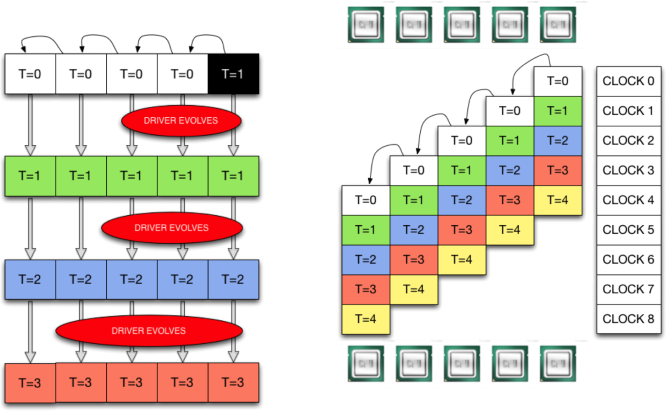


Figure 6.3.1.: Left: traditional computation scheme, all processors compute their subdomain at the same time.
Right: processors compute subdomains epochs in a pipeline.
Subdomain evolution depends on the status of the same subdomain in the previous driver epoch (for driver evolution) and the status of the right subdomain in the current epoch. Once the two dependency requirements are met, the subdomain is ready to be advanced to the next epoch. If the driver is subluminal, there are no dependencies on the left.

The dependencies that processor P_i needs to wait for computing (D_i, t_j), are the statuses of the driver and quasi-static wakefield locally at the previous epoch (D_i, t_{j-1}) and of the domain on the right at the current epoch (D_{i-1}, t_j) as shown in figure 6.3.1.

The mechanism works only if the non-quasi-static driver/beam evolution can be integrated in a right to left fashion, i.e. if the driver is subluminal and there is no information propagation traveling in the positive ζ direction. The envelope field solver (Subsection 5.1.2) in INF&RNO supports this way of splitting the integration in ζ subdomains.

6.4. Pipeline load balancing

The parallelization using pipelining provides ideal scalability if the time for transferring the subdomain boundaries across processors is smaller than the time required for computation and all the subdomains D_i require the same amount of CPU time to be computed.

Due to our iterative scheme (section 6.2), the latter condition is not satisfied for highly nonlinear cases $a_0 > 2$. In fact, the number of iterations required for the convergence of the scheme dramatically varies along the longitudinal coordinate ζ. As an example, the slices in proximity of the back E_z zero crossing point of a bubble wake require up to hundredths of times more iterations than the average. This kind of distribution of the number of iterations is the least favorable for achieving parallel speedups, as the processors computing the "difficult" slices slow down the entire pipeline, making the work of all other processors worthless. Figure 6.4.1 shows the issue in detail.

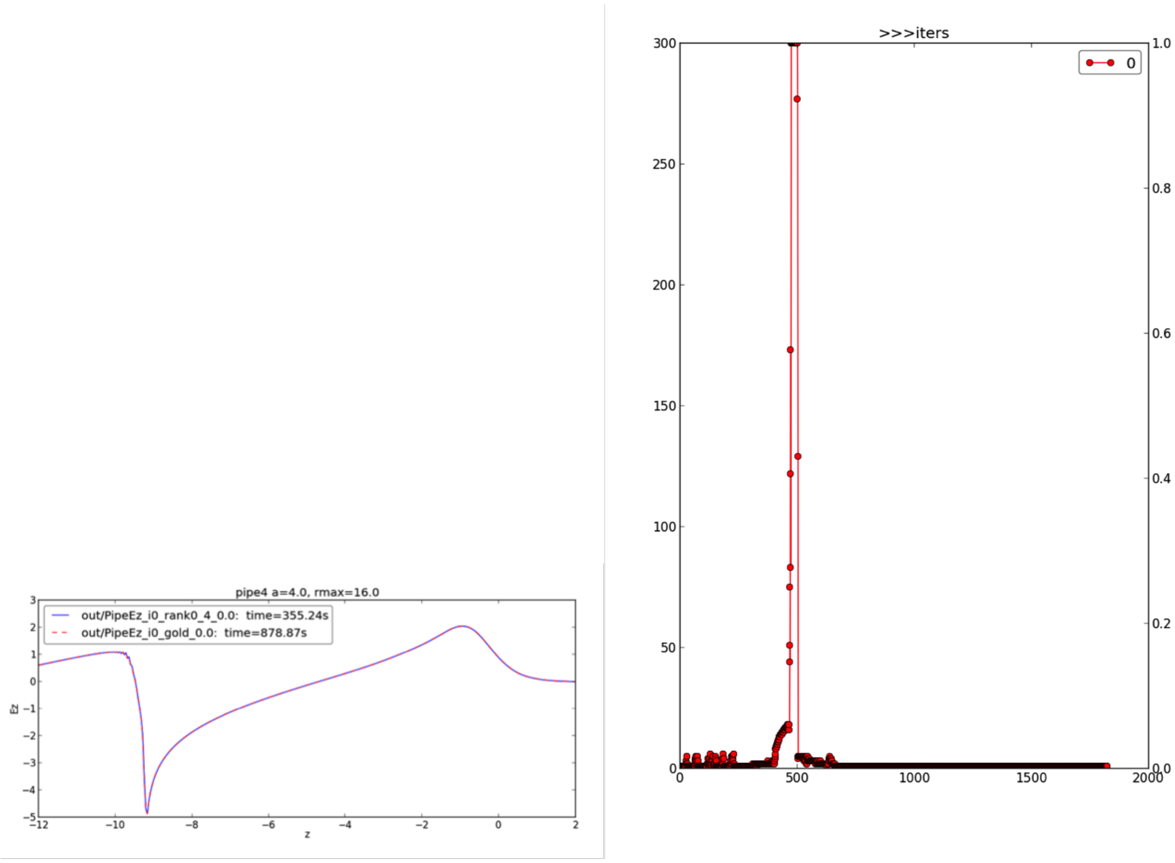


Figure 6.4.1.: Left: E_z lineout for laser-plasma parameters: $a_0 = 4, L = 2, W = 2\sqrt{a_0}$, computed serially and in parallel (pipelining). The right box shows the number of iterations (y-axis) versus ζ-slices (x-axis). More load/iterations for the processors/subdomains containing the bubble wake E_z zero crossing point located at the back of the bubble. Such subdomains slow the entire pipeline and the maximum speedup achievable is always less than 2x.

The solution we have implemented in INF&RNO/quasi-static is to dynamically resize the subdomain sizes in order to restore the load balance across processors.

Every other driver evolution step, the load of a processors is measured and compared with the first neighbors. The processor with the highest load ratio with its left/right (the direction of comparison alternates every other rebalance step) neighbor's domain is shrinked by moving the boundary with the unloaded neighbor. Dynamic re-balancing is required because the wake geometry changes as the driver evolves, but the initial wake longitudinal radius (Eq. 2.1.1) could be in principle used to start from an initially good load distribution.

Dynamic load re-balancing is performed on the fly, without restarting the pipeline at each modification.

The domain boundary reassignment must be performed enforcing the physical time synchronicity and ensuring the continuity of the integration in ζ. Since neighbor processors compute different physical times, that means performing the boundary shift must occurs at different CPU "clocks" on neighbor processors. For example, if the domain D_i is shrinked on the left side after the epoch t_j, the domain D_{i+1} (on its left) is expanded (on its right) in the next CPU "clock", when D_{i+1} computes t_j (as in figure 6.4.2 (c)): this way the t_j integration in ζ is continued from the point where P_{i+1} left it so the integration in D_{i-1} at t_{j-1} . A similar rule is followed when shrinking a node on its right side, but in this case the first move is to expand the domain on the right (i.e. the first subdomain to be resized is always the one on the right, which is an epoch ahead). In this case, when the unloaded subdomain D_i is expanded on its left side at t_{j+1}, it is necessary that the left neighbor D_{i+1} (which is shrinking) sends it the data for the part of the driver that was not in the D_i subdomain at the past epoch t_j.

