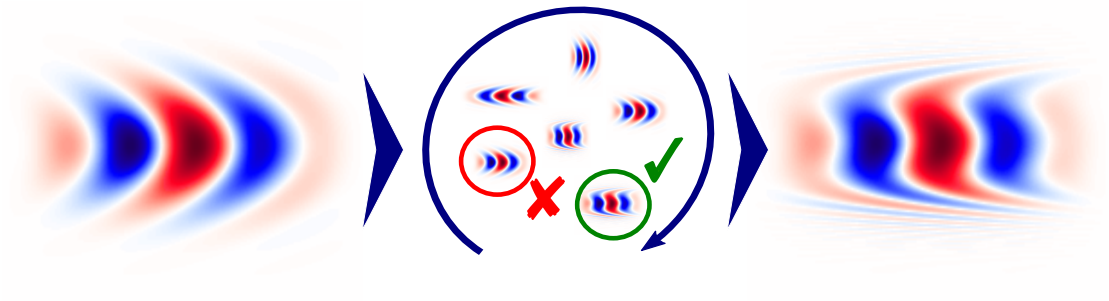
TÉCNICO LISBOA



Machine learning controlled plasma based acceleration

Bernardo Farinha Malaca

Thesis to obtain the Master of Science Degree in

Engineering Physics

Supervisors: Prof. Jorge Miguel Ramos Domingues Ferreira Vieira
Prof. Ricardo Parreira de Azambuja Fonseca

Examination Committee

Chairperson: Prof. João Pedro Saraiva Bizarro
Supervisor: Prof. Jorge Miguel Ramos Domingues Ferreira Vieira
Members of the Committee: Doctor Marija Vranic
Doctor Nelson Manuel Carreira Lopes

November 2018

“Queria proteger contra o esquecimento. A maior vulnerabilidade do humano, a contingência de não lembrar e de não ser lembrado.”

— Valter Hugo Mãe, A Desumanização

Acknowledgments

Firstly, I would like to thank both my supervisors for their availability, support and encouragement throughout this year. Of course, this project would be much harder without the rest of the colleagues in GoLP, whose criticism and curiosity for a new topic was always welcome. A special thanks goes to Rui Calado and Rui Torres, my office cohabitants, who shared not only the space but also some ideas that were deemed very important for this work.

Since the path is more important than the end destination, a deep thanks has to go to all the people who participated in this 5 year journey, especially to my friends that shared the P10 room for so many consecutive nights during exam season.

I am particularly grateful to my family, for always being there when I needed it the most.

Finally, a sincere and heartfelt thanks to my girlfriend, whose support was immense throughout this work.

Resumo

Recentemente utilizaram-se algoritmos genéticos para controlar aceleradores a plasma usando impulsos lasers ultracurtos para aceleração de eletrões. Neste trabalho esta técnica é usada para fazer experiências virtuais de controlo e otimização desses mesmos aceleradores. O algoritmo genético foi implementado no ZPIC, um código *particle-in-cell* (PIC) totalmente relativista. O algoritmo é bastante automatizado: depois de informado do conjunto de parâmetros iniciais, lança várias simulações em paralelo, analisa os resultados e termina automaticamente quando os critérios de convergência são alcançados. Este código pode portanto ser corrido em supercomputadores. Depois de um *benchmark* a PIC 1D, resultados de simulações 1D e 2D são apresentados. Em 1D, a otimização focou-se em plasmas com densidade não uniforme e em lasers com perfis longitudinais variáveis. Em 2D, o papel da frente de onda na aceleração é considerada: a distância de aceleração, a radiação emitida pelos electrões e a eficiência energética (mais partículas num intervalo de energia) foram todas otimizadas via um laser que não era perfeitamente gaussiano. Os resultados da eficiência energética corroboram as teses de outros estudos, mas os resultados da distância de aceleração e radiação são novos e podem levar a experiências para os testar. O algoritmo é geral, e pode ser imediatamente aplicado a qualquer outra classe de problemas de otimização em física de plasma.

Palavras-chave: Aprendizagem Automática, Algoritmo Genético, Plasma, Otimização, Laser

Abstract

Recent experiments have employed genetic algorithms to control plasma-based accelerators, in particular using ultra-short laser pulses to drive the acceleration of electrons. In this work this technique is used to perform virtual experiments of control and optimization of such accelerators. The genetic algorithm was implemented in ZPIC, a fully relativistic particle-in-cell code. The algorithm is almost fully automated: after being informed of the first set of inputs to test, it creates an initial set of input parameters, launches several simulations in parallel, analyses the results, and ends automatically once given convergence criteria are reached. The algorithm can thus take full advantage of large-scale supercomputers. After a benchmark test in 1D, results from both 1D and 2D simulations are presented. In 1D, the optimization focus on plasmas with non-uniform density and lasers with variable longitudinal envelope profiles. In 2D, the role of distorted wave-fronts in the acceleration is considered: the acceleration distance, the radiation produced by the electrons and the energy efficiency (most particles in an energy range) were all optimized via a laser which was not perfectly gaussian. The energy efficiency result corroborate findings from other studies, but the acceleration distance and radiation results are new and can pave way to experiments to test these discoveries. The algorithm is general, and can be readily applied to any other class of optimization problems in plasma physics.

Keywords: Machine Learning, Genetic Algorithm, Plasma, Optimization, Laser

Contents

Acknowledgments v
Resumo vii
Abstract ix
List of Tables xiii
List of Figures xv
Nomenclature xix
Glossary xix

1 Introduction **1**
1.1 Plasma Accelerators 1
1.2 PIC simulations 2
1.3 Numerical simulations and experiments 4
1.4 Machine learning 6

2 Genetic algorithm controlled PIC simulations **9**
2.1 Initial input to the genetic algorithm 9
2.2 Genetic algorithm 11
2.3 Launching machine-learning controlled PIC simulations 12
2.4 Automatic PIC simulation analysis: Clustering and k-means 13

3 Application to one-dimensional scenarios **19**
3.1 Using machine learning to recover known optimizations 19
3.2 Enhancing beam quality 20
3.2.1 Optimization parameters 21
3.3 Results from machine learning optimization with 1D PIC code 23
3.3.1 Simulation parameters 24

4 Two-dimensional simulations: the role of the internal laser structure **27**
4.1 High repetition lasers and the tailoring of the wavefront 27
4.2 Wave front shaping: the Zernike polynomials 27
4.3 Enhancing the acceleration efficiency 29
4.3.1 Energy efficiency 29

4.3.2 Decreasing the acceleration distance 30
4.4 Enhancing the radiation emission 32

5 Conclusions **37**
5.1 Contributions 37
5.2 Future Work 38

Bibliography **39**

A Classes Developed **43**
A.1 1D classes 43
A.1.1 Laser with fixed energy 43
A.2 2D 44
A.2.1 Laser with changed wave front 44

List of Tables

1.1 Normalized units in ZPIC 6

4.1 The first five Zernike polynomials 28

List of Figures

1.1 Numerical simulation of an intense laser pulse (rainbow color) propagating in plasma (grey, white corresponds to positive charges and black to negative ones), superimposed with a graph of the local longitudinal electric field in the axis, in red. 3

1.2 Scheme of the loop that advances both particles and fields. The index i refers to the particles and the index j to the grid points. 4

1.3 Image depicting an example of a genetic algorithm. The fittest elements can produce other offspring, something that is denied to the remaining individuals. This will produce a next generation that will be fitter. 7

2.1 Flow chart of the general procedure 10

2.2 Diagram of the algorithm: From the initializer the several input-decks are distributed for the available computer nodes. The simulation results are sent to the root core, which creates a new text file with new parameters which are again fed to the available nodes, until convergence is reached. 12

2.3 Application of the kmeans method. The starting point is a set of points identified with some variables (x,y and z) with no labels (left). The number of clusters (3) is predetermined. At the convergence point each object is assigned label, identified as a color in the plot (right). 15

2.4 Example of a accelerating structure after a laser beam collides with a plasma. At $x_1 = 15(c/\omega_0)$ a small beam is observed, separated from all other particles by an electron free zone. 16

2.5 Example of the k-means clustering algorithm applied to particles in a two-dimensional PIC simulation. The particles were identified by their position (x and y axis) and kinetic energy (z axis) 16

2.6 Example of the k-means clustering algorithm applied to particles in a one-dimensional PIC simulation. The particles were identified by their position (x axis) and kinetic energy (y axis) 17

3.1 Evolution of the flat time of the laser. Upper and lower bound of the blue region represent the maximum and minimum value of t_{flat} in each generation. Mean value in the generation in dark blue. In dashed red the value for π (theoretical prediction) is shown. . . 21

3.2 Optimization parameters. On a) an example of a longitudinal density profile - in the studied cases only the length of the ramp and the density at the end of the ramp were changed. This is equivalent to change the coordinates of the point in red. On b) a down-chirped wave (travelling to the right). 22

3.3 Example of a longitudinal profile of the laser across generations. The points at the laser extremities are always 0 to ensure continuity. In this case, in generation 0 the profile is a straight line. After optimization, the small changes in the first twenty generations yield a profile which is in principle better than the one the algorithm started with. 23

3.4 Set-up of the simulated cases. A laser beam is placed before the electrons and it will produce an electron beam, whose optimization is the goal of this work. Next to each element some specific variables used for optimization. 24

3.5 Evolution of $Q = n^{1/2}E/(\sigma_E + 0.1)$ (optimized quantity) 25

3.6 Evolution of the properties of the electron beam at the same iteration the instant Q was maximal. Only the best result for instant Q in the whole generation is plotted. 25

3.7 Evolution of the properties of the laser and plasma at the beginning of the simulation. . . 26

3.8 Evolution of the longitudinal profile 26

4.1 Examples of Zernike polynomials 28

4.2 Evolution of the number of particles in the 20 to 25m_ec^2 range (n_{par}) during the simulation. Upper and lower bound of the colored region in black represent the maximum and minimum value of n_{par} in each generation. Mean value in the generation appears in dark blue. The value for a Gaussian pulse is shown in dashed red. 30

4.3 Electron energy profile for Gaussian (green) and optimized (blue) beams. The profile was done for the iteration 3502, where the optimized beam had the maximum number of particles in the energy range. The energy range is represented by the two vertical lines. . 31

4.4 Histogram (energy, angle) for particles in the [15,32] m_ec^2 energy range. At the left side, the result for the Gaussian pulse at the iteration where the number of electrons in the [20,25]m_ec^2 is maximum for the optimized beam. The angle is calculated via the momentum ratio p_{perp}/p_{long}, where p_{long} was defined as the momentum parallel to the laser propagation direction. 31

4.5 Transverse electric field at the initial time. a) The perfectly Gaussian laser. b) The optimized laser with 4 Zernike polynomials. 32

4.6 First iteration at which the sum of the energies of particles with $E > 15m_ec^2$ is bigger than 20000 m_ec^2. The upper and lower bound of the colored region represent the maximum and minimum values inside the generation. The mean inside the generation is shown in dark blue. The value for a Gaussian pulse is shown in dashed red. 33

4.7 Transverse electric field at the initial time. a) The perfectly Gaussian laser. b) The optimized laser with 6 Zernike polynomials. 33

4.8 Estimated cumulative radiation originated from particles with energy $E > 15m_ec^2$.The upper and lower bound of the colored region represent the maximum and minimum values inside the generation. The mean inside the generation is shown in dark blue. The value for a Gaussian beam is shown in dashed red. 34
4.9 Transverse electric field at the initial time. a) The perfectly Gaussian laser. b) The optimized laser with 6 Zernike polynomials. 35

Glossary

GA Genetic Algorithm is an optimization method based on the Darwinian theory of evolution.

GoLP Group of Lasers and Plasmas

IST Instituto Superior Técnico

LWFA Laser Wakefield Acceleration is a technique to accelerate electrons using a laser pulse.

MPI Message Passing Interface (MPI) is a standardized and portable message-passing standard designed to function on a wide variety of parallel computing architectures.

PIC Particle in cell is a method that is used to solve partial differential equations, extremely important in plasma physics.

Chapter 1

Introduction

1.1 Plasma Accelerators

Particle accelerators are fundamental to a myriad of different areas of physics, ranging from high-energy physics (CERN) to low-energy medical appliances. Accelerated particles can also be a source of high energy radiation that may be used to perform several diagnostics and measurements with light, e.g. spectroscopy.

Radio frequency (RF) cavities are the basis for most particle accelerators. This technology is based on applying large electric fields in a metal and/or superconductor cavity. This provides a theoretical limit to the acceleration in cavities - the threshold for dielectric breakdown of the acceleration tube (electric field that damages the material, typically at a micrometer scale), that is known to be under 200 MV/m [1]. These cavities allowed major scientific findings, such as the Higgs Boson discovery, so its importance must not be undervalued.

In 1979, Tajima and Dawson [2] provided the guidelines for a new type of particle accelerator. It was called a plasma accelerator, as the medium used to sustain the electric fields responsible for the acceleration was a plasma. This medium is, in fact, a state of matter, an ionized gas highly electrically conductive where collective phenomena of electrons and ions dominate the dynamics. It differed from the conventional accelerators because it could theoretically have a much higher acceleration per unit length. Plasmas are able to sustain waves with electric fields [3] given by

$$E_0 = \frac{cm_e\omega_p}{e}, \tag{1.1}$$

or, in SI units, E_0 [V/m] $\approx 96\sqrt{n_0[\text{cm}^{-3}]}$, where e is the electron charge and m_e is the electron mass, $\omega_p = (4\pi n_0 e^2/m_e)^{1/2}$ is the plasma frequency and n_0 the plasma density. For example, with an electron density of 10^{18} cm^{-3}, the electric fields can get close to 100 GV/m, which is approximately 3 orders of magnitude higher than those of RF cavities. This is rather innovative as it may reduce the length of acceleration by the same factor. The reduction could pave way for linear accelerators able to collide electrons at energies similar to those of the protons of the LHC. The linear acceleration is essential as

electrons lose more energy than protons when they are forced to perform a circular motion (like the LHC). A smaller scale makes the total cost of the plasma accelerator smaller than that of RF cavities, easing the replacement of the latter by plasma-based accelerators.

Generally, plasma acceleration uses relativistic wakefields or plasma waves, associated with electron density waves in the plasma. These relativistic plasma waves can be excited by ultra-intense lasers (laser wakefield acceleration, or LWFA) [2] or by ultra-relativistic charged particle bunches (plasma wakefield accelerator, or PWFA) [4]. The element that excites these waves is called a driver.

For the LWFA, the laser exerts a ponderomotive force on the electrons (radiation pressure) when entering the plasma. The typical values for the laser are duration t_l=100 fs and peak intensity I_l=10 TW - 1 PW [3]. For the PWFA the wave is excited by the electrostatic forces associated with an intense and ultra-relativistic particle bunch. Here the values [5] for this driver particle bunch are σ_z=100 μm, N_p=10^{11}, E_p=1 - 100 GeV (longitudinal size of the bunch, the particle number and particle mean energy, respectively).

The ponderomotive force of an intense laser or the electric field of a particle bunch repells plasma electrons as the driver propagates through the plasma. The background plasma ions remain nearly immobile at the time scale associated with the plasma electron response, because they are around 2000 times more massive than electrons. The resulting space-charge fields result in the generation of a relativistic plasma wave, whose phase speed is close to the speed of the driver. When the driver (laser or particle beam) is sufficiently intense, it can expel all plasma electrons at its passage. In this strongly nonlinear regime, also known as the nonlinear blowout regime [6, 7], the plasma wave is nearly a sphere, which is commonly known as a bubble due to its geometry. The region void of electrons is formed at the back of the driver (in the case of the LWFA, see fig.1.1, white region). These expelled electrons are attracted back towards the axis and therefore end up forming an area with a negative charge (fig. 1.1 in black) located in the axis. The electrons overshoot this area, forming again an ion bubble. This charge separation (positive and negative charges separated along the axis) is the source of the longitudinal electric field needed for the acceleration. This wave propagates at a group velocity which is approximately the speed of light, which implies that a relativistic electron can be trapped in such a wakefield (between the electron-populated negative-charged area and the ion bubble). However, to be trapped, it needs to travel at the same velocity, in order to keep accelerating. This is quite similar to a surfer in a wave, as one needs to have a minimum velocity before catching it.

Both the LWFA and PWFA have been subjected to proof-of-principle experimental demonstrations [3, 8]. Although the fundamental aspects and properties of these accelerators are accessible through simplified analytical models, the self-consistent dynamics of the driver in the plasma, which ultimately determines the output of the accelerator, can only be captured by sophisticated numerical simulations.

1.2 PIC simulations

A plasma is a many-body system. A potential approach to simulate a problem involving a plasma is to evaluate the forces on every particle by calculating all interactions between pairs of particles, which will

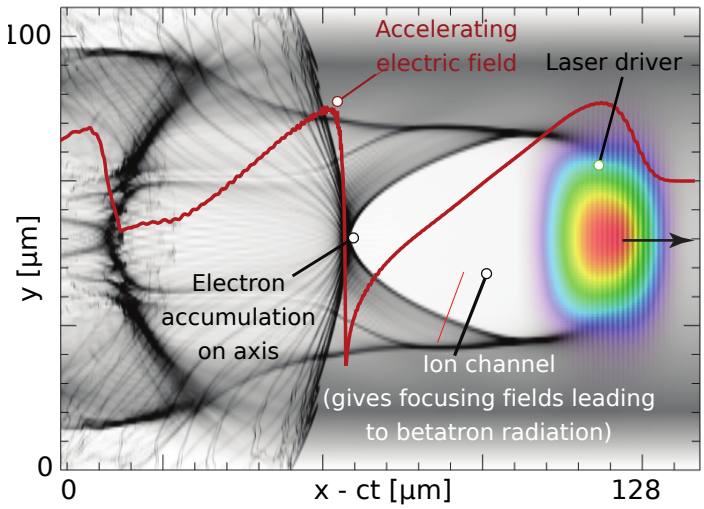


Figure 1.1: Numerical simulation of an intense laser pulse (rainbow color) propagating in plasma (grey, white corresponds to positive charges and black to negative ones), superimposed with a graph of the local longitudinal electric field in the axis, in red.

be purely electromagnetic. The electric field felt by particle i due to particle j, for example, would be given as

$$\vec{E}_{ij} \propto q_i q_j \frac{\vec{r_i} - \vec{r_j}}{|\vec{r_i} - \vec{r_j}|^3} \tag{1.2}$$

where the position of the j-th particle is taken at the retarded time. Using this method, one ends with a number of computations proportional to N_p^2, where N_p is the number of particles, which means that the simulation time would be gigantic when simulating experiments ($n \approx 10^{17}$ cm^{-3}).

There are several efficient ways to simulate a plasma . In plasma-based acceleration, particle-in-cell (PIC) simulations are among the most successful. They allow quantitative predictions, that are in good agreement with experiments and play a crucial role in exploring new concepts and ideas before testing them experimentally. With the available computing power, PIC simulations fully play the role of virtual experiments. In a PIC simulation, charged particles interact self-consistently through the EM fields defined in a grid. The PIC loop involves four key steps (i) particle push using the fully relativistic Lorentz force, (ii) current deposition onto a discretised spatial grid, (iii) electric and magnetic field update using the time-dependent Maxwell's equations, given by:

$$\frac{\partial \mathbf{E}}{\partial t} = 4\pi \mathbf{j} - c\nabla \times \mathbf{B} \qquad \frac{\partial \mathbf{B}}{\partial t} = -c\nabla \times \mathbf{E}, \tag{1.3}$$

and (iv) electromagnetic field interpolation back to the position of the particles. The loop that is used to calculate the time evolution of particles is described in fig.1.2. One needs to take into account that the initial conditions for the fields must respect the time-independent Maxwell equations.

The space derivatives are calculated using finite-difference approximation. In order to get a 2nd order method concerning the time derivatives, one deviates the E grid from the B grid by half grid spacing, such that the method is both time and space centered. The first scheme developed to obtain the evolution of the EM fields in a grid in space and in time is the Yee method, which uses the first neighbour cells to

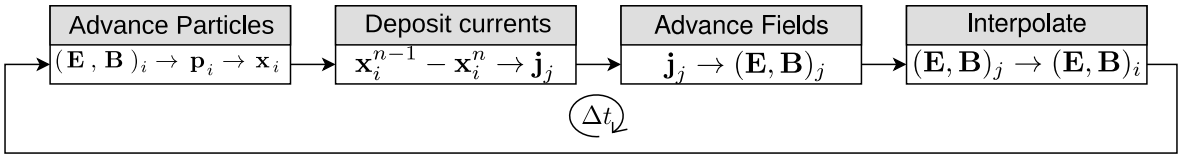


Figure 1.2: Scheme of the loop that advances both particles and fields. The index i refers to the particles and the index j to the grid points.

compute the temporal and spatial derivatives of the Maxwell equations. This is called the Yee scheme [9], and it is a standard algorithm in science and engineering. There are other higher order algorithms that provide higher accuracies (e.g. better EM dispersion properties) but the Yee scheme is an essential first step.

The computational time in PIC codes is proportional to $N_p + N_g$, where N_g is the number of grid points. This linear scaling massively improves the range of the number of particles that are treatable by the simulations. A small example illustrates the speed up from a PIC method: with 128 particles per cell and 1000 cells, and admitting that each particle/grid time advance takes 1 μs, a PIC code would need (128000+1000)$\times 10^{-6}$=0.129 s to advance in time all its elements, whereas a code that calculates every force would need $(128\times 1000)^2\times 10^{-6}$=16384 s. Another great advantage is that no major approximations are done, allowing extremely complex experimental set-ups to use this type of simulations to try to explain various phenomena. It is also worth mentioning that one can propagate a laser on the Yee grid, but that would not be possible in a particle only code as the laser must be able to propagate through a vacuum.

1.3 Numerical simulations and experiments

The interplay between advanced numerical simulations and experiments has lead to an explosion in experimental progress. Using parallel computing allows the total simulation time to decrease. Because the standard PIC algorithm requires local operations, i.e. because the motion of the particles depends on the fields from nearby grid-cells, the PIC scheme is ideal to take advantage from distributed computing approaches. By assigning a given region of the total simulated volume to different computing nodes can strongly relax the wall-clock time requirements for these simulations. Still, 3D modelling of experiments is computationally intensive as the code makes no physical approximations to the extent where quantum mechanical and gravitational effects can be neglected (the discretization of the Maxwell's equations is an approximation, but a numerical one).

However, with the advent of large super-computers, 3D modelling is now possible. The existing synergies between simulations and experiments have lead to major improvements of plasma-based accelerators. Despite the tremendous success, improving beam quality is still a major research issue. The energy gain record for a laser wakefield accelerator was recently established in experiments leading to an acceleration of 4.2 GeV electron bunches in a 9 cm plasma [10]. Early experiments at SLAC (Stanford Linear Accelerator Center), using a particle bunch as a driver, demonstrated an energy doubling of some

of the electrons of the driver from 40 to 82 GeV in a 1 meter long plasma [11]. More recent experiments at SLAC managed to obtain energy conversion efficiencies of up to 30% [12], competitive in comparison to conventional approaches. Another example of research is the EuPRAXIA project [13], where a conceptual design of a highly compact, cost-effective, industrial quality 5 GeV plasma accelerator is to be designed until 2020, funded by the EU.

Even though the mean energy of an accelerated particle bunch is immensely improved with regard to RF cavities, additional internal properties of the accelerated beams must also be tightly controlled, as they are an essential aspect of the overall quality of the beam. The energy spread, efficiency (conversion of energy from the driver to the bunch) and angular divergence (angular dispersion of the bunch) are examples of beam quality parameters that still need to be improved in experiments. The plasma technology is still not on a par with the RF cavities when it comes to these other parameters.

Since this improvement is necessary, optimization is required to get a better output beam quality. The number of free available parameters that can be used to control these properties and improve beam quality is very large.

There is flexibility to change the details of the plasma density profile. In the case of a laser pulse driver, is is also possible to vary the details of the laser profile and internal spatio-temporal structure, such as dimensions, intensity, envelope profile and phase profile. Because experiments are expensive, it is not possible in practice to test and investigate all these possibilities in the lab.

Computer simulations are thus promising tools to explore the wide range of parameters that are available. There are no analytical models capable of describing laser plasma accelerators and that includes the exact role of the plasma density profile, laser pulse profile and internal structure. In fact, one of the most widely used analytical models to describe laser plasma accelerators has only two free parameters (the laser energy and plasma density), thus completely disregarding the role of any other parameter. As a result, exploring the vast parameter space that is currently available to control the acceleration and the output beam properties would require extensive parameter scan simulations, covering the entire range of parameters that can be envisioned. Because of computational constraints this is not possible even in the largest supercomputers. In this thesis, a radically new approach that uses machine learning is proposed to address this challenge and facilitate the parameter search

. To regulate the phase-space quality, the laser-plasma initial conditions should be treated carefully. It is possible to use simulations to provide the conditions that optimize the beam quality. Unfortunately, these simulations are often computationally expensive, so a regular parameter scan is not desirable to tackle the optimization problem.

It has already been discussed [14] that in non-linear cases (like this one) machine learning techniques may not only be useful to predict results but to model the process itself.

A machine learning algorithm, based on genetic algorithms, was incorporated in the PIC scheme. The algorithm was developed in the PIC code ZPIC. ZPIC is a free, open-source project (`https://github.com/zambzamb/zpic`), that is being developed at the group of lasers and plasmas (GoLP) at IST.

The coordinates in ZPIC are normalized for faster performance and improved accuracy. This requires a normalization reference frequency, ω_n. Time is normalized to $1/\omega_n$. The density is normalized to a

Variables	Normalized units
Space	$x' = \frac{\omega_n}{c} x$
Velocity	$v' = \frac{v}{c}$
Momentum	$u' = \frac{u}{c} = \frac{\gamma v}{c}$
Electric Field	$E' = e\frac{c/\omega_n}{m_e c^2} E$
Magnetic Field	$B' = e\frac{c/\omega_n}{m_e c^2} B$

Table 1.1: Normalized units in ZPIC

background density n_0. The electric charge is normalized to e and the masses are normalized to the electron mass m_e. The normalization for the electromagnetic fields follows. Some normalized units are shown in Table 1.1.