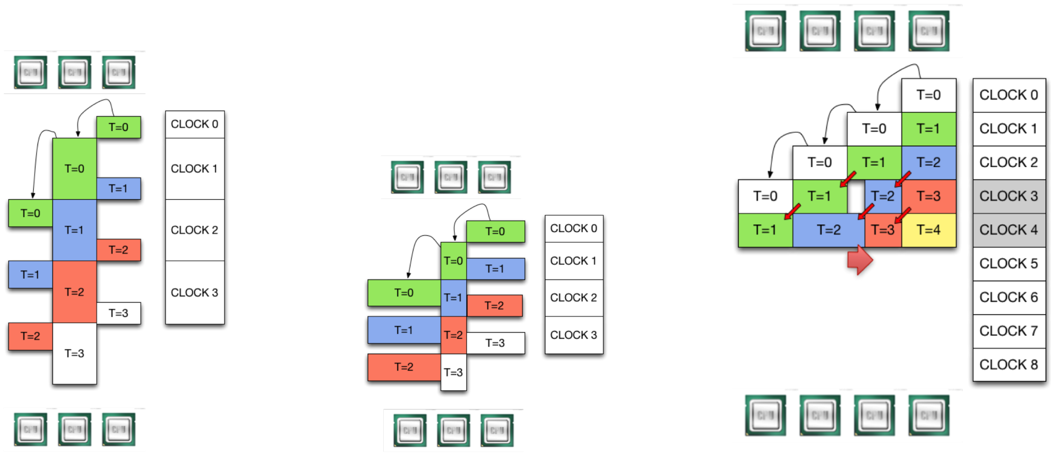


Figure 6.4.2.: (a)(b)(c)
a) Unbalanced load across subdomains (horizontal axis) causes the pipeline to stall, as most of CPU time (vertical axis) is spent in idle status by processors.
b) If the most loaded subdomain is shrinked, the pipeline starts working efficiently.
c) Temporal synchronization at the right ζ coordinate when shrinking the second subdomain from the right.

6.5. INF&RNO/Quasi Static Benchmark: ~10 GeV acceleration stage simulation

The efficiency of INFERNO/Quasi Static was measured by running a virtual, staged (injector+accelerator), LWFA experiment, fig 6.5.1.

A BELLA-class [25] $40J$ laser pulse ($100fs$ (fwhm) long and with a transverse waist of $63\mu m$) is first used to inject and accelerate up to $\sim 300MeV$ an electron bunch, self-injected in the negative density gradient (density downramp) following a $\sim 100\mu m$-long electron density peak (generated by a gas jet). The pulse then is propagated for $50cm$ in a plasma channel (gas cell) of axial density of $2.0 \cdot 10^{17}\ cm^{-3}$, and a radial realistic profile calculated using MHD simulations. In the second stage, the laser, guided for many Rayleigh ranges by the plasma channel, drives a quasi-linear wakefield that manages to accelerate the bunch until its dephasing with the wake. The quasi-static code can reproduce the acceleration behavior and the laser evolution in the second stage, after the injection. Simulations show that dephasing is reached after $45mm$ of propagation, with the injected beam reaching a maximum energy of $\sim 9.5GeV$.

The acceleration stage was modeled using the quasi static code, for various numerical (the B_ϕ-blending scheme) and parallelization configurations. In the following table we summarize the overall speedup (time to wait for a simulation result) and the computational efficiency (total number of core hours) obtained by using the quasi-static code. With the quasi-static code, the same simulation results can be obtained using about one thousand less computing power and waiting tens of times less for the result. Compared to the baseline version (serial and uniform B_ϕ-blending), the parallel, multi-frequency B_ϕ-blending version of the quasi-static code is ~ 20 times faster, using 8 cores.

Method	Processors	Time (h)	Time (CPU h)	Speedup	Efficiency
Explicit INF&RNO	200	˜250	˜50000	1x	1x
Quasi-Static	1	122	122	2x	410x
Quasi-Static	8	18	144	14x	347x
Q-S (Multi-frequency)	1	20	20	12.5x	2500x
Q-S (Multi-frequency)	8	6	48	42x	1040x

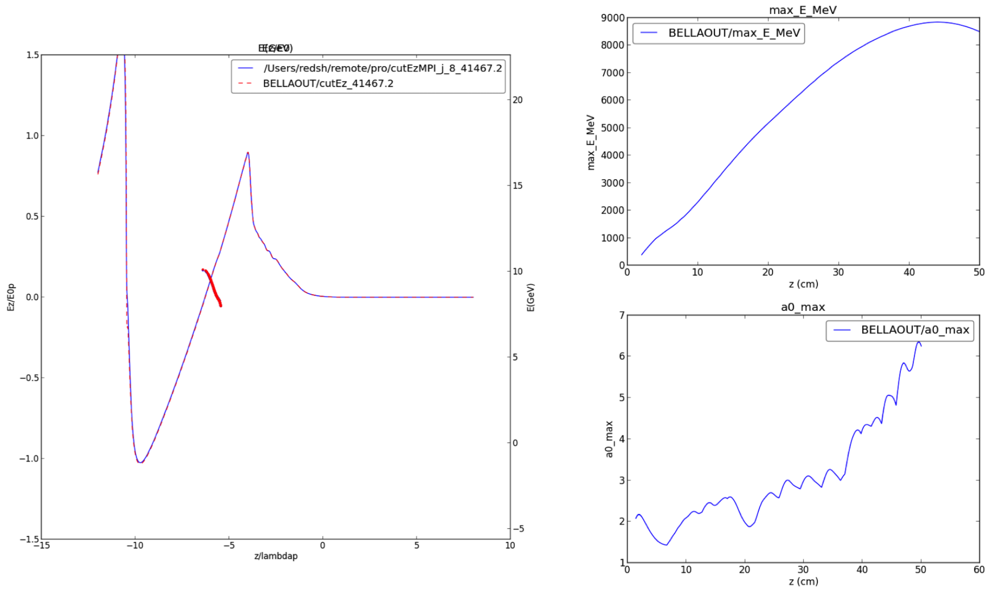


Figure 6.5.1.: Quasi-static stage of a two-stage (explicit PIC+quasi static) simulation, the injected beam is obtained with an explicit PIC simulation. Acceleration continues until dephasing, where the beam has reached an energy of ~9.6GeV (left and top right plots). Good guiding of the pulse over ~50cm is achieved by the matched radial profile of plasma channel (as can be seen from the bottom right plot, showing the normalized vector potential). Laser-plasma parameters are: $40J$ laser pulse, length $100fs$ (fwhm), waist $63\mu m$, plasma density: $2.0 \cdot 10^{17}\ cm^{-3}$. The propagation length is $50cm$.

Part III.

Modeling of Experiments

7. Electron acceleration case studies

7.1. INFN-LNF FLAME

The INFN-LNF FLAME (Frascati Laser for Acceleration and Multidisciplinary Experiments) laser [40] is a nominal $220TW$ laser system that uses 11 YAG pump lasers and 5 titanium-sapphire multi-pass amplifiers to produce linearly polarized pulses with a central wavelength of $\lambda_0 = 0.805\mu m$, pulse duration of $\tau_L < 30fs$ and maximum energy $7J$ at a $10Hz$ repetition rate. The laser is focused via an f/10 off-axis parabolic mirror in a $15\mu m$ diameter (FWHM) spot inside the vacuum chamber.

Laser wakefield electron acceleration experiments were run with FLAME in the framework of the self-injection acceleration programme, aimed at establishing the specifications of self-injected bunches required for the on-going all-optical γ-ray source development [41] based upon the Thomson scattering.

7.1.1. Interaction with flat top density profile

The laser FLAME meets both the conditions of short pulse length and high intensity for self-injecting and accelerating electrons in the laser driven bubble regime. In order to have a “controlled” acceleration process, which ensures a better final bunch quality, the initial plasma and laser parameters were chosen according to the phenomenological theory described in Ref. [56, 55]. For instance, for generating a “matched” (non oscillating) bubble, it is required to focus the laser pulse according to $k_p w_0 \simeq 2\sqrt{a_0}\,(\simeq k_p R_{bub})$. If self-guiding occours, the maximum energy of accelerated elctrons is limited by the dephasing and the pump-depletion length. According to the model presented in Ref. [56, 55], from injection, trapped electrons outrun the accelerating wakefield inside the bubble after a distance $L_{dephasing} \simeq \frac{2}{3}\frac{k_0^2}{k_p^2}R_{bub}$ and the laser pulse is depleted after a distance of $L_{depletion} \simeq \frac{k_0^2}{k_p^2}c\tau_{fwhm}$ of propagation in the plasma.

Analytical/numerical studies were performed in order to define the possible optimal working point of the laser FLAME for the INFN-SITE experiment (Ref. [43]). Given these constraints, a possible working point was choosen to be $w_0 = 13\mu m$, yielding a peak normalized vector potential intensity $a_0 = 5.8$ and an electron density $n_p = 3.8{\cdot}10^{18}cm^{-3}$. For this case, the theretical energy gain would be $\Delta W = 0.9GeV$ and the acceleration is limited at $4mm$ of propagation by both pump depletion and dephasing.