ZPIC can run in a standard desktop or laptop computer requiring only a C compiler to install. It is a simple PIC code that incorporates all the essential features of the PIC algorithm, without having advanced features. Because it is easy to read and modify, ZPIC is an ideal tool to incorporate a machine learning algorithm. The final goal is to have a machine learning controlled PIC simulation code.

1.4 Machine learning

Machine learning algorithms have been employed in several fields, and its importance has been growing over the last few years. Some authors predict [14] that it should be as common as numerical simulations in the future to come. Machine learning, generally speaking, is the field that provides computers the ability to learn without being explicitly programmed.

Machine learning is booming. This kind of approach is motivated by the increased computational capacity. Some fields have been completely taken over by these techniques, like computer vision [15].

In October 2015, AlphaGo defeated the best European Go player [16]. Eventually, the same algorithm beat the best player in the world, and was selected in 2016 by *Science* as one of the runner-ups for the Breakthrough of the year award. Google was able to create an even better algorithm, *AlphaGoZero* [17], that required no human interaction. It gained knowledge by playing against itself 4.9 million games, something unthinkable for a human.

It is also worthy mentioning that one can propagate a laser on the Yee grid, but that would not be possible in a particle only code as the laser must be able to propagate through vacuum.

Artifical intelligence (and in particular deep neural networks) were substantially improved over the course of the past decade. Genetic algorithms also improved during this time, benefiting from the research on neural networks [18].

What has been delaying the adoption by the physical community is the lack of transparency of the algorithms, which means that it is difficult to extract knowledge from the raw output, as the internal processes of the methods are often hidden from the user. However, because simulations provide the full 6D (3 positions plus 3 momenta) phase-space of the particles and field evolution in space and time, this approach provides a path to potentially generalize the findings of the machine learning algorithm to new situations, and make theoretical models to justify them.

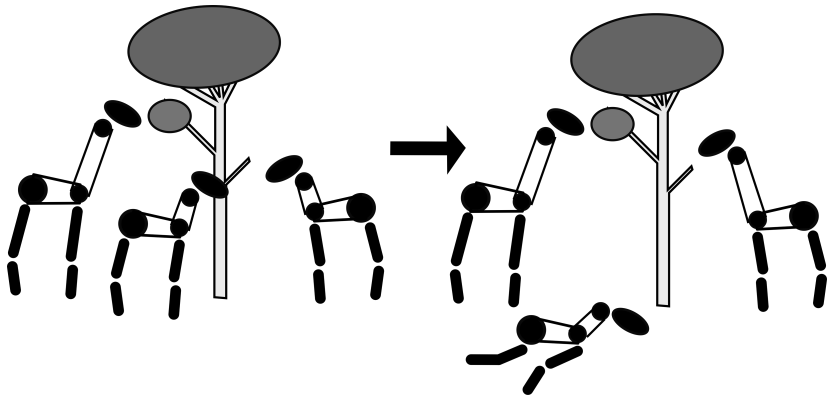


Figure 1.3: Image depicting an example of a genetic algorithm. The fittest elements can produce other offspring, something that is denied to the remaining individuals. This will produce a next generation that will be fitter.

Genetic algorithms (GA), which are standard optimization methods [19], mimic the process under which the living beings give their characteristics to their offspring. First several random ”subjects” (input parameters) are created. Then, the inputs are rated according to a fitness function (in this case, the ”subjects” would be simulations and the fitness function could be the largest electric field at some definite time). From all the inputs only the fittest (with the best fitness function) are chosen to create the offspring for the next generation of inputs. The modifications include mutations (small random change of the parents' inputs) and crossovers (two fit members of the population are used to form a new one). Figure 1.3 elucidates this concept - the giraffes that are fitter (heads closer to the leaves) are able to produce offspring whereas the less fit do not, making the next generation taller.

Machine learning algorithms have already been applied alongside PIC simulations [20]. Multi-layer perceptrons (MLP, artificial neurons) were employed to study high-order harmonic generation. The final goals were to predict optimal interaction scenarios for producing harmonics with certain characteristics, such as brightness, efficiency or wavelength range, which was done with several architectures of neural networks. This approach uses neural networks to predict the results of the simulations. Machine learning, however, has not yet been incorporated in PIC simulations with the goal of optimizing the initial simulation parameters towards a specific goal. One of the disadvantages of neural networks [21] is that they require a large amount of data before it is effectively applied, which doesn't blend well with the computational time it would cost. Genetic algorithms are ideal to this end because they can optimize the search over a wide parameter space.

Genetic algorithms have already been tried in the control of real plasma experiments [22]. To change the wave front of a laser, the voltages of some piezoelectric actuators of a deformable mirror were used as the set of unknown variables that served as individuals of the population. This new methodology allowed to control the energy gain of a low energy electron beam (tens of keV) in a laser-plasma accelerator by acting on the internal wavefront structure of the laser. This is a very promising experimental result. Numerical simulations controlled by machine learning could significantly expand these results into novel territories, in higher energy accelerators, or to enhance radiation generation. In addition not all variables are known in an experimental setup, making it limited compared with this approach - an ex-

periment may in principle measure the energy of the beam only at a finite number of locations, whereas our simulations can give insight over temporal and spatial evolutions of the most important parameters. In the particular experiment, the electron beams were collected once at a predetermined distance from the laser focus. Therefore, it is likely that the score obtained by a beam would change if the beam was detected at a different spatial location.

Chapter 2

Genetic algorithm controlled PIC simulations

The initial parameters of a PIC simulation are defined in a text file, usually known by input-deck. Commonly the input-deck is designed and written by a user, who also launches the simulation in a computer. The goal is to use machine learning to write the input-decks automatically and to design a procedure that launches the simulations in parallel in a supercomputer, analyses the results, launches new simulations, and repeat this process until the results converge, all without human intervention, except for the initial definition of the goal of the simulations and the boundaries for the machine learning guided search. This procedure is illustrated in Figure 2.1.

The algorithm was kept as generic as possible, so it can be applied to a plethora of optimization cases. Thus, before running the algorithm, one should prepare two files, the text file, which was called the initializer, with the initial values to be explored, and the input-deck. For example, an optimization of the length of the laser requires an initial set of values for the laser length to be used in each simulation. These values can be initially chosen randomly, as long as each set of points is valid (the user can choose conditions for the validity - an example could be of a time being larger or equal to 0).

2.1 Initial input to the genetic algorithm

The listing below shows a portion of the input deck file where a laser is added to the simulation with its starting point, a_0=10 (normalized vector potential), ω_0=10 (ω_p) (laser frequency) and polarization constant = 10π. The rise, flat and fall time of the laser are to be determined - in the first generation those values will be selected by the user itself (via the initializer), and in posterior generations the genetic algorithm will automatically determine the best values for these variables in order to optimize some property.

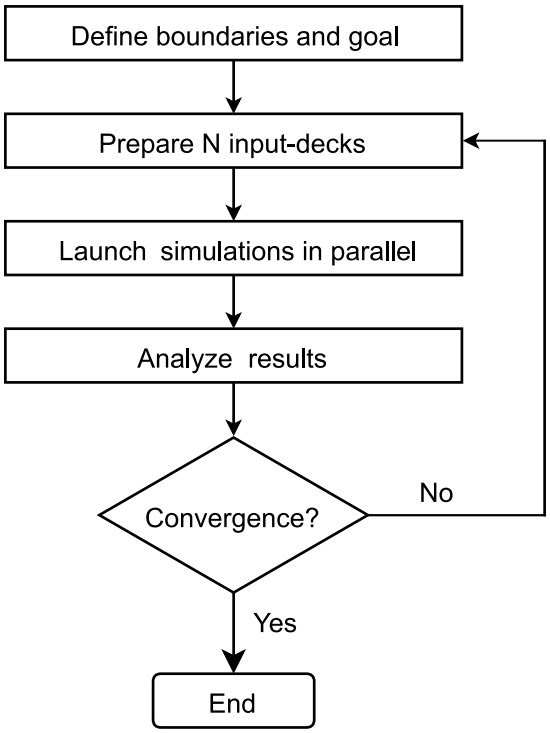


Figure 2.1: Flow chart of the general procedure

```
t_emf_laser laser = {
  .start =10,
  .rise=variable[0],
  .flat=variable[1],
  .fall=variable[2],
  .a0 = 10,
  .omega0 = 10,
  .polarization = 10*M_PI
  };

sim_add_laser( sim, &laser );
```

The initializer can be written with the help of a simple python program. Each line consists of the variables of each simulation, separated by whitespace. Every line of the initializer is read and stored into the `variable` vector of the input deck (the first column of the initializer matches `variable[0]`, and so on). The initializer can be used to control every aspect of the simulations simultaneously. In this work the search was restricted to the physical parameters of the simulation (e.g. laser dimensions, plasma profile) but the numerical aspects (e.g. resolution, box size, etc.) could have been changed as well. An example of an initializer preparing 4 simulations with 2 free variables each is presented in `init_example.txt`.

init_example.txt

```
#Lines that start with # are not read
#All these lines work
1        0
                1        1
1        3
1                    4
```

2.2 Genetic algorithm

The genetic algorithm used for optimization is shown in fig.2.2. After both files are created, the number of simulations (n_scan, chosen by the user) is equally distributed by the available computer nodes. Because ZPIC can only run in a single core, it was designed a script that launches simultaneously n_scan simulations, one per computer core. This procedure is fully automatized, and can run in an arbitrary number of computing cores. Each core will attribute a score to each simulation. The score is a measure of how good the results yielded by the simulation are. These scores are then sent through MPI routines to a root core, which is responsible to store the scores from every simulation of a given generation, choosing the best scores, making the necessary changes and preparing the next set of inputs. A buffer text file is created and the root core gives the order for the new simulations to be launched.

At the end of the algorithm the results for all generations can be found in the same folder as the text file that defined the first set of inputs in a straightforward manner - if the original file is called acceleration.txt the results for generation 0 are saved into acceleration_000000.txt, and the results for the remaining generations follow the same pattern. In order to collect and plot the results simple python programs were designed.

The genetic algorithm was implemented in C, as it was the language of the original ZPIC program. To perform the parallelization, MPI was used. This allowed for the program to be run in the Accelerates cluster at GoLP running up to 1580 simulations at the same time.

At the end of the runs in each generation, the goal is to take the parents (inputs from the simulations that just ended) to create the children (inputs that will be run afterwards). The best N_MUTATIONS (number of the parents that will effectively be used to produce children) scores are selected. The respective inputs are mutated to create new inputs, using the standard expression found in

$$\text{new\_input} = \text{input}\ (1 + c_0 \times \text{random}()) + c_1 \times \text{random}(), \tag{2.1}$$

where random() is a function that returns a random number between -1 and 1 and c_0, c_1 are rates of change that are responsible for how different the children and the parents are. Normally c_0 is the same for all variables and may change at runtime: if the algorithm keeps finding a higher score then it is good practice to let c_0 grow so it converges faster, but if the scores plateau one should decrease it so that a

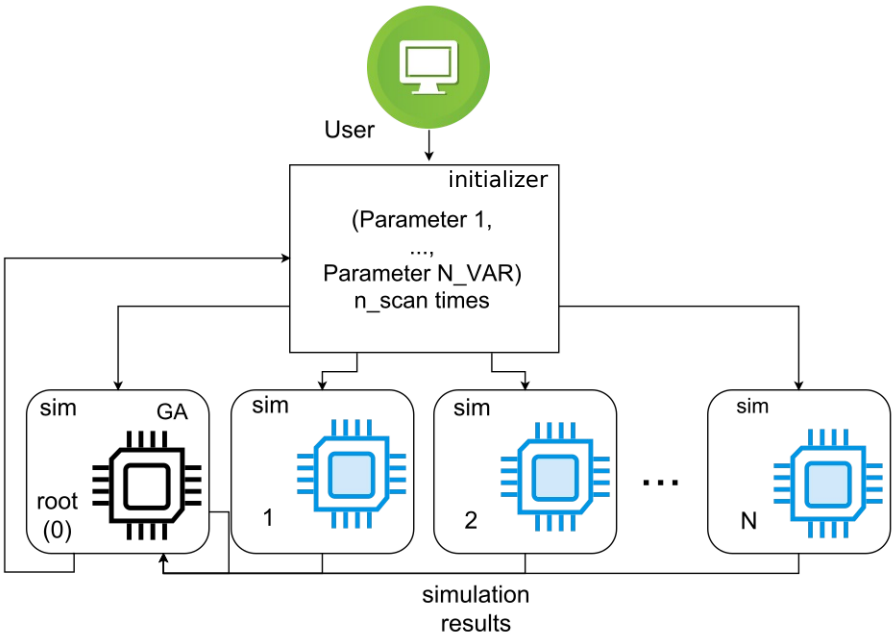


Figure 2.2: Diagram of the algorithm: From the initializer the several input-decks are distributed for the available computer nodes. The simulation results are sent to the root core, which creates a new text file with new parameters which are again fed to the available nodes, until convergence is reached.

local maximum is reached. The c_1 variable serves the purpose of not letting a variable be kept constant if the initial value is 0: as seen in Equation 2.1, every time c_1 is 0 and `input` is 0 the `new_input` variable is 0 no matter the values obtained by the `random()` function.

The number of times the same input parameters are mutated depends, of course, of the `n_scan/N_MUTATIONS` ratio - if the half of the simulations serve as the parents for the next generation, each of those sets of inputs will give origin to two descendants. There is also the possibility to reject the new set of inputs obtained with Equation (2.1) and create a new one if they do not obey certain criteria (a particle density above zero would be an example of a mandatory condition).

All the metavariables of the algorithm, such as the maximum number of generations, the number of set of inputs (individuals), the number of parameters for each individual and the rates of change, are selected by the user.

2.3 Launching machine-learning controlled PIC simulations

In section 2.1 part of an input deck was shown. In the main file the user decides to take the maximum of the electric field and the electron density at some determined time. If the user decided to have a simple initializer `data.txt` like shown below, three simulations would be run in the first generation, with $(t_{rise}, t_{flat}, t_{fall}) = (1, 2, 3)$ for simulation 0, $(t_{rise}, t_{flat}, t_{fall}) = (2, 3, 1)$ (the rise, flat and fall time of the laser, respectively) for simulation 1 and $(t_{rise}, t_{flat}, t_{fall}) = (3, 1, 2)$ for simulation 2. The next generations are determined by the results of these first simulations, as they depend on the genetic algorithm.

data.txt

```
# Lines that start with # are not read and do not transpose to
# the following text files.
# Inputs: trise; tflat; tfall
# Outputs: max e field; max density;
1 2 3
2 3 1
3 1 2
```

In this particular case two outputs are taken from the simulations - maximum of the electric field and electron density. The results of the first generation can be accessed via the text file `data_000000.txt` after the first set of simulations is completed (this particular output file does not come from the algorithm and only serves an illustrative purpose).

data_000000.txt

```
1 2 3        0.0001        0.0003
2 3 1        0.0002        0.0002
3 1 2        0.0003        0.0001
```

Supposing the variable `N_MUTATIONS` was 1, only the best simulation will be used to create the new set of inputs. This will depend on the variable the user chooses to optimize (the maximum electric field or the maximum density in this example)

If the maximum electric field is selected, the algorithm sorts the results by the first output column, selecting line 3 with inputs (3,1,2) and tweaks the inputs using Equation (2.1). These inputs are then stored into the `buffer.txt` file. This assures that in case one wishes to stop the program and resume it they can restart at the latest generation.

buffer.txt - optimization of first output column

```
3.11 1.13 1.99
2.94 0.99 2.14
3.03 0.94 2.02
```

Using `buffer.txt` as the input of the program instead of `data.txt`, the user resumes what was already done and thus the first set of simulations does not have to be run again.

2.4 Automatic PIC simulation analysis: Clustering and k-means

One of the major goals of this work is to improve the properties of the accelerated beam. Thus, the first goal was to automatically isolate an accelerated electron bunch from the background plasma electrons.

The standard way to identify particles that belong to a beam is to detect energies of the particles and select a threshold that is able to group the majority of the accelerated particle bunch. However, this method is *ad hoc*, as the threshold needs to be selected manually so that the 'beam' appears to be a separate entity.

In PIC simulations, the momenta and positions of each particle are accessible at all times. The possibility of getting beam properties at runtime is very helpful. In this case, running several simulations at the same time requires an automatic way to identify the beam. Because the 2D beam has a specific spatial shape, a standard clustering technique was employed to classify particles as part of the beam or not.

In machine learning problems, the supervised learning branch refers to when the correct answer is known for some cases. An example would be to tell what is in a new picture provided one has that information for a number of different pictures (which would be used to train the classifier).

In this case, the electrons which are part of the beam do not have a specific label that easily allows the separation between them and the background electrons that support the accelerating structure. This means that the correct answer is not known and therefore beam classification is an unsupervised learning problem.

The simplest way to approach such problems is through clustering. A clustering algorithm takes a set of objects and divides them into groups (clusters). In this case, one of such clusters would ideally represent the beam. Specifically, the k-means clustering algorithm was used.

This algorithm depends on the concept of distance. If x_n^i refers to the n-th variable of object i then the distance between points a and b is calculated according to:

$$d_{ab} = \sum_n (x_n^{(a)} - x_n^{(b)})^2 \ . \tag{2.2}$$

In the k-means algorithm, the number of cluster is predetermined (k). From the whole set, k objects are chosen as centres of the clusters. These centres can be selected randomly between the full set of objects but if the distribution of the objects is known this information may be used to speed up the process. After choosing the centres for the first iteration, the k-means loop involves three steps (i) The distance between every non-centre point to the centres is calculated, (ii) Each point is assigned to the cluster whose centre is closer (smaller distance), (iii) New centres are calculated from the mean of the points in the cluster in that iteration, If after step (ii) the points keep the same assignment as in the last iteration convergence is reached and the algorithm stops. An example of the application of this method is presented in Figure 2.3. This method is general, as the axis could represent any phase space variable (position, momentum) or functions of the phase space variables (e.g. kinetic energy, transverse momentum, etc.).

In this work, the chosen variables were the particle position (the number of independent variables depends on the code that is being used - a one dimensional PIC code will have only one variable and a two dimensional one will have two, x_1 and x_2, for example) and kinetic energy. This is easily understood as a beam in a two dimensional simulation normally has a defined position (Figure 2.4)and the particles

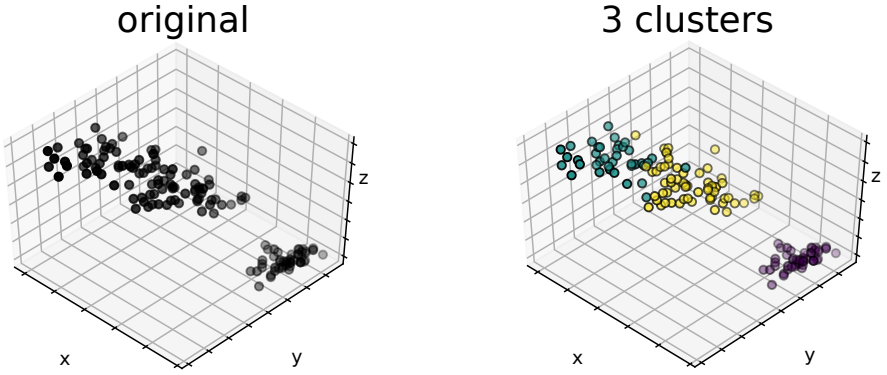


Figure 2.3: Application of the kmeans method. The starting point is a set of points identified with some variables (x,y and z) with no labels (left). The number of clusters (3) is predetermined. At the convergence point each object is assigned label, identified as a color in the plot (right).

of the beam have a higher energy than the background.

In these simulations, particle number can be of the order of millions. Because the time until convergence in clustering algorithms increases with the number of objects, a threshold energy was selected in order to reduce that number. This is fundamentally different than the threshold energy discussed at the beginning of this chapter - this new threshold only acts as a way for the algorithm to consider less particles. For example, a good threshold energy could be as low as 1 MeV, with the beam itself having more than 15 MeV.

In a 2D simulation, a gaussian laser with a length of 4 c/ω_p, W_0=4, and peak electric field $E_{max} = 10m_e c\omega_p e^{-1}$ was launched against a constant plasma density n_0. After the accelerating structure is defined, the particles were subject to the k-means algorithm with k=2 (Figure 2.5a) and k=3 (Figure 2.5b).

The results seem to be rather satisfactory - good beam separation was obtained with both two and three clusters. In the case with two clusters, two labels were defined: 'Beam' and 'Not beam'. The eye test suggests that the beam is defined up to a waist (where the beam gets slightly larger in x_2) and that would probably be very similar to where a human would choose the threshold energy. In the case of three clusters, three labels were defined: 'Beam', 'Semi-beam' and 'No beam'. The 'Beam' here appears to collect less particles than a human would, as there is no apparent reason to have a threshold at that specific energy. The 'Semi-beam' collects the remaining particles that were in the 'Beam' category in the two clusters case and some more. Both the 'Beam' category in 2D and the 'Beam'+'Semi-beam' category in 3D yield results that are very similar to what a human would choose with a threshold. This is useful to determine, for example, the number of electrons, the electronic spatial configuration, the mean energy, and other variables at runtime.

It is important to take into consideration that in these simulations the size of the simulation box is important. The simulations run for this section only had the length to accommodate one of these accelerating structures. Having a box with two or three accelerating structures and respective electron beams requires more clusters so all the beams can be isolated from one another and from the background

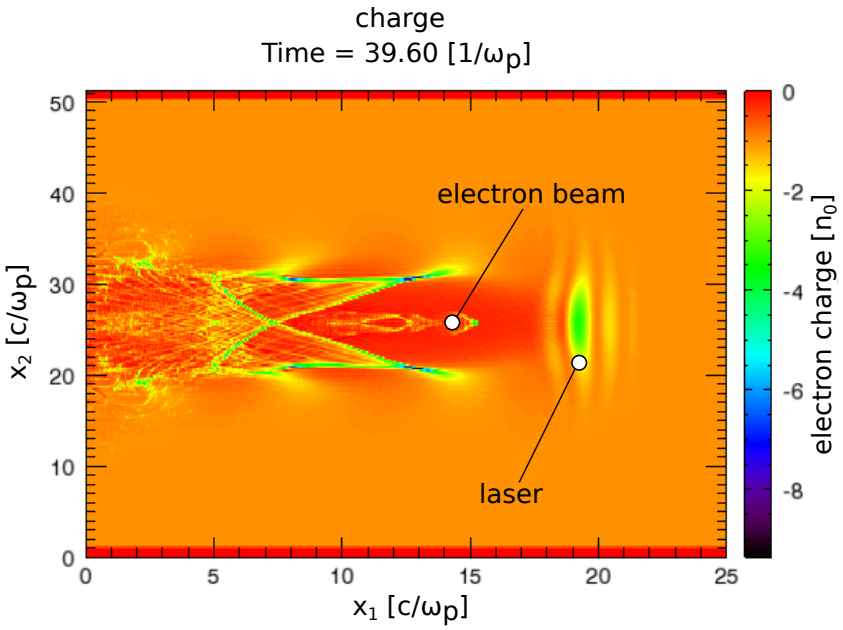
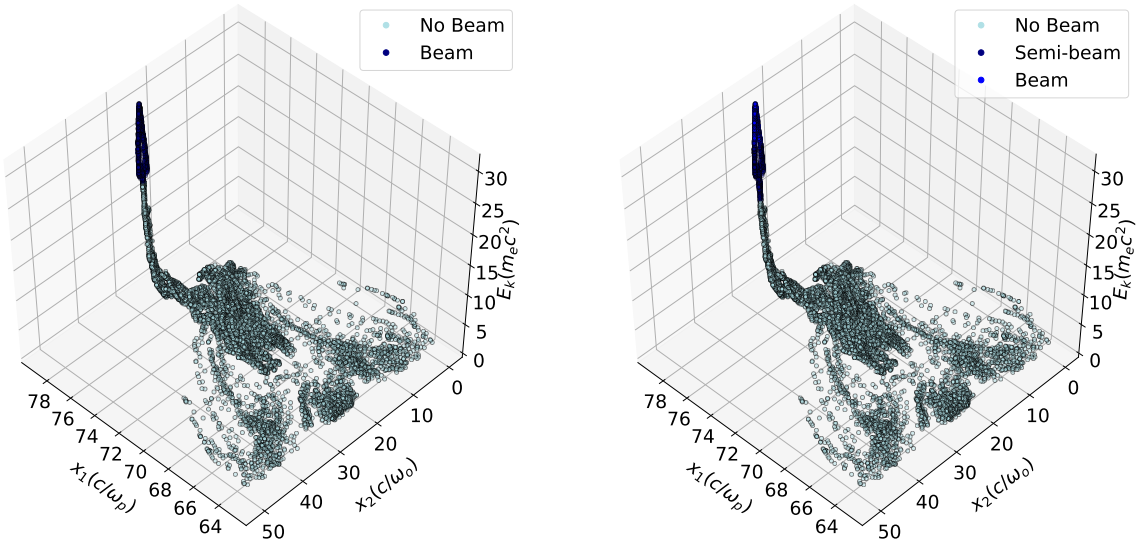


Figure 2.4: Example of a accelerating structure after a laser beam collides with a plasma. At $x_1 = 15(c/\omega_0)$ a small beam is observed, separated from all other particles by an electron free zone.