3D simulations, performed with the code jasmine (chapter 4 and Ref. [9]) show that, after $2.8mm$ of propagation, the accelerated beam is already dephased with the accelerating wakefield. Figure 7.1.1 (left) shows the snapshot of a 3D simulation at $ct = 2.8mm$, in which the decelerating effect of dephasing is clearly visible, limiting the final energy gain to $\sim 0.65GeV$. The anticipated dephasing is due to nonlinear effects that alter the laser shape, and hence the accelerating wakefield, at propagation times of the order of the pump depletion time (Ref. [44]).

To prevent this effect, the pump depletion length was increased by lowering the density to $3{\cdot}10^{18}cm^{-3}$, and the waist was increased to $15.5\mu m$ (yielding $a_0 = 4.9$ for FLAME), in order to increase the bubble size and restore the maximum accelerating field ([43]). In this optimized case, acceleration continues past the $2.8mm$ of propagation and simulations show a final $0.9GeV$ beam, with momentum spread ˜5% and a total charge of $\sim 0.6nC$.

7.1.2. Structured Gas Jet

In this chapter we discuss the numerical modeling of the recent results obtained using the FLAME (Frascati Laser for Acceleration and Multidisciplinary Experiments) laser [40] , focused on a “structured” gas-jet target, generated by a $10mm \times 1.2mm$ rectangular nozzle (Helium). Laser propagation was set to be longitudinal, along the 10 mm side, which is characterized by a structured longitudinal profile consisting of a double-peak density profile (see Figure 7.1.2).

Propagation along the double-peaked, $10mm$, density profile was therefore experimentally investigated to explore the possibility of future schemes of staged acceleration and to primarily investigate laser and electron propagation in such a structured plasma profile, with the first part of the profile acting as the accelerating region and the second part acting as a target plasma for the accelerated bunch.

Experiments were run at three different gas-jet backing pressures, namely 5, 8 and 15 bars, corresponding to plasma densities of $1.2 - 3.6 \cdot 10^{19}cm^{-3}$ in the peaks and $3.5 - 10 \cdot 10^{18}cm^{-3}$ in the central plateau, and the results are presented in Ref. [42].

3D Particle-in-Cell simulations have been performed using the *jasmine* (chapter 4 and

Ref. [9]) code to unfold the electron acceleration regime activated in the first peak and to calculate expected bunch properties at the exit of the first peak. The laser was modeled using a gaussian pulse with normalized vector potential $a_0 = 2.45$, waist $w_0 = 15.5 \mu m$ laser and pulse duration $\tau_{fwhm} = 30 fs$. The plasma was modeled using the structured density profile reported in Fig. 7.1.2, for the 5 bar gas-jet backing pressure.

Simulations show the formation of a bubble in the first density peak, which then propagates through the plateau gradually losing its shape as the laser driver intensity decreases. In the initial bubble, a bunch of electrons is self-injected and accelerated up to 242 MeV with a peak in the energy spectrum at 186 MeV with 6.1% relative energy spread and ˜10 mrad divergence (Figure 7.1.3, left).

While the laser depletes and defocuses due to the lower density in the plateau, the bubble loses its shape and the electron bunch partially loses energy and increases its divergence and its relative energy spread (Figure 7.1.3, right). During defocusing, at the simulated density, simulations show the formation of a wake driven by a bunch propagating in the plateau. This is clearly visible in the density plot of Figure 7.1.4 left: the laser pulse is visible at $6525 \mu m$ of propagation and at this stage has expanded to a diameter of $\sim 60 \mu m$ and so is the first electron bunch just behind. A second, moderately collimated bunch is located at 6505 μm and has created a strong wake just behind. A third, smaller bunch injected in this wake is also visible at $6495 \mu m$.

Simulation snapshots at later times (Figure 7.1.4 (right)) show that as the pulse propagates further and enters the second density peak, a partial refocusing of the residual laser light occurs, due to relativistic self- focusing, which drives a new bubble that accelerates additional electrons.

Additional numerical simulations were carried out at higher densities, to better understand its roles in these processes. Indication from these simulations is that at the maximum density explored experimentally, stronger self-focusing occurs in the propagation in the first density peak, with consequent strong defocusing and significant reduction of the acceleration length in the plateau. On one hand, better conditions for a cleaner beam–plasma interaction are set in this case, free from laser-driven wakefield effects. On the other, the higher density in the acceleration peak, makes acceleration peak less optimal, with lower energy and broader spectrum.

7.2. Experiments at ILIL

The 10TW Ti:Sa laser system (delivering up to $400mJ$ on target and featuring a nanosecond contrast better than 10^9) installed at the Intense Laser Irradiation Laboratory of the INO of the CNR in Pisa is currently used for the study of two acceleration regimes [24], one aimed at producing high charge$\sim$ $15MeV$ bunches for radiobiology applications using high density ($> 5 \cdot 10^{19} el./cm^3$) targets and the other aimed at producing monochromatic bunches suitable for developing all-optical Thomson X-ray sources [41]. In both regimes, the target consists of a supersonic nitrogen (N_2) gas-jet, generated from a rectangular nozzle, with the laser propagating across the (shorter) $1.2mm$ side.

7.2.1. High density regime for radiobiology applications

The first regime has been tuned for radiobiology studies where the accelerated electron beam is delivered on a sample for dosimetry and in-vitro sample exposure. The electron energy was tuned to match the spectral and beam divergence features of IORT linacs, with maximum energies up to $15MeV$ and high energy spread. This regime can be accessed with the ILIL laser system focusing the pulse in a spot size of $6.2\mu m$ (FWHM), using a f/4.5 off-axis parabolic mirror, yielding an intensity $I_L = 8 \cdot 10^{18} W/cm^2$ ($a_0 = 1.92$). The optimal density for this regime has been found at $1.6 \cdot 10^{19}$ atoms/cm^3. In these conditions, accelerated electrons are emitted in a $0.5rad$ cone and energies up to 20 MeV. In dosimetric measurements and monte-carlo simulations, cumulated doses up to 10 Gy have been observed, in approximately 1 minute of exposure at 1Hz laser repetition rate (Ref.[24]). The estimated overall average energy was $1.5MeV$, with a total charge of $2.6nC$ per shot.

Due to the high density, both experimental data (Thomson scattering images) and simulations, performed in 3D with the code jasmine, show that the laser plasma interaction is limited by depletion to the first $300\mu m$ of propagation.

In this regime, a 3D simulation was performed for laser intensity $a_0 = 1.92$ and a plasma density profile with peak density $n_p = 1 \cdot 10^{20} 1/cm^3$ and a linear $100\mu m$ long entrance ramp. Laser depletion is reached after $\sim 300\mu m$ of propagation, and a high divergence, high-energy spread (>50%) electron bunch is accelerated with an average energy of $9MeV$ (Figure 7.2.1).

7.2.2. Low density regime for Thomson scattering

The second regime is accessed at lower densities and focusing the pulse to a spot size of $20\mu m$ FWHM, giving an intensity on target of about $2 \cdot 10^{18} W/cm^2$ ($a_0 = 0.96$). Prerequisites for Thomson scattering sources are reproducibility, collimation and small energy spread. Laser-plasma sources require a significant amount of optimization before acceptable conditions are satisfied. An extensive experimental study was carried out for optimizing the acceleration and select the optimal density for acceleration. The injection threshold has found to occur at a ion density of $\sim 1.2 \cdot 10^{18} cm^{-3}$. In experiments, the most stable conditions were found with nitrogen density in the range $1.4 - 1.6 \cdot 10^{18} cm^{-3}$. In this configuration, electron bunches with energies up to $60 MeV$, with a $40 MeV$ monoenergetic component of less than 10% BW were observed [24].

According to numerical simulations, the ionization degree on axis is expected to exceed N^{5+}. The ionization map in Figure 7.2.2 was obtained via a fully consistent 3D PIC simulation run with the code *jasmine*, in which the ionization process was modeled using the ADK model as described in section 4.4.1. For Nitrogen atom density $n_N = 2 \cdot 10^{18} cm^{-3}$, the simulations shows that during early propagation in the plasma, the laser pulse self-focuses until the normalized peak potential reaches $a_0 > 2$, allowing the ionization beyond the N^{5+} state within a radius of $\sim 3\mu m$ (Figure 7.2.2, top right) from the propagation axis and self-injection of electrons after $\sim 500\mu m$ of propagation. In our simulation, electrons reach maximum energies of $\sim 70 MeV$.