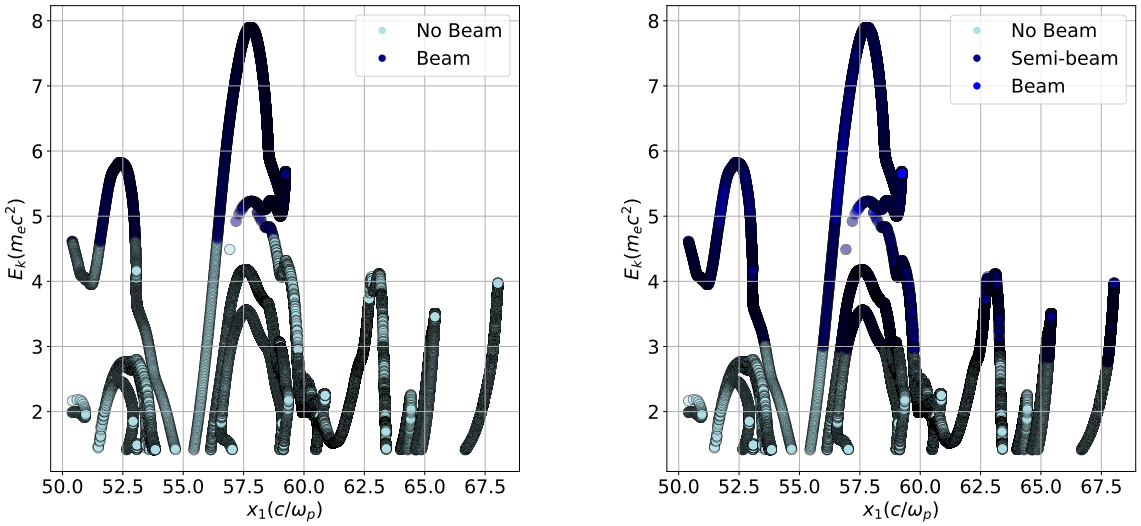
(a) Two clusters

(b) Three clusters

Figure 2.5: Example of the k-means clustering algorithm applied to particles in a two-dimensional PIC simulation. The particles were identified by their position (x and y axis) and kinetic energy (z axis)

electrons. However, if having beam statistics from the second or third accelerated electron beam is important, the k-means algorithm should in principle suffice provided k (number of clusters) is enough.

In the case of 1D simulations the k-means algorithm can still be applied. However, if the position and energy of the particle is chosen, it is seen that no good separation of the beam can be done, as shown in Figure 2.6a (2 clusters) and Figure 2.6b (3 clusters). Even when other variables were chosen, such as the transversal momentum of the particles, the algorithm did not output results as good as the ones obtained in 2D - there are particles which are labelled as part of the beam that are one plasma



(a) Two clusters

(b) Three clusters

Figure 2.6: Example of the k-means clustering algorithm applied to particles in a one-dimensional PIC simulation. The particles were identified by their position (x axis) and kinetic energy (y axis)

length far from each other. This is understandable - there is no accelerating structure governing the background electrons and therefore it is harder to isolate the beam, i.e. there is no spatial separation between accelerated and background electrons and therefore fewer differences in the fields felt by both groups.

The clustering algorithm thus provides the required framework to explore 2D simulation results automatically.

Chapter 3

Application to one-dimensional scenarios

The 1D version of a PIC code takes a four-dimensional phase space into consideration (one spatial dimension and three momenta). The genetic algorithm was first applied to this subset of problems as the runtime is smaller than that of extra spatial dimensions, and because it allows to benchmark against conditions where the optimal configuration is known theoretically. A number of configurations was explored with the goal of improving the outputs of a laser wakefield accelerator in 1D.

The number of optimization parameters is restricted, as important components of the acceleration such as the laser wave-front have to be completely discarded (only plane waves are allowed). In order to study more parameters than those offered by the original ZPIC, new features were added, such as the possibility to introduce a chirp and to have a fine control of the longitudinal profile of the laser.

3.1 Using machine learning to recover known optimizations

In order to check if the genetic algorithm can be successfully applied to PIC simulations, a simple benchmarking test was conducted. According to Esarey [3], the equations for the density perturbation normalized to the background plasma density n_0 ($\delta n/n_0$) and electric field of the wake (normalized to E_0, defined in Equation 1.1 that is driven by a laser beam with $a^2 \ll 1$ (normalized vector potential) and $\omega_0 \gg \omega_p$ (laser frequency much bigger than plasma frequency) are given respectively by:

$$\delta n/n_0 = (c^2/\omega_p) \int_0^t dt' \sin[\omega_p(t-t')] \nabla a^2(r,t')/2 \tag{3.1}$$

$$E/E_0 = -c \int_0^t dt' \sin[\omega_p(t-t')] \nabla a^2(r,t')/2 \tag{3.2}$$

Solving these equations provides information about some known features of the optimized laser

profile. If the laser is perfectly flat, for instance, then

$$a(x,t) = a_0 H(-(x - x_{st}) + ct) H((x - x_{end}) - ct), \tag{3.3}$$

where H is the Heaviside step function and x_{st} and x_{end} the positions that limit the flattop. The derivative gives two delta functions. The effect of the δ at the front of the laser is a push forward on the plasma electrons, which sets up a plasma wave. The effect of the δ at the back of the laser is a push backwards on the plasma electrons. The final amplitude of the plasma wave depends on the timing between the two pushes, which is set by the laser duration. The final result for the longitudinal electric field is given by:

$$E_x(x > x_p, t) = \begin{cases} 0 & , x > x_{st} + t \\ -sin(t - (x - x_{st})) & , x_{st} + t > x > x_{end} + t \\ -sin(t - (x - x_{st})) + sin(t - (x - x_{end})) & , x < x_{end} + t \end{cases} \tag{3.4}$$

In this particular case, the best shape for the laser for optimization of the maximum longitudinal electric field comes from a $x_{st} - x_{end} = \lambda/2$, where λ is the plasma wavelength[3].

The genetic algorithm was run with 20 subjects, 5 of which were selected to mutate. A laser with $a_0 = 0.05$, $\omega_0 = 10\ (\omega_p)$, and variable length (free parameter) initialized in vacuum and propagated towards a plasma. The plasma density was constant and equal to n_0. The fitness score of each simulation was the highest electric field in the grid.

In Figure 3.1 the results for the genetic algorithm are shown. The algorithm converged to a solution which is very similar to the theoretical prediction of π (0.9% error) but still deviated. However, when the exact value was tried the result was slightly worse than the one found by the algorithm which shows that the algorithm is finding the maximum, but numerical aspects such as the resolution are responsible for the aforementioned deviation.

3.2 Enhancing beam quality

Since the algorithm is able to retrieve known results in a simple configuration, the next step was moving into a more complex configuration where there is no analytical solution. The goal was to increase beam quality.

A definition of beam quality depends strongly on the specific application for the beam. There are however a few features that are desirable across different applications. Beams with low energy spreads σ_E, with high energy, and high number of particles are usually sought in any application (e.g. in HEP, radiation emission, etc.). A measure of beam quality Q is therefore defined as:

$$Q = \frac{n^j \bar{E}}{\sigma_E + \sigma_{min}}, \tag{3.5}$$

where

$$n = \sum_{i, E_i > E_t} 1, \text{ is the number of particles in the beam,}$$

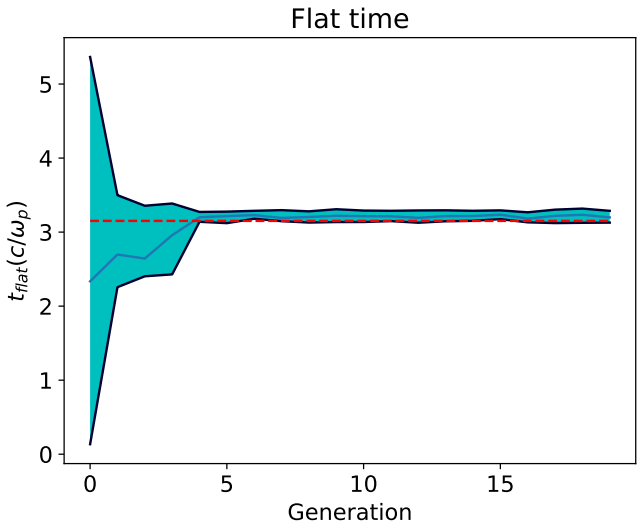


Figure 3.1: Evolution of the flat time of the laser. Upper and lower bound of the blue region represent the maximum and minimum value of t_{flat} in each generation. Mean value in the generation in dark blue. In dashed red the value for π (theoretical prediction) is shown.

$$\overline{E} = \frac{1}{n} \sum_{i, E_i > E_t} E_i, \text{ is the mean energy of the beam,}$$

$$\sigma_E^2 = \frac{1}{n} \sum_{i, E_i > E_t} \frac{1}{n} E_i^2 - \overline{E}^2 \text{is the energy spread of the beam,}$$

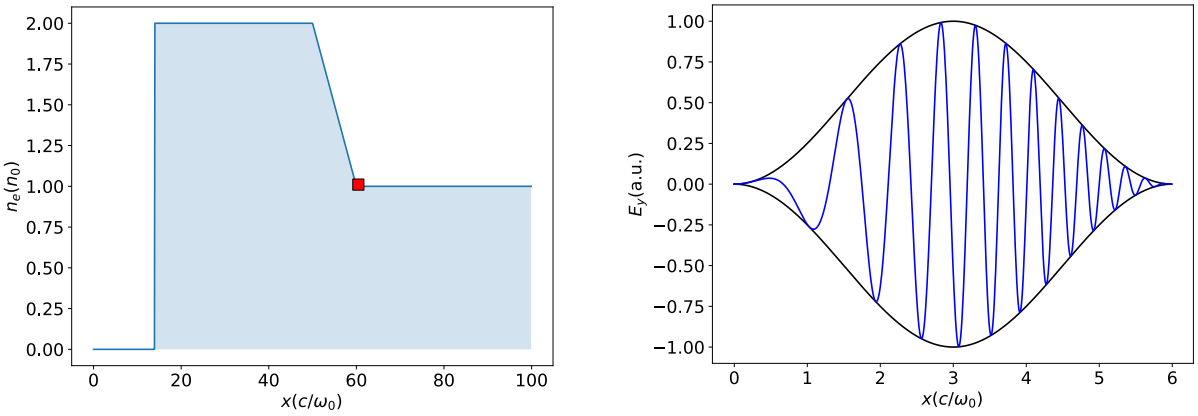
and σ_{min} is an auxiliary parameter: it was seen that σ_E dominated the result for Q, since the algorithm was most of the time converging to a final set-up with a very low relative energy spread, and because of it, variations of σ_E are more important to the overall final result. This led to a different equation whose purpose is to take importance from the σ_E variable (Equation (3.5)) using another variable σ_{min}. This variable is to remain constant throughout the algorithm and allows to fix a minimum spread under which the minimization of the spread should not be a priority. The motivation behind the j parameter (exponent of n in Equation 3.5) is to allow the user to tune the relative importance of the two variables in the numerator - the mean energy and the number of particles. A value of j>1 will give more importance to the number of particles, as doubling n will have a greater impact than doubling $\bar{E}$. The reverse happens for j<1.

In order to perform the optimization a number of initial parameters from the laser (chirp and longitudinal profile) and from the plasma (density profile) have to be made free. These parameters are thoroughly described below.

3.2.1 Optimization parameters

Density ramp

The electron density profile is extremely important for LWFA. The electric field generated by the wake is dependent on the local electron density (Equation (3.1)). Because it is hard, experimentally, to have a control of the density at every point along the laser trajectory, a simple ramp was implemented (with the



(a) Plasma longitudinal density profile

(b) Chirp

Figure 3.2: Optimization parameters. On a) an example of a longitudinal density profile - in the studied cases only the length of the ramp and the density at the end of the ramp were changed. This is equivalent to change the coordinates of the point in red. On b) a down-chirped wave (travelling to the right).

free variables being the length of the ramp and densities at the start and end point).

Physically, the wake changes as the density change. If the slope of the density is negative, the wake starts with a smaller wavelength. As the density decreases, the wavelength gets larger and the wake is able to reach more electrons, increasing the number of particles in the accelerated bunch. The whole process is called downramp injection [23].

Chirp

The frequency of the laser can be changed as a function of time, as seen in Figure3.2b.

Enabling this feature in LWFA acceleration provides an extra degree of freedom in the shape of the accelerated electron spectra [24]. In this code, a simple chirping factor was introduced as shown in Equation (3.6). The number n can be selected in the input deck.

$$\omega(t) = \omega_0 + t\alpha_1 + t^2\alpha_2 + ...t^n\alpha_n \tag{3.6}$$

Since the spatial resolution of the system is directly determined by the laser frequency (because $\omega > \omega_p$), there is a maximum frequency that a laser should have depending on the number of grid points (20-30 points per wavelength are required to properly resolve the laser beam). That way, if the algorithm detects that due to the chirp, the frequency will be above that maximum, the outputs have to be rejected and a new output is requested to the algorithm.