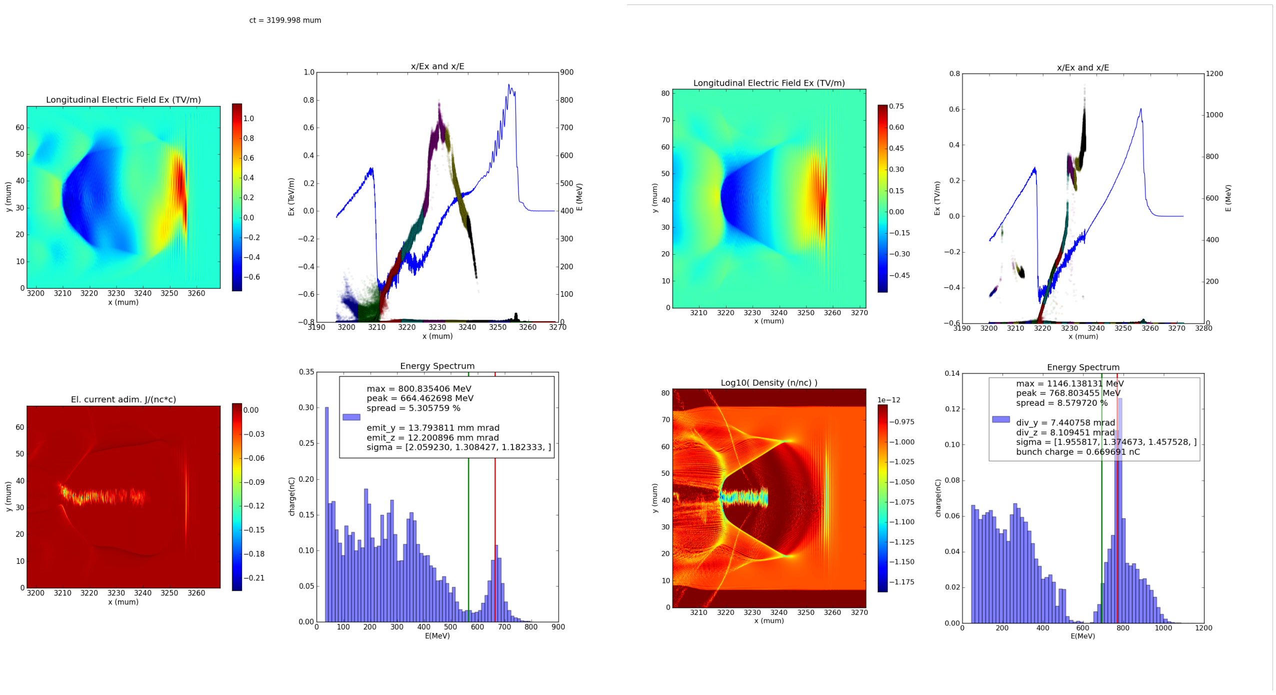


Figure 7.1.1.: FLAME, flat top plasma density profile, 3D PIC simulation snapshots at $ct = 2.8mm$ of propagation, for the case with $w_0 = 13\mu m$ (left) and for the optimized one $w_0 = 15.5\mu m$ (right). With the original parameters ($n_p = 3.8 \cdot 10^{18}$, $w_0 = 13\mu m$), the bend in the phase space plot (top right quadrant in the left simulation) clearly shows (anticipated) dephasing (due to nonlinear laser pulse evolution), while acceleration continues for the optimized case with lower density $n_p = 3 \cdot 10^{18} cm^{-3}$ and waist $w_0 = 15.5\mu m$.

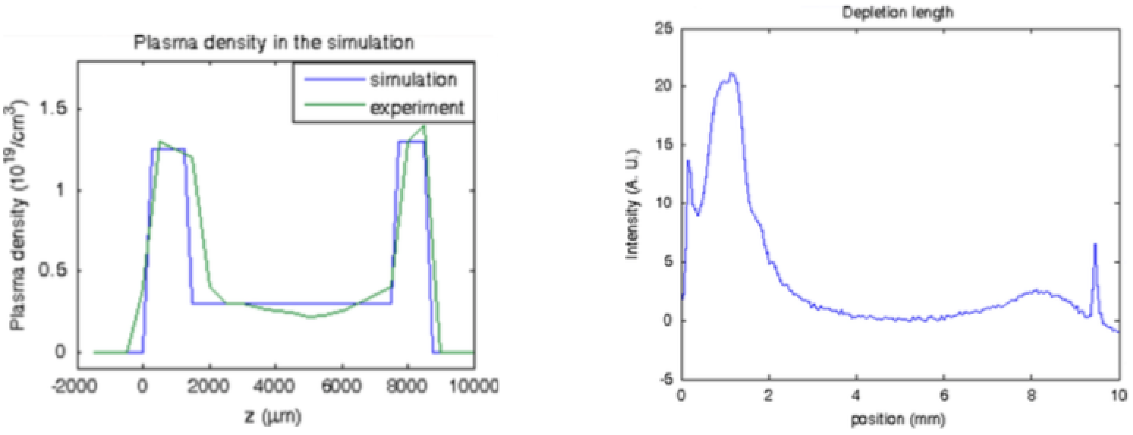


Figure 7.1.2.: (Left) Plasma density structured profile scheme.
(Right) Thomson scattering image, from experimental runs. Intensity profile of the red light integrated on the shorter side of the gas-jet.

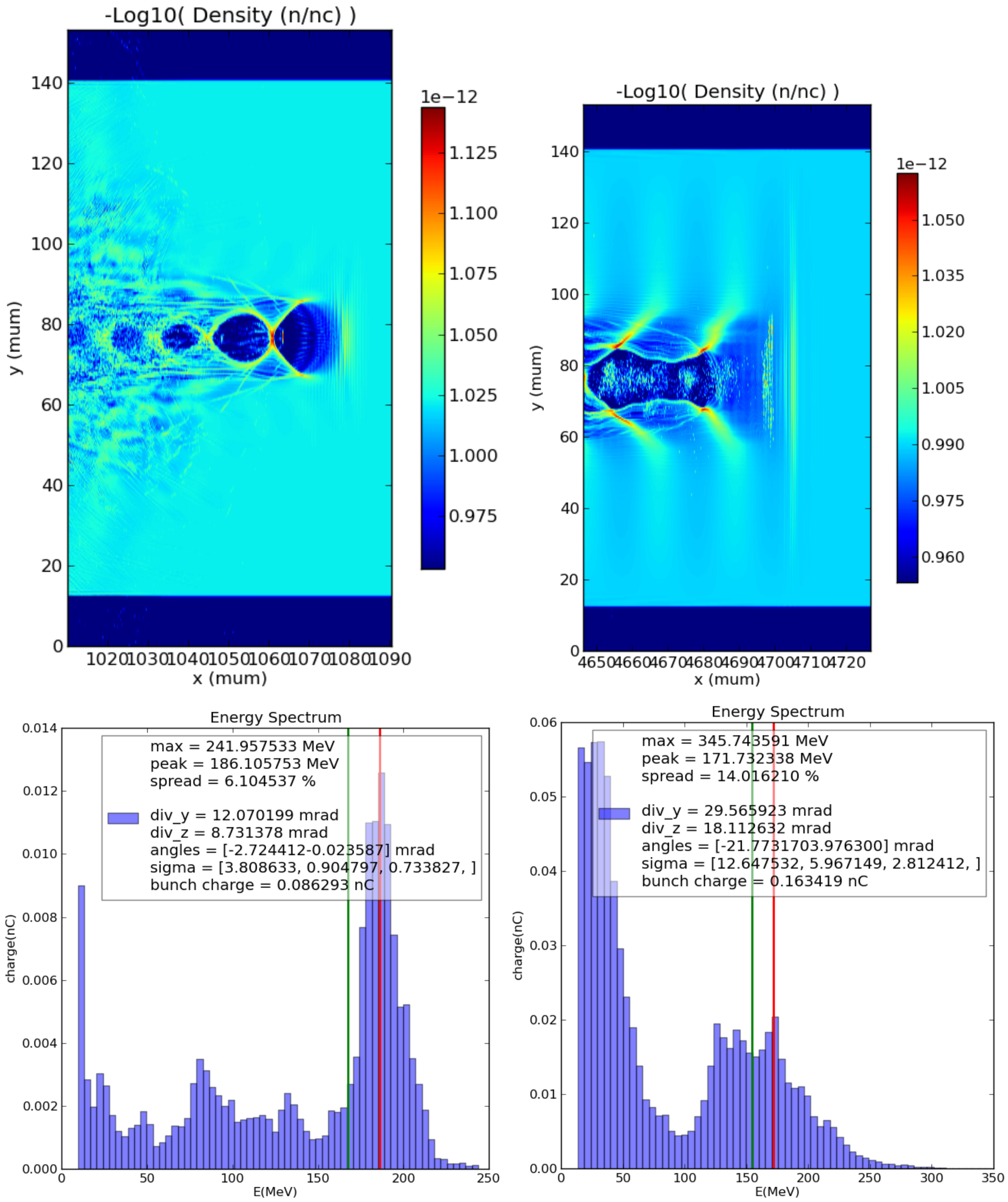


Figure 7.1.3.: Electron density slices and spectra after the first density ramp (left) and after 4.6mm of propagation (right).

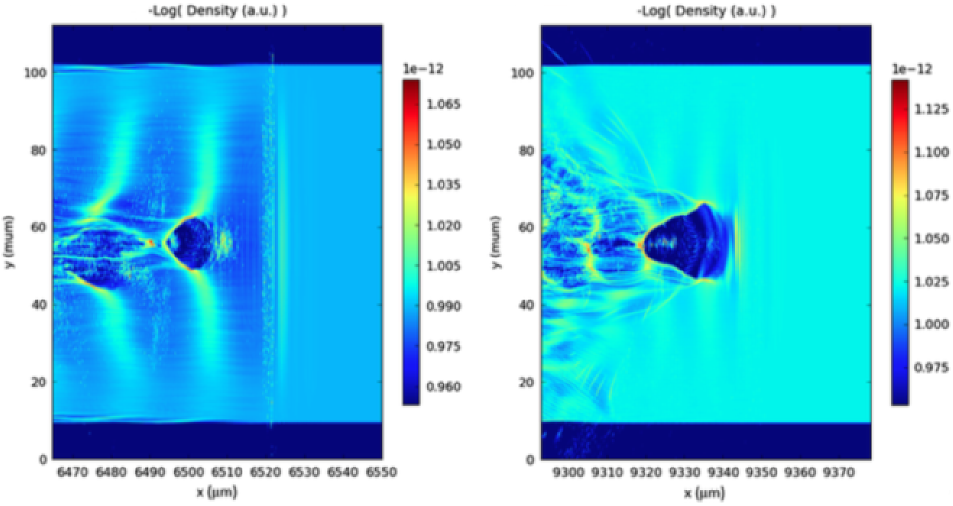


Figure 7.1.4.: Electron density plot obtained from simulations after $6525\mu m$ (left) and $9350\mu m$ (right) of propagation in the plateau and in the second density peak of the density profile. The wake produced by the electron bunch is visible in the left plot at $6505\mu m$.

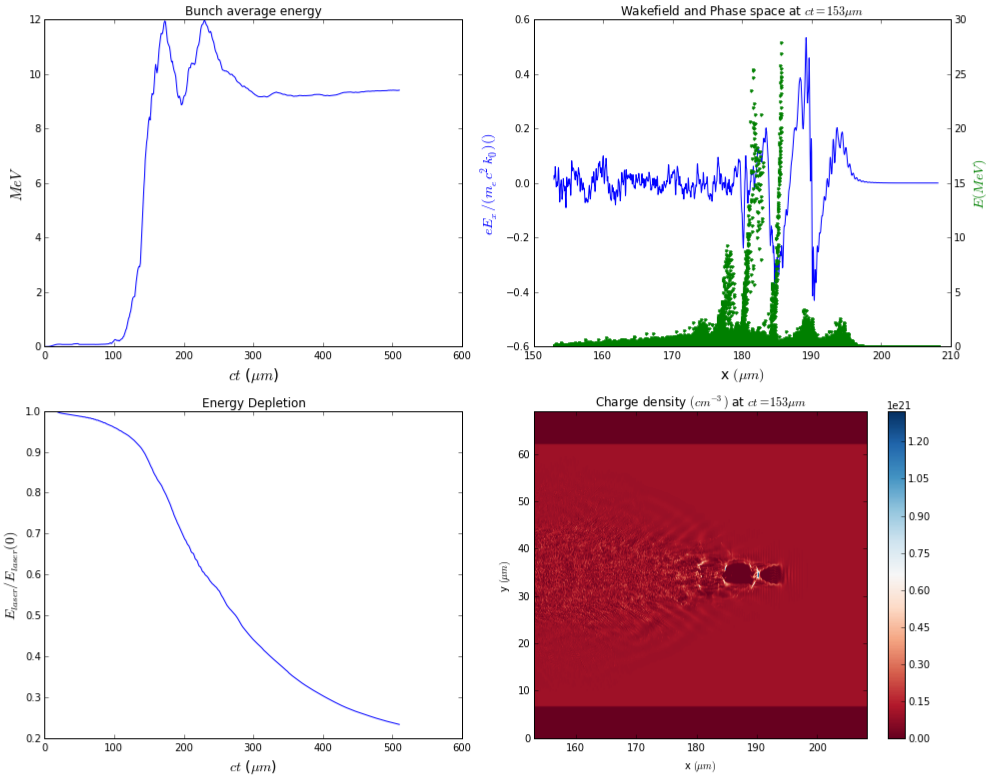


Figure 7.2.1.: Simulation in the ILIL laser high density regime ($a_0 = 1.92$ and $n_p = 1 \cdot 10^{20} cm^{-3}$). Temporal evolution of laser energy (normalized to the initial value, bottom left) and of the average bunch energy (top left). Simulation snapshot after $ct = 153\mu m$ of propagation, showing the phase space and wakefield structure (top right) and a 2D slice of the density grid (bottom right).

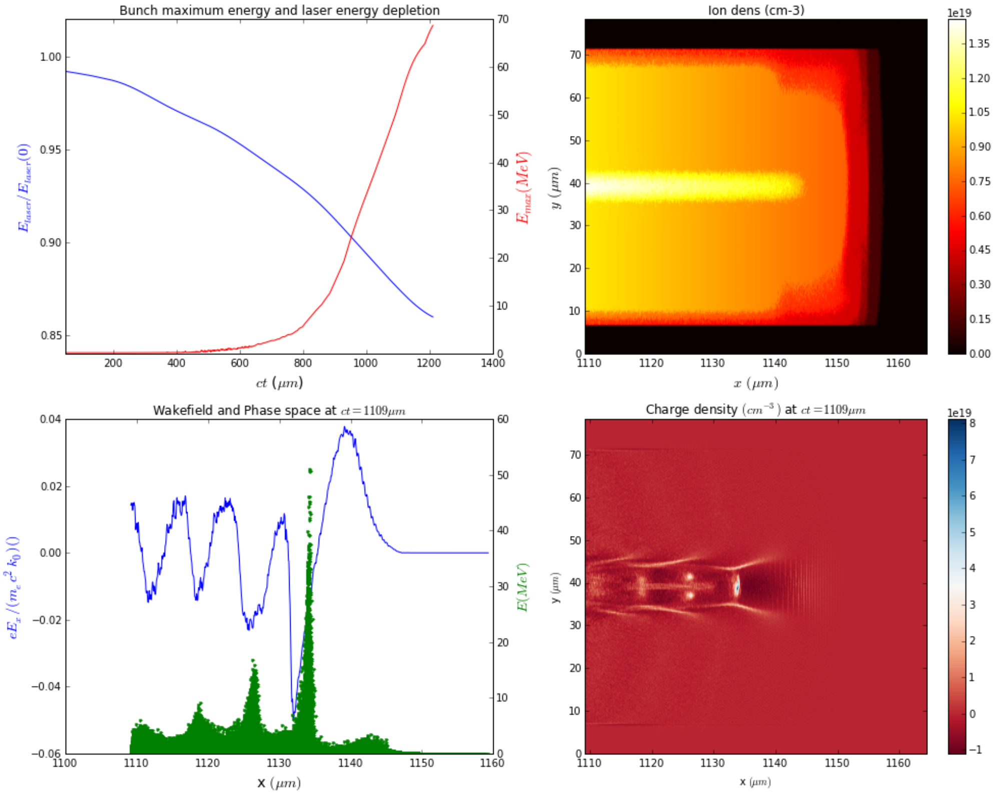


Figure 7.2.2.: 3D *jasmine* simulation in the ILIL laser low density regime. The simulation shows higher ionization status than N^{5+} on-axis ionization, injection and acceleration up to maximum energies of $70MeV$. A laser pulse with $a_0 = 1.0$, $w_0 = 20\mu m$, $\tau = 40fs$ is focused on a Nitrogen target with atom density $n_N = 2 \cdot 10^{18} cm^{-3}$ and $150\mu m$ long ramps. See the discussion in the text for interpretation.
(Top Left) Maximum electron energy and laser energy depletion versus propagation distance.
(Top Right) Ionization map: charge density of ionized nitrogen atoms divided by the absolute value of the elementary charge, showing f after $\sim 1.1mm$ of propagation.
(Bottom Left) Accelerating wakefield and longitudinal phase space after $\sim 1.1mm$ of propagation.
(Bottom Right) Plasma density after $\sim 1.1mm$ of propagation.