Laser longitudinal profile

The laser longitudinal profile has also a preponderant effect in the outcome of the experiments. If the energy and shape are kept constant, a longer pulse will necessarily lower the peak intensity of the laser, which can have dramatic effects if it is below the ionization threshold (below it no electrons are taken out

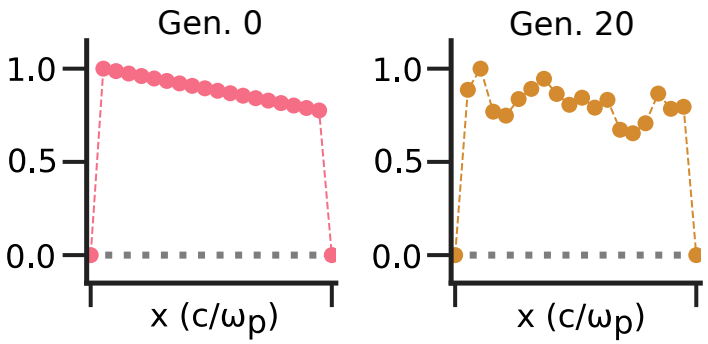


Figure 3.3: Example of a longitudinal profile of the laser across generations. The points at the laser extremities are always 0 to ensure continuity. In this case, in generation 0 the profile is a straight line. After optimization, the small changes in the first twenty generations yield a profile which is in principle better than the one the algorithm started with.

of the atom and thus no acceleration takes place).

In order to be able to change the profile, the user is allowed to select the number of points for the interpolation, the relative intensity in each one and the total length of the pulse. Two interpolation methods can be chosen, a simple line interpolation or a cubic spline. This forces the electric field to be continuous, which is a mandatory condition for an electric pulse in vacuum. An example of such profiles can be found on fig. 3.3.

If the genetic algorithm changes the inputs as equation (2.1) suggests, it will create a bias at points with lower intensities, since the intensities at such points will change at smaller rates (the change is linearly dependent on the input). To bypass it, two steps are followed. Firstly, the maximum intensity for all the points is calculated and there is a recalibration such that the maximum is always one (this balances all input sets among each other). Then, all points are changed according to the equation (2.1) except as if `input` was one (to balance all points in the same set).

The use of the 1D PIC code hides some physical phenomena that occur at higher dimensions. For example, the bubble effect analyzed at section 1.1 has no analogous in a PIC code with one spatial dimension.

3.3 Results from machine learning optimization with 1D PIC code

The output variables of the algorithm should only take into account particles that are part of the final accelerated electron bunch. Particles were considered accelerated when their energy was above a certain threshold E_t. The threshold has to be chosen by the user, as there is no structure governing the electron acceleration, unlike the ones seen in 2D or 3D. For all the runs the energy of the initial laser was kept constant.

There is also the possibility of easily adding new variables, like transverse momentum, beam efficiency, maximum energy.

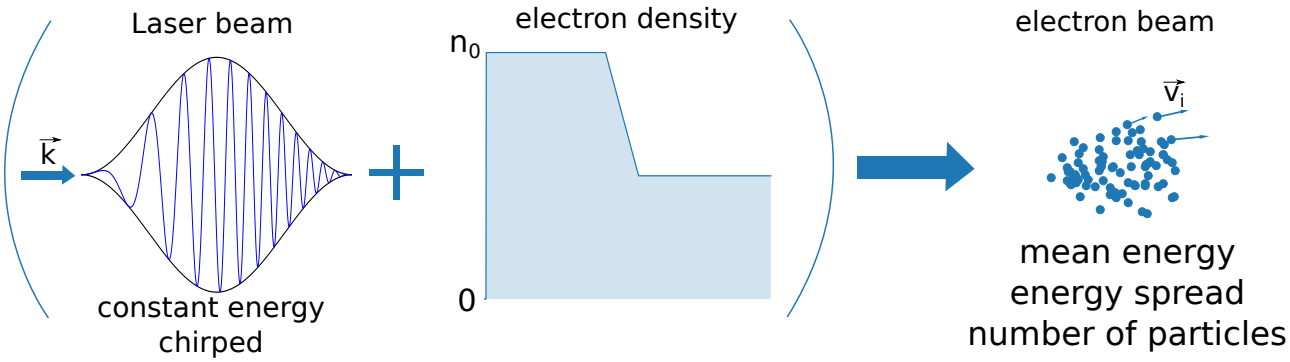


Figure 3.4: Set-up of the simulated cases. A laser beam is placed before the electrons and it will produce an electron beam, whose optimization is the goal of this work. Next to each element some specific variables used for optimization.

3.3.1 Simulation parameters

In order to study laser wakefield acceleration, the laser beam was placed before the plasma itself (Figure 3.4). The simulations were run with a moving window: the finite difference models can be run using a moving simulation window that moves at the speed of light. This allows following the relevant phenomena in display here: relativistic electron beams and laser pulses. This does not mean that the phase space variables are calculated in a frame moving at c: these values are still calculated in the lab frame, but this allows for a smaller box and thus a smaller computational time.

The laser beam is in 1D so only plane waves can be placed inside the simulation domain. The total length of the laser was limited to 9 c/ω_p and 20 points (18 of them free) were used to characterize the longitudinal profile.

The laser energy E_l is defined as

$$E_l = \sum_i (E[i]^2 + B[i]^2)\, dx, \tag{3.7}$$

where $E[i]$ ($B[i]$) represents the electric (magnetic) field in the cell i and dx was the grid size then the laser energy was constant and equal to 50 $c^3 m_e \omega_p / e^2$.

The number of particles per cell was 128 (standard) and the number of cells was chosen such that a single laser oscillation could always be resolved with 20-30 cells: as the laser initial frequency was $\omega_0 = 2\omega_p$ and the frequency varied with $\omega = \omega_0 + \alpha(x - x_{start})$ (chirp) if the frequency at the back of the laser ($x = x_{start} - 9$) was over 2.4 the α parameter was deemed invalid.

The ramp final density as well as the length of the ramp were also free. Every generation had 200 individuals and the algorithm was run for 400 generations. In each simulation, Q is calculated at each time step and the largest value for instant Q represents the score attributed to the simulation.

In this run of the code the free parameters from Equation (3.5) were chosen to be $j = 1/2$ and $\sigma_{min} = 0.10$. The selected threshold energy was $E_t = 4 m_e c^2$.

Among all individuals in the generation, the one with the best Q was selected, and the results are

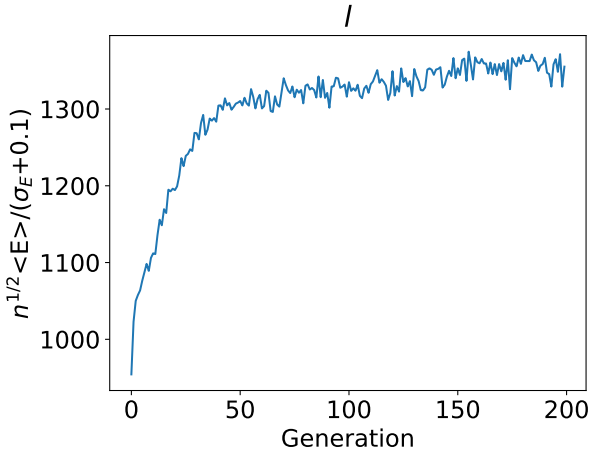
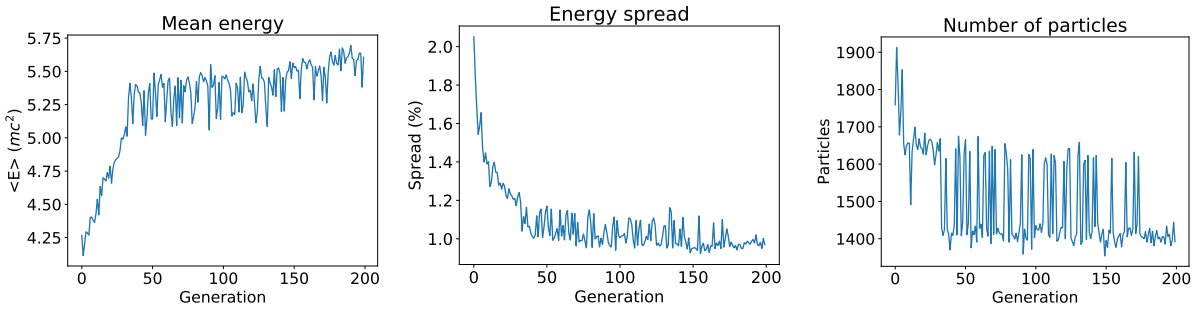


Figure 3.5: Evolution of $Q = n^{1/2}E/(\sigma_E + 0.1)$ (optimized quantity)



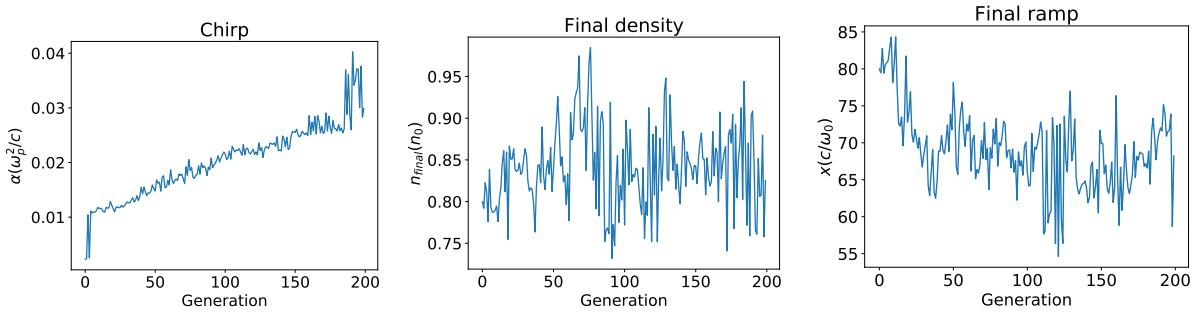
(a) Mean energy (b) Energy spread (c) Number of particles

Figure 3.6: Evolution of the properties of the electron beam at the same iteration the instant Q was maximal. Only the best result for instant Q in the whole generation is plotted.

shown in figure 3.5. The increase in Q was about 45% from the first (945) to the last (1370) generation, which shows the effectiveness of the genetic algorithm in this setup.

The first 40 generations were enough to see a great improvement in the Q-factor. From the three main parameters of the beam, 2 of the three were improved. The relative energy spread decreased from 2 to 1.2 % at the same time the mean energy of the beam went up from 4.2 to 5 $m_e c^2$. The number of particles went down from 1900 to 1650. However, having the parameter $j = 1/2$ means that the score due to the number of particles only went down by $(1650/1900)^{1/2} = 0.931$, which is much smaller than the improvement in the mean energy and energy spread. After 40 generations some oscillations are observed in all three parameters. This means that little changes in the inputs of the simulation lead to large changes in the parameters themselves, but these changes do not seriously affect the Q-parameter as it is defined.

The chirp factor went up consistently across generations (Figure 3.7a, which shows it is a very important factor to consider, as variations of the chirp may influence greatly the final electron beam. Considering the plasma density profile, both the final density of the ramp and the end position appear to have little significance to the end result - high Q-factors were obtained with end positions from 60 to 75 and with final densities from 0.75 to 0.95 (Figures 3.7c and 3.7b, respectively).



(a) Chirp parameter α ($\omega = \omega_0 + \alpha x$) (b) Final density of the ramp (c) Position of the end of the ramp

Figure 3.7: Evolution of the properties of the laser and plasma at the beginning of the simulation.

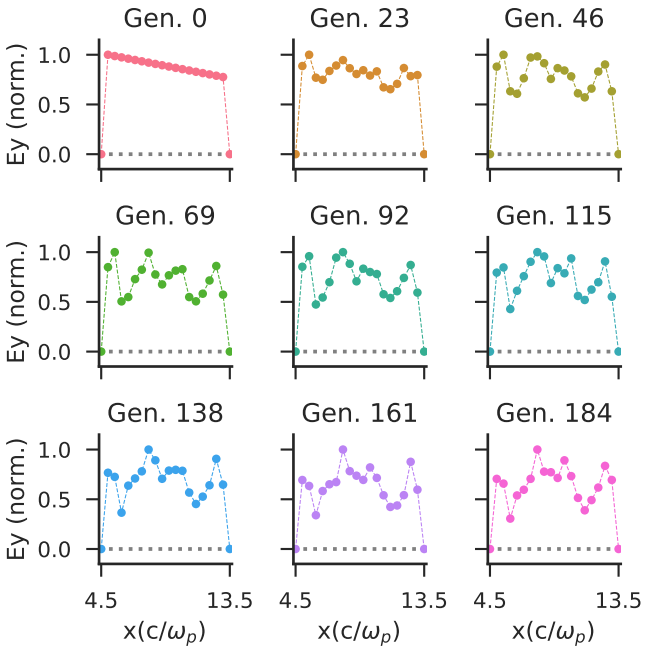


Figure 3.8: Evolution of the longitudinal profile

The longitudinal profile of the laser started as a straight line and changed to a laser with a higher peak electric field. As the generations progress, it becomes apparent that the internal laser structure (chirp and laser longitudinal profile) is the most important factor to consider when optimizing Q.