Conclusions

Understanding the details of the propagation of short and intense laser pulses in plasmas and the properties of the generated wakefield are a topics of fundamental importance in the field of laser-plasma interaction and, in particular, for the design and optimization of laser-driven plasma-based electron accelerators, which have recently demostrated the capability to accelerate high quality multi-GeV electron bunches over extremely short distances (of the order of the centimeter) and are interesting candidates for applications to future high energy colliders and radiation sources.

In Chapter 1, laser evolution and plasma wave excitation are characterized, including the effects of pulse steepening and energy depletion, for a relativistically intense, short-pulse laser propagating in a preformed parabolic plasma channel. Novel analytical expressions are derived in 3D, and in the weakly-relativistic intensity regime ($a_0 < 1$), for the initial values laser energy depletion, the pulse self-steepening rate, the laser intensity centroid velocity, and the phase velocity of the plasma wave. Analytical results have been succesfully validated numerically using the 2D-cylindrical, ponderomotive code INF&RNO. The dependence of the phase velocity on laser driver evolution, identifying and discussing the role of transverse and longitudinal evolution. We found that, in 3D, the evolution of the phase velocity is mainly determined by the details of the transverse laser evolution, whereas changes in the phase velocity related to longitudinal driver evolution only play a role over propagation distances comparable with depletion length.

Due to the high nonlinearity and complexity of the phenomena involved, numerical simulations are necessary tools for studying of laser plasma-interaction at higher intensities, both for modeling experiments and for creating theoretical models.

In Chapter 2, the wake properties and the process of electron self-injection in the nonlinear bubble wake generated by a short and intense laser pulse propagating in a uniform underdense plasma are systematically studied by means of fully self-consistent Particle-in-Cell simulations. Considering a wake generated by a non-evolving laser pulse traveling with a prescribed velocity the injection dynamics is decoupled from driver evolution, but a realistic structure for the wakefield is retained. It has been demostrated that a threshold for self-injection into a non-evolving bubble wake exists, and the dependence of the self-injection threshold on laser intensity and wake velocity has been characterized for a range of parameters of interest for current and future laser-plasma accelerators.

3D Particle-in-Cell simulations are the most robust tools available for simulating nonlinear laser-plasma interaction physics, but they are extremely demanding in terms of computational power. In Part II, we have reviewed the numerical methods, and the numerical and computational optimizations that allow to accurately model the 3D Physics of laser plasma accelerators with present supercomputing architectures.

In Chapter 4, we have presented the GPU Particle-in-Cell code jasmine and we have shown its single-GPU performance benchmark. The code scalability was measured for a warm plasma test simulation run on multiple GPU nodes. A GPU-friendly load-balancing algorithm has been proposed, helping to efficiency for laser plasma acceleration simulations in which the load imbalance across nodes would severely hurt performance. The relative performance of jasmine, versus our CPU PIC implementation ALaDyn [3], has been measured running full scale, 3D, benchmark simulations in the same numerical conditions. The performance of jasmine per single, Fermi-generation, GPU board has shown to be roughly equivalent to the one of ALaDyn run on 40-50 modern CPU cores. In addition, GPU performance is scaling (and it is planned to scale) well with years, while single core performance is not following the same trend. In jasmine, we have observed a “free” ~25% performance increase moving from the Fermi to the Kepler architectural generation. Furthermore, we think that the algorithms we have developed for GPUs today can fit future high performance computing architectures. The code *jasmine* has been used to model recent LWFA experiments (Sections 7.1 and 7.2) run with the $220TW$ INFN-LNF FLAME laser system and with the $10TW$ laser system installed at the Intense Laser Irradiation Laboratory (ILIL) of the INO of the CNR in Pisa, with the goal of studying optically driven electron beam sources for Thomson scattering [41].

Scaling to longer acceleration lengths is the key to success for multi-GeV LWFA acceleration schemes. Modeling such long stages with full 3D Particle-in-Cell models is computationally prohibitive, with present supercomputing architectures, because of the extremely large scale separation between the shortest distance to resolve, the laser wavelength, and the interaction (acceleration) length.

The LBNL codes INF&RNO and INF&RNO/quasi-static use reduced models to scale down the computational demands by several orders of magnitude, but, for accurately describing the nonlinear 3D Physics of high energy laser-plasma accelerators, several thousands of CPU core hours are still requried. For these codes, efficient parallelization is therefore still of fundamental importance. The advanced numerical schemes required to develop advanced parallelization techniques. The two codes, and the numerical and computational optimizations, that only combined give the possibility to accurately model multi-GeV LWFA stages with present architectures in practical times, were described in Chapter 5 and 6.

Bibliography

[1] E. Esarey, C. B. Schroeder, and W. P. Leemans, Rev. Mod. Phys. **81**, 1229 (2009).

[2] W. P. Leemans, B. Nagler, A. J. Gonsalves, C. Tóth, K. Nakamura, C. G. R. Geddes, E. Esarey, C. B. Schroeder, and S. M. Hooker, Nature Phys. **2**, 696 (2006).

[3] C. Benedetti, A. Sgattoni, G. Turchetti, P. Londrillo. "ALaDyn: A High-Accuracy PIC Code for the Maxwell-VlasovEquations" IEEE Transactions on Plasma Science, 36(4), 2008.

[4] Londrillo, P., Benedetti, C., Sgattoni, A., and Turchetti, G., "Charge preserving high order PIC schemes," Nucl. Inst. Meth. Phys. Res. A 620(1), 28–35 (2010). 23 .

[5] W.P. Leemans, A.J. Gonsalves, H.-S. Mao, K. Nakamura, C. Benedetti, C.B. Schroeder, Cs. Tóth, J. Daniels, D.E. Mittelberger, S.S. Bulanov, J.-L. Vay, C.G.R. Geddes, and E. Esarey, "Multi-GeV Electron Beams from Capillary-Discharge-Guided Subpetawatt Laser Pulses in the Self-Trapping Regime," Physical Review Letters 113, 245002 (8 December 2014).

[6] W. P. Leemans, B. Nagler, A. J. Gonsalves, C. Tóth, K. Nakamura, C. G. R. Geddes, E. Esarey, C. B. Schroeder, and S. M. Hooker, Nature Phys. **2**, 696 (2006).

[7] C. B. Schroeder, C. Benedetti, E. Esarey, and W. P. Leemans, Phys. Rev. Lett. **106**, 135002 (2011).

[8] Benedetti, C.; Schroeder, C. B.; Esarey, E.; Rossi, F.; Leemans, W. P., Physics of Plasmas, Volume 20, Issue 10, 103108 11 pp. (2013).

[9] Rossi, Francesco and Londrillo, Pasquale and Sgattoni, Andrea and Sinigardi, Stefano and Turchetti, Giorgio, AIP Conference Proceedings, 1507, 184-192 (2012), DOI:http://dx.doi.org/10.1063/1.4773692

[10] X. Wang, R. Zgadzaj, N. Fazel, Z. Li, S. A. Yi, X. Zhang, W. Henderson, Y.-Y. Chang, R. Korzekwa, H.-E. Tsai, C.-H. Pai, H. Quevedo, G. Dyer, E. Gaul, M. Martinez, A. C. Bernstein, T. Borger, M. Spinks, M. Donovan, V. Khudik, G. Shvets, T. Ditmire, and M. C. Downer, Nat. Commun. 4, 1988 (2013).

[11] C. B. Schroeder, E. Esarey, C. Geddes, C. Benedetti, W. P. Leemans, "Physics considerations for laser-plasma linear colliders," Phys. Rev. ST Accel. Beams 13, 101301 (2010).

[12] W. P. Leemans, C. G. R. Geddes, J. Faure, Cs. Toth, J. van Tilborg, C. B. Schroeder, E. Esarey, G. Fubiani, D. Auerbach, B. Marcelis, M. A. Carnahan, R. A. Kaindl, J. Byrd, and M. C. Martin, Phys. Rev. Lett. 91, 074802 (2003).

[13] A. Rousse, K. T. Phuoc, R. Shah, A. Pukhov, E. Lefebvre, V. Malka, S. Kiselev, F. Burgy, J.-P. Rousseau, D. Umstadter, and D. Hulin, Phys. Rev. Lett. 93, 135005 (2004).