Chapter 4

Two-dimensional simulations: the role of the internal laser structure

In chapter 3, a 1D PIC code was used to predict the best conditions for LWFA under certain restrictions. However, as already stated, there are phenomena such as the formation of a bubble which cannot happen in a 1D code but it is essential to the acceleration mechanics.

To have results that can more consistently match those captured by experimental setups, the original 2D ZPIC code was also restructured and a genetic algorithm was made available.

4.1 High repetition lasers and the tailoring of the wavefront

Most LWFA experiments work with high powered lasers (1 Joule @ 100 TW) and have a repetition rate below 1 Hz. Many studies have been conducted on LWFA with higher repetition rates (kHz) with smaller laser energies (≈ 1mJ) because of the several advantages it conveys over the 1-J class - (i) larger mean electron current, (ii) higher level of stability, (iii) easier optimization via feedback control and (iv) fast averaging to increase the signal-to-noise ratio [25, 24, 26].

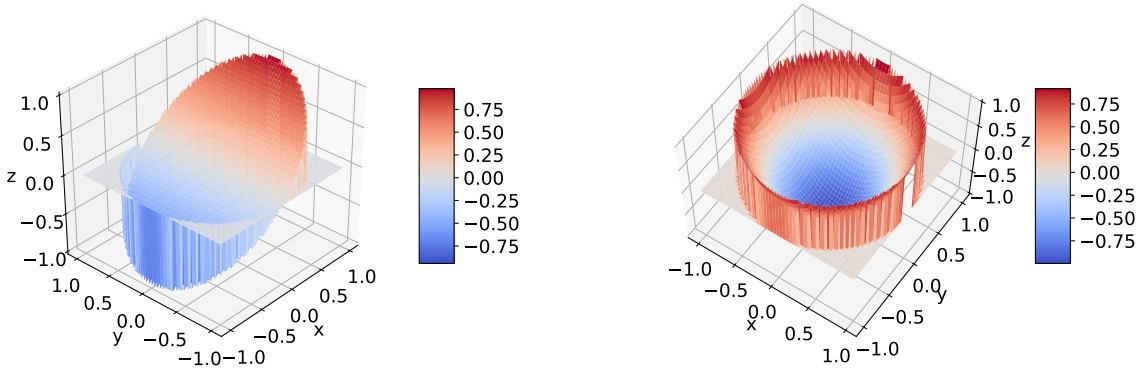
Since the technology has already been validated by multiple experiments, the effort at the experimental side is to optimize beam quality changing plasma or laser related variables, while trying to control the characteristics of the final beam. Some recent results [27] suggest that the tailoring of the wavefront could potentially be used to control the electron spatial distribution, as the distortion of the wavefront causes inhomogeneities in the laser intensity and consequently changes its focusing and defocusing properties.

4.2 Wave front shaping: the Zernike polynomials

The additional degree of freedom of 2D simulations brings a new range of possibilities that can be varied in the laser. In 2D, the laser pulse does not need to be a plane wave. Instead, the phase of the laser

No.	Polynomial
0	1
1	$\rho\cos(\theta)$
2	$\rho\sin(\theta)$
3	$\rho^2 - 1$
4	$2\rho^2\cos(2\theta)$

Table 4.1: The first five Zernike polynomials



(a) Third Zernike polynomial, $z = \rho\sin(\theta)$

(b) Fourth Zernike polynomial, $z = \rho^2 - 1$

Figure 4.1: Examples of Zernike polynomials

can have complex spatiotemporal profiles that can change the dynamics of the laser in the plasma. The basis of the optimization in 2D runs was the shaping of the wavefront.

The electric field of a perfectly Gaussian beam is given by equation (4.1).

$$E(r,z) = E_0 x \frac{\omega_0}{\omega(z)} \exp\left(\frac{-r^2}{\omega(z)^2}\right) \exp\left(-i\left(kz + l\frac{r^2}{2R(z)} - \phi(z)\right)\right) \tag{4.1}$$

The Gaussian phase inside the complex exponential in (4.1) can be changed with deformable mirrors [22]. The Zernike polynomials are the standard way to describe wavefront shaping, as they represent a sequence of orthogonal functions in the unit circle. They are better than other orthogonal sets as they are made of terms that have the same form as the aberrations often observed in optical tests[28]. The first five Zernike polynomials are shown in table 4.1. Both the Zernike polynomials no. 1 and no. 3 are shown in Figures 4.1a and 4.1b, respectively.

The Zernike polynomials are defined in 3D. Because in 2D there is only one transverse direction, the orthogonality condition is not verified. More so, the polynomials stop being linearly dependent, which modifies the method altogether (if the limit to the increment is 5%, two linearly dependent could potentially increase one variable up to 10%). To avoid this situation, in 2D the Zernike polynomials were replaced by the ρ^n set, being ρ the only transverse direction.

Thus, the final laser expression is given by

$$E(r,z) = E_0 x \frac{\omega_0}{\omega(z)} \exp\left(\frac{-r^2}{\omega(z)^2}\right) \exp\left(-i\left(kz + l\frac{r^2}{2R(z)} - \phi(z) + 2\pi/\lambda \sum_{n=0}^{k} a_n r^n\right)\right), \tag{4.2}$$

where the a_n set is given the name of the wavefront coefficients.

An important variable is the ratio k_{perp}/k_{long}. This ratio should be calculated at the apperture (a measure of transversal length, normally associated with the spot size of the laser). This ratio is defined as

$$\sum_{n=0}^{k} |a_n|((d/dr)r^n)_{r=\rho}/\omega_0 = \sum_{n=0}^{k} n|a_n|\rho^{n-1)}/\omega_0, \tag{4.3}$$

where ρ is the apperture of the laser pulse. This was given the name of maximum ratio.

4.3 Enhancing the acceleration efficiency

In a laser-plasma acceleration, energy flows from the laser to the plasma, and from the plasma to the accelerated beam. A key feature associated with the process is energy conversion efficiency: from the laser to the accelerated particle beam. To estimate the efficiency of the acceleration, the total energy of each beam is calculated, and the laser energy is kept constant for all simulations.

In the 1D case, plane waves were used, which means that there is no focusing or defocusing of the laser. For the 2D simulations, a focus has to be selected for the laser. In experiments [22, 27] normally a gas is ionized by the laser in the first place, creating ion-electron pairs (plasma) and only then LWFA takes place. In ZPIC however, the set-up is similar to the one described in Figure 3.4. To be able to compare different laser pulses, their energy was kept constant throughout the algorithm, as already done in section 3.3.

For all the simulations in 2D , the box size was (25,51.2), with (nx,ny)=(1250,256) with a particle per cell number of 8, and a maximum ratio at the aperture (Equation 4.3) of 0.1 except when it is specified.

4.3.1 Energy efficiency

The energy range 20 to 25 $m_e c^2$ was selected, with 200 individuals per generation and the best 50 being mutated with a total of 8 generations. The score of a simulation was given by the number of particles n_{par} whose energy was within the [20,25]$m_e c^2$ range.. The free variables were only associated with the wavefront coefficients (6). The plasma started at x=25.5 c/ω_0. The density was a simple step from n=2n_0 to n=n_0 at x=30.6. The first 4 Zernike polynomials were used. The length of the pulse was 4 and if the laser energy density is defined as

$$E_l = \sum_{i,j} (E[i,j]^2 + B[i,j]^2)\, dx\, dy, \tag{4.4}$$

where $E[i,j]$ / $B[i,j]$ represents the electric/magnetic field in the cell (i,j) and dx,dy the respective grid size. The laser energy was constant and equal to 500 $c^4 m_e^2/e^2$. The focus of the laser pulses was fixed at z=0 (before the plasma started).

The results from the genetic algorithm are shown in Figure 4.2. The improvement of n_{par} was of the order of 20% in relation to a Gaussian beam. However, if it is taken into account that the iteration at which both maximums were attained is different and the energy profile of electrons is plotted in the exact

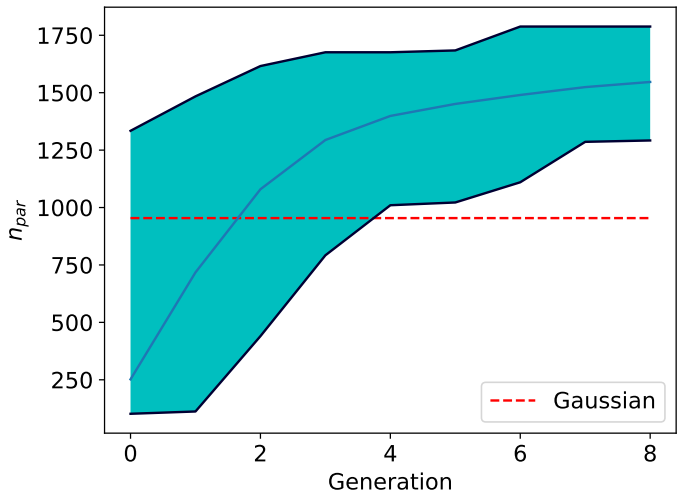


Figure 4.2: Evolution of the number of particles in the 20 to 25$m_e c^2$ range (n_{par}) during the simulation. Upper and lower bound of the colored region in black represent the maximum and minimum value of n_{par} in each generation. Mean value in the generation appears in dark blue. The value for a Gaussian pulse is shown in dashed red.

iteration where the optimized beam has its maximum, it is observed the profile in Figure 4.3. The reason for this comparison is to replicate what was done in experiments before (the detector is in a fixed spatial position). This plot was run with a larger particle per cell (ppc) number (36) than in the original run (8). This shows that an fourfold increase in that parameter did not change the overall results and therefore the simulations for the energy efficiency are not very ppc-sensitive.

These results are very promising, as the improvement in that range is of the order of 100%. It is seen that the optimized beam not only surpasses the Gaussian beam at the desired range but it is also better for the whole energy range of 12 to 30 $m_e c^2$. This reveals that the wavefront may be used as an independent variable to control the energy of the electron beam.

A full comparison with energy and angle for both lasers can be found in Figures 4.4a and 4.4b. The major difference between beams is the position of the maximum density of particles in energy, which is slightly higher in the Gaussian beam. Even though the optimized electron beam is more disperse, it is clear that the number of electrons in the desired energy range is greater in the optimized beam.

The electric fields of both Gaussian and optimized laser pulses can be found in Figures 4.5a and 4.5b, respectively. The global phase stayed fairly similar, but in the optimized laser one denotes a slight tilt counterclockwise. The transformation originated both from the r and r^3 polynomial.

4.3.2 Decreasing the acceleration distance

As explained in subsection 1.1, plasma accelerators could improve significantly the acceleration length when compared to conventional accelerators. This is calculated from the maximum electric fields a plasma can hold. However, the internal structure of the laser driver could be a new degree of freedom to further reduce the acceleration distance. In order to explore this possibility, the score of a simulation now corresponds to the first iteration at which the sum of all the energies of particles whose energy exceeds

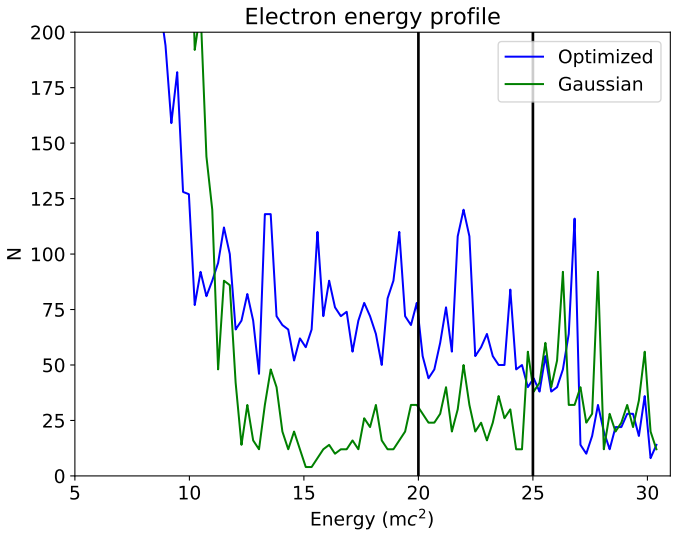
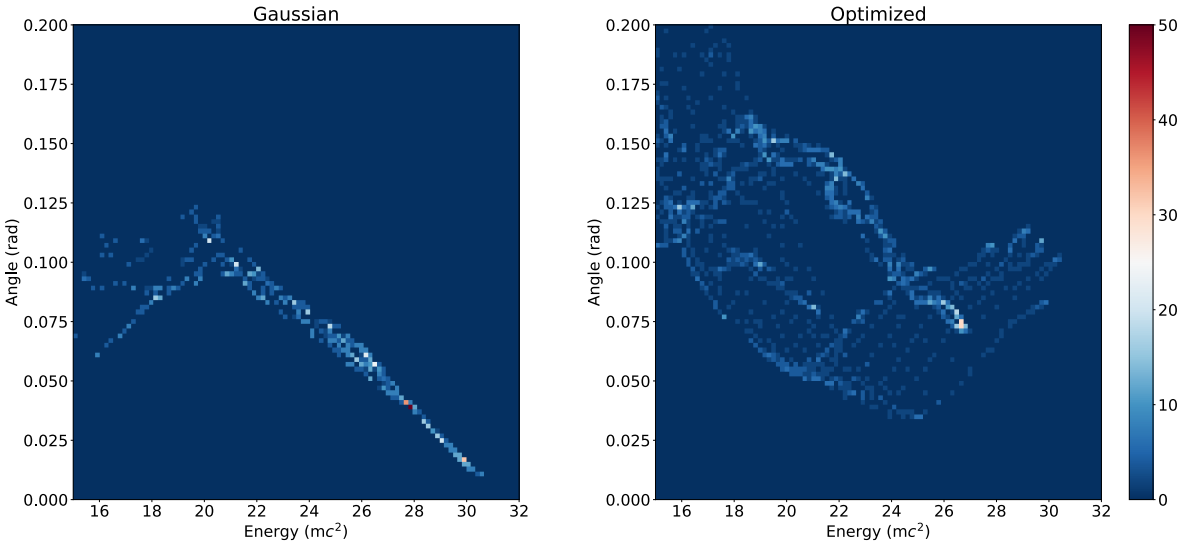


Figure 4.3: Electron energy profile for Gaussian (green) and optimized (blue) beams. The profile was done for the iteration 3502, where the optimized beam had the maximum number of particles in the energy range. The energy range is represented by the two vertical lines.



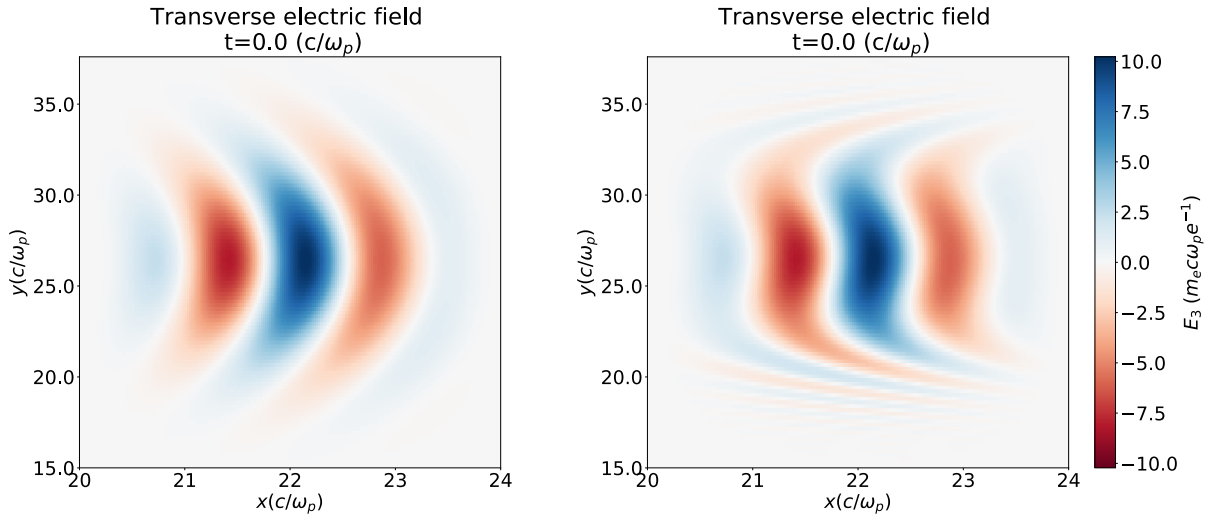
(a) Electron beam from Gaussian laser

(b) Optimized electron beam

Figure 4.4: Histogram (energy, angle) for particles in the [15,32] $m_e c^2$ energy range. At the left side, the result for the Gaussian pulse at the iteration where the number of electrons in the [20,25]$m_e c^2$ is maximum for the optimized beam. The angle is calculated via the momentum ratio p_{perp}/p_{long}, where p_{long} was defined as the momentum parallel to the laser propagation direction.

E_t is larger than a number. In this case the variable $E_t = 15 m_e c^2$ was selected and the total energy was 20000 $m_e c^2$. This ensures that the number of particles considered for the total energy is statistically significant.

In this case 600 individuals per generation and the best 50 being mutated with a total of 70 generations. The free variables were only associated with the wavefront coefficients (6). The plasma started at x=25.5 c/ω_0. The density was a simple step from n=2n$_0$ to n=n$_0$ at x=30.6. The first 6 Zernike polynomials were used. The length of the pulse was 4 and the laser energy was constant and equal to 500



(a) Gaussian laser

(b) Optimized laser

Figure 4.5: Transverse electric field at the initial time. a) The perfectly Gaussian laser. b) The optimized laser with 4 Zernike polynomials.

$c^4 m_e^2/e^2$. The focus of the laser pulses was fixed at x=22.5 c/ω_0(start of the plasma).

The results and comparison with the Gaussian beam are in Figure 4.6. It is relevant to point out that the number of generations of this optimization study far exceeded that of the other studies in 2D. This is due to the fact that the algorithm is actively trying to reduce the number of time iterations (the simulation runs until the threshold E_t is obtained) and therefore the computational time for each simulation is reduced.

The improvement shown in figure 4.6 is self-evident - the acceleration time (and therefore, distance) is reduced by just over 10% (11%) in comparison with the Gaussian beam. The red line represents the Gaussian phase whose global phase was optimized via the same algorithm, which makes the result more meaningful. It is not clear that this procedure was followed in [22]. Even though the optimization value is not impressive, saving 10% in plasma length purely by adjusting the laser beam may reduce costs in that same magnitude, increasing even further the gap between plasma and conventional accelerators.

The transverse electric field for both the Gaussian pulse and the optimized pulse are presented in Figures 4.7a and 4.7b. This pulse is dominated by the 1 and r^{2n} polynomials, as the global phase change is almost π and the pulse appears symmetrical in relation to the x axis.

4.4 Enhancing the radiation emission

The energy radiated by a particle is, for $p_\parallel \gg p_\perp$, given by [29]:

$$E_{rad} \propto p_\perp^2 \gamma \tag{4.5}$$

where γ is the Lorentz factor and $p_\perp$ is the transverse momentum. This is just an estimate of the radiated power. It is important to refer that no actual radiation is being calculated explicitly. To perform

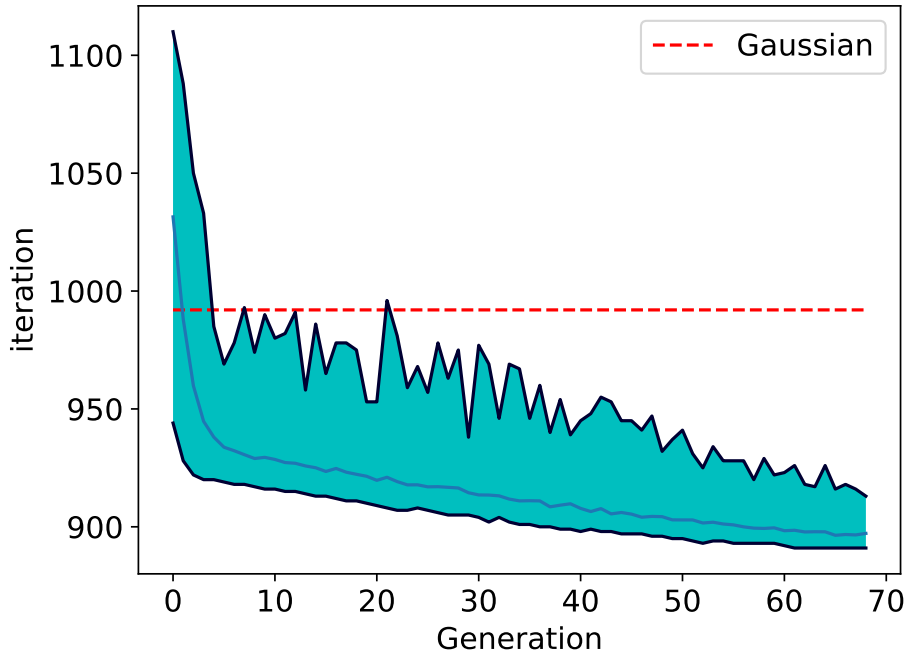
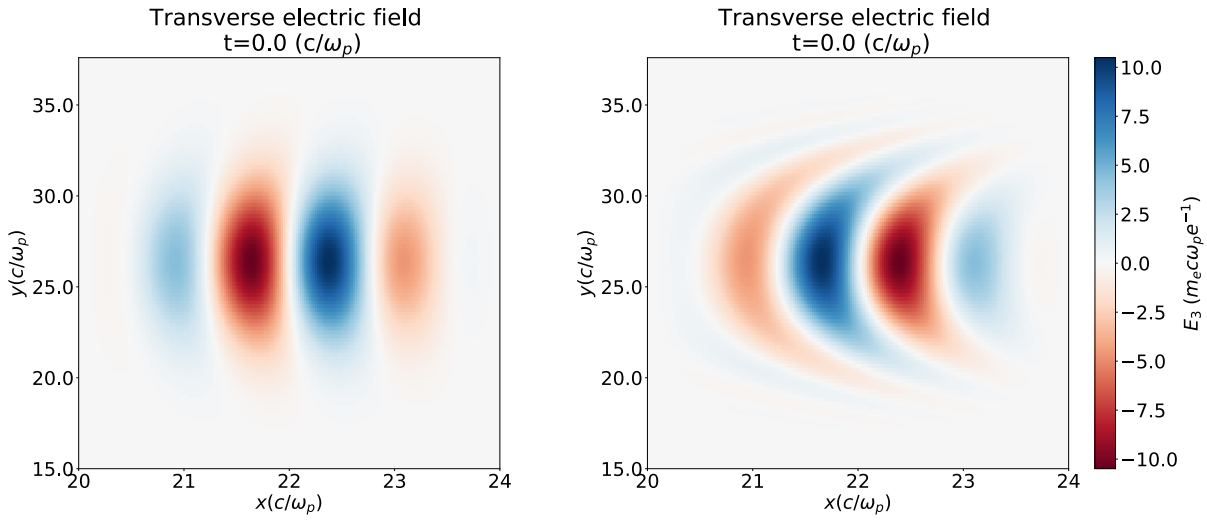


Figure 4.6: First iteration at which the sum of the energies of particles with $E > 15m_ec^2$ is bigger than 20000 m_ec^2. The upper and lower bound of the colored region represent the maximum and minimum values inside the generation. The mean inside the generation is shown in dark blue. The value for a Gaussian pulse is shown in dashed red.



(a) Gaussian laser

(b) Optimized laser

Figure 4.7: Transverse electric field at the initial time. a) The perfectly Gaussian laser. b) The optimized laser with 6 Zernike polynomials.

the optimization in this variable the cumulative radiation across a certain time frame should be calculated. The peak radiation, albeit relevant, is not measured by a detector, where only a time-averaged intensity is stored.

In this case, in order to compare beams, the best way should be to select an energetic threshold. To have a clustering of the beam could mean that some variation of the number of particles could take

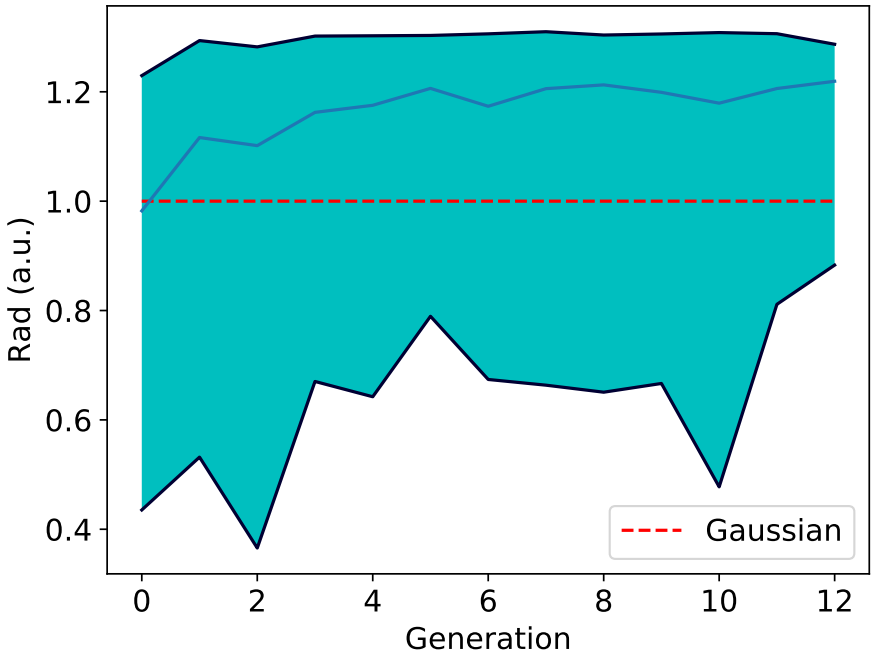