[14] M. Fuchs, R. Weingartner, A. Popp, Z. Major, S. Becker, J. Osterhoff, I. Cortrie, B. Zeitler, R. H€ orlei, G. D. Tsakiris, U. Schramm, T. P. Rowlands-Rees, S. M. Hooker, D. Habs, F. Krausz, S. Karsch, and F. Gr€ uner, Nat. Phys. 5, 826 (2009).

[15] E. Esarey, R. F. Hubbard, W. P. Leemans, A. Ting, and P. Sprangle, Phys. Rev. Lett. 79, 2682 (1997).

[16] J. Faure, C. Rechatin, A. Norlin, A. Lifschitz, Y. Glinec, and V. Malka, Nature 444, 737 (2006).

[17] S. Bulanov, N. Naumova, F. Pegoraro, and J. Sakai, Phys. Rev. E 58, R5257 (1998).

[18] C. G. R. Geddes, K. Nakamura, G. R. Plateau, Cs. Toth, E. Cormier-Michel, E. Esarey, C. B. Schroeder, J. R. Cary, and W. P. Leemans, Phys. Rev. Lett. 100, 215004 (2008).

[19] A. J. Gonsalves, K. Nakamura, C. Lin, D. Panasenko, S. Shiraishi, T. Sokollik, C. Benedetti, C. B. Schroeder, C. G. R. Geddes, J. van Tilborg, J. Osterhoff, E. Esarey, Cs. Toth, and W. P. Leemans, Nat. Phys. 7, 862 (2011).

[20] A. Pak, K. A. Marsh, S. F. Martins, W. Lu, W. B. Mori, and C. Joshi, Phys. Rev. Lett. 104, 025003 (2010).

[21] C. McGuffey, A. G. R. Thomas, W. Schumaker, T. Matsuoka, V. Chvykov, F. J. Dollar, G. Kalintchenko, V. Yanovsky, A. Maksimchuk, K. Krushelnick, V. Yu. Bychenkov, I. V. Glazyrin, and A. V. Karpeev, Phys. Rev. Lett. 104, 025004 (2010)

[22] C. Benedetti, F. Rossi, C. B. Schroeder, E. Esarey, and W. P. Leemans, "Laser Pulse Propagation and Phase Velocity of Laser-driven Plasma Waves in the Weakly-Relativistic Intensity Regime"*, Submitted to Physical Review E*

[23] T. Tajima and J. M. Dawson, Phys. Rev. Lett. 43, 267

[24] Laser-plasma acceleration of electrons for radiobiology and radiation sources L.A. Gizzi, L. Labate, F. Baffigi, F. Brandi, G.C. Bussolino, L. Fulgentini, P. Koester, D. Palla, F. Rossi, *Submitted to Nuclear Instruments and Methods B*

[25] W. Leemans, J. Daniels, A. Deshmukh, A.J. Gonsalves, A. Magana, H.S. Mao, D.E. Mittelberger, K. Nakamura, J.R. Riley, D. L. Syversrud, C. Tóth, N. Ybarrolaza, "BELLA Laser and Operations", Proceedings of PAC2013, Pasadena.

[26] C. Benedetti, C. B. Schroeder, E. Esarey, C. G. R. Geddes, and W. P. Leemans, in Proceedings of 2010 AAC Workshop, edited by G. Nusinovich and S. Gold (AIP, NY, 2010), Vol. 1299, p. 250-255.

[27] C. Benedetti, C. B. Schroeder, E. Esarey, and W. P. Leemans, in Proceedings of PAC11 (JACoW, 2011) p. MOP082.

[28] C. Benedetti, C. B. Schroeder, E. Esarey, and W. P. Leemans, in Proceedings of ICAP12 (JACoW, 2012) p. THAII02.

[29] S. Gordienko and A. Pukhov, Phys. Plasmas 12, 043109 (2005).

[30] J.-L. Vay, Phys. Rev. Lett. 98, 130405 (2007). 9.

[31] S.F. Martins, et al., in: Proc. 13th Advanced AcceleratorWorkshop, Santa Cruz, CA, p. 285 (2008). 10.

[32] S.F. Martins, et al., Nature Physics 6, p. 311 (2010). 12.

[33] D.L. Bruhwiler, et al., in: Proc. 13th Advanced Accelerator Workshop, Santa Cruz, CA, p. 29 (2008). 11.

[34] C.B. Schroeder, et al., Phys. Rev. Lett. 106 (2011) 135002.

[35] W. Zhu, et al., Phys. Plasmas 19, (2012) 033105.

[36] B.M. Cowan, et al., J. Comp. Phys. 230 (2011) 61.

[37] J.S. Shang, J. Comp. Phys. 153 (1999) 312.

[38] P. Mora, T.M. Antonsen, Phys. Plasmas 4, (1997) 217

[39] C.K. Birdsall, A.B. Langdon. Plasma physics via computer simulation, Taylor & Francis

[40] L. Gizzi, A. Bacci, S. Betti, C.A. Cecchetti, M. Ferrario, A. Gamucci, A. Giulietti, D. Giulietti, P. Koester, L. Labate, T. Levato, V. Petrillo, L. Serafini, P. Tomassini, C. Vaccarezza, Eur. Phys. J. Spec. Top. 175 (2009) 3.

[41] L.A. Gizzi, M.P. Anania, G. Gatti, D. Giulietti, G. Grittani, M. Kando, M. Krus, L. Labate, T. Levato, Y. Oishi, F. Rossi, Nucl. Instr. Methods Phys. Res. Sect. B: Beam Interact. Mater. Atoms 309 (2013) 202.

[42] G. Grittani, M.P. Anania, G. Gatti, D. Giulietti, M. Kando, M. Krus, L. Labate, T. Levato, P. Londrillo, F. Rossi, L.A. Gizzi, High energy electrons from interaction with a structured gas-jet at FLAME, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 740, 11 March 2014, Pages 257-265, ISSN 0168-9002, http://dx.doi.org/10.1016/j.nima.2013.10.082.

[43] C. Benedetti, Simulation of particle acceleration in the PLASMONX project, The 2nd International Conference on Ultra-Intense Laser Interaction Science. AIP Conference Proceedings, Volume 1209. AIP Conference Proceedings, Volume 1209, Issue 1, p.11-14

[44] B. A. Shadwick et al., Phys. Plas. 16, 056704 (2009)

[45] P. Sprangle, E. Esarey, and A. Ting, “Nonlinear theory of intense laser-plasma interactions,” Phys. Rev. Lett., vol. 64, pp. 2011-2014, 1990; “Nonlinear interaction of intense laser pulses in plasmas,” Phys. Rev. A

[46] A. Ting, E. Esarey, and P. Sprangle, “Nonlinear wakefield generation and relativistic focusing of intense laser pulses in plasmas,” Phys. Fluids B, vol. 2, pp. 1390-1594, 1990.

[47] P. Mora, T.M. Antonsen, Kinetic modeling of intense, short laser pulses propagating in tenuous plasmas, Phys. Plasmas 4 (1997) 217.

[48] Mora P and Antonsen T M 1996 Phys. Rev. E 53 R2068

[49] C. Huang et al, QUICKPIC: a highly efficient particle-in-cell code for modeling wakefield acceleration in plasmas, J. Comput. Phys. 217 (2006) 658.

[50] W. An et al. / Journal of Computational Physics 250 (2013) 165–177

[51] T. Mehrling, C. Benedetti, C. B. Schroeder, and J. Osterhoff, "HiPACE: a quasi-static particle-in-cell code," Plasma Physics and Controlled Fusion 56, 084012 (22 July 2014).

[52] W. Lu et al, Nonlinear theory for relativistic plasma wakefields in the blowout regime, Phys. Rev. Lett. 96 (2006) 165002.

[53] C. Benedetti, C.B. Schroeder, E. Esarey, and W.P. Leemans, "Plasma wakefields driven by an incoherent combination of laser pulses: A path towards high-average power laser-plasma accelerators," Physics of Plasmas 21, 056706 (27 May 2014).

[54] Min Chen , Estelle Cormier-Michel, Cameron Geddes, David Bruhwiler, Lule Yu, Eric Esarey, Carl Schroeder, Wim Leemans, "Numerical modeling of laser tunneling ionization in explicit particle-in-cell codes," J. Comp. Phys. 236, 220–228 (2013).

[55] W. Lu, C. Huang, M. Zhou, W. B. Mori, and T. Katsouleas, Phys. Rev. Lett. **96**, 165002 (2006).

[56] W. Lu, M. Tzoufras, C. Joshi, F. S. Tsung, W. B. Mori, J. Vieira, R. A. Fonseca, and L. O. Silva, Phys. Rev. ST Accel. Beams **10**, 061301 (2007).

[57] C. Benedetti, C. B. Schroeder, E. Esarey, and W. P. Leemans, Phys. Plasmas **19**, 053101 (2012).

[58] C. B Schroeder, C. Benedetti, E. Esarey, J. van Tilborg and W. P. Leemans, Phys. Plasmas **18**, 083103 (2011).

[59] B. A. Shadwick, C. B. Schroeder, E. Esarey, Phys. Plasmas **16**, 056704 (2009).

[60] C. B. Schroeder, E. Esaerey, B. A. Shadwick, and W. P. Leemans, Phys. Plasmas **13**, 033103 (2006).

[61] C. Benedetti, C. B. Schroeder, E. Esarey, F. Rossi, W. P. Leemans, Physics of Plasmas **20**, 103108 (2013)

[62] C. D. Decker and W. B. Mori, Phys. Rev. Lett. **72**, 490 (1994).

[63] E. Esarey and M. Pilloff, Phys. Plasmas **2**, 1432 (1995).

[64] C. B. Schroeder, E. Esarey, and B. A. Shadwick, Phys. Rev. E **72**, 055401(R) (2005).

[65] Burau, Widera, Hönig, Juckeland, Debus, Kluge, Schramm, Cowan, Sauerbrey, Bussmann. "PIConGPU: A Fully Relativistic Particle-in-Cell Code for a GPU Cluster", IEEE Transactions on Plasma Science 38(10)

[66] Kong, Huang, Ren, Decyk. "Particle-in-cell simulations with charge-conserving current deposition on graphic processing units". J. Comput. Phys. 230,4 (February 2011)

[67] K. Amyx, "GPU-Accelerated 3-D Electromagnetic Particle-in-Cell Implementations in VORPAL", GPU Technology Conference 2012 Poster

[68] Abreu, Fonseca Pereira, Silva. "PIC Codes in New Processors: A Full Relativistic PIC Code in CUDA-Enabled HardwareWith Direct Visualization". IEEE Transactions on Plasma Science, 39, issue 2

[69] G. Stantchev, W. Dorland, and N. Gumerov. "Fast parallel Particle-To-Grid interpolation for plasma PIC simulations on the GPU". Journal of Parallel and Distributed Computing, 68(10): 1339–1349, 2008.

[70] K.J. Bowers, B.J. Albright, L. Yin, B. Bergen, T.J.T. Kwan, "Ultrahigh performance three-dimensional electromagnetic relativistic kinetic plasma simulation", Physics of Plasmas 15 (2008) 055703.

[71] NVIDIA Corporation, "NVIDIAs Next Generation CUDA Compute Architecture: Fermi", Santa Clara, CA, 2009.

[72] T. Esirkepov, “Exact charge conservation scheme for particle-in-cell simulation with an arbitrary form-factor”. Computer Physics Communications, 135(2):144–153, 2001.

[73] J. Berenger, “A perfectly matched layer for the absorption of electromagnetic waves”. Journal of Computational Physics 114(2): 185–200.

[74] V.K. Decyk, T.V. Singh, S.A. Friedman, “Graphical Processing Unit-Based Particle-in-Cell Simulations”, in: Proceedings of the 10th International Computational Accelerator Physics Conference (ICAP2009), San Francisco, CA, 2009.

[75] Sengupta, Harris, Zhang, Owens. “Scan primitives for GPU computing”. Graphics Hardware 2007, Aug.2007.

[76] Hoberock, Bell. “Thrust: A Parallel Template Library” https://github.com/thrust/thrust

[77] Bayer, “Mako Templates for Python”, http://www.makotemplates.org

[78] Higham, Nicholas J. (1993), "The accuracy of floating point summation", SIAM Journal on Scientific Computing 14 (4):783–799

[79] R.A. Fonseca, L.O. Silva, F.S. Tsung, V.K. Decyk, W. Lu, C. Ren, W.B. Mori, S. Deng, S. Lee, T. Katsouleas, J.C. Adam, “OSIRIS: a three-dimensional, fully relativistic particle in cell code for modeling plasma based accelerators”, Lecture Notes in Computer Science 2331 (2002) 342–351.

[80] R. A. Fonseca, S. F. Martins, L. O. Silva, J. W. Tonge, F. S. Tsung, and W. B. Mori, “One-to-one direct modeling of experiments and astrophysical scenarios: pushing the envelope on kinetic plasma simulations,” Plasma Phys Contr Fusion, 2008, vol. 50, no. 12, p. 124034.

[81] Min Chen, Estelle Cormier-Michel, Cameron Geddes, David Bruhwiler, Lule Yu, Eric Esarey, Carl Schroeder, Wim Leemans, "Numerical modeling of laser tunneling ionization in explicit particle-in-cell codes," J. Comp. Phys. 236, 220–228 (2013).

[82] F.A. Ilkov, J.E. Decker, S.L. Chin, Ionization of atoms in the tunnelling regime with experimental evidence using hg atoms, J. Phys. B At. Mol. Opt. Phys. 25 (19) (1992) 4005–4020.

[83]] L. Landau, E. Lifshitz, Quantum Mechanics, Butterworth-Heinemann, New York, 1965.

[84] M.V. Ammosov, N.B. Delone, V.P. Krainov, Sov. Phys. JETP 64 (1986) 1191

[85]] M. Kre, T. Löffler, M.D. Thomson, R. Dörner, H. Gimpel, K. Zrost, T. Ergler, R. Moshammer, U. Morgner, J. Ullrich, H.G. Roskos, Determination of the carrier-envelope phase of few-cycle laser pulses with terahertz-emission spectroscopy, Nat. Phys. 2 (2006) 327–331.

[86] K.-Y. Kim, J.H. Glownia, A.J. Taylor, G. Rodriguez, Terahertz emission from ultrafast ionizing air in symmetry-broken laser fields, Opt. Express 15 (8) (2007) 4577–4584.

[87] M. Chen, A. Pukhov, X.-Y. Peng, O. Willi, Theoretical analysis and simulations of strong terahertz radiation from the interaction of ultrashort laser pulses with gases, Phys. Rev. E 78 (2008) 046406.

[88] M. Chen, Z.-M. Sheng, Y.-Y. Ma, J. Zhang, Electron injection and trapping in a laser wakefield by field ionization to high-charge states of gases, J. Appl. Phys. 99 (5) (2006) 056109.

[89] A. Pak, K.A. Marsh, S.F. Martins, W. Lu, W.B. Mori, C. Joshi, Injection and trapping of tunnel-ionized electrons into laser-produced wakes, Phys. Rev. Lett. 104 (2) (2010) 025003

[90] C. McGuffey, A.G.R. Thomas, W. Schumaker, T. Matsuoka, V. Chvykov, F.J. Dollar, G. Kalintchenko, V. Yanovsky, A. Maksimchuk, K. Krushelnick, V.Y. Bychenkov, I.V. Glazyrin, A.V. Karpeev, Ionization induced trapping in a laser wakefield accelerator, Phys. Rev. Lett. 104 (2) (2010) 025004

[91] M. Chen, E. Esarey, C.B. Schroeder, C.G.R. Geddes, W.P. Leemans, Theory of ionization-induced trapping in laser–plasma accelerators, Phys. Plasmas 19 (3) (2012) 033101

[92] A. G. R. Thomas, Algorithmfor calculating spectral intensity due to charged particles in arbitrary motion, Phys. Rev. ST Accel. Beams 13, 020702 (2010)

[93] Krall, J., et al, Phys. Plasmas 1, 1738 (1994).

[94] I. Kostyukov, E. Nerush, A. Pukhov, and V. Seredov, Phys. Rev. Lett. 103, 175003 (2009)

[95] A. G. R. Thomas, Phys. Plasmas 17, 056708 (2010)

[96] S. Kalmykov, S. A. Yi, V. Khudik, and G. Shvets, Phys. Rev. Lett. 103, 135004 (2009)

[97] S. Corde, A. Stordeur, and V. Malka, Phys. Plasmas 18, 034701 (2011)

[98] A. G. R. Thomas, Phys. Plasmas 18, 034702 (2011)

[99] S. A. Yi, V. Khudik, C. Siemon, and G. Shvets, Phys. Plasmas 20, 013108 (2012)

[100] Kane S. Yee, IEEE Trans. Antennas and Propagation (1966) 302–307

[101] E. Esarey, R.F. Hubbard, W.P. Leemans, A. Ting, and P. Sprangle, Phys. Rev. Lett. 79, 2682-2685 (1997).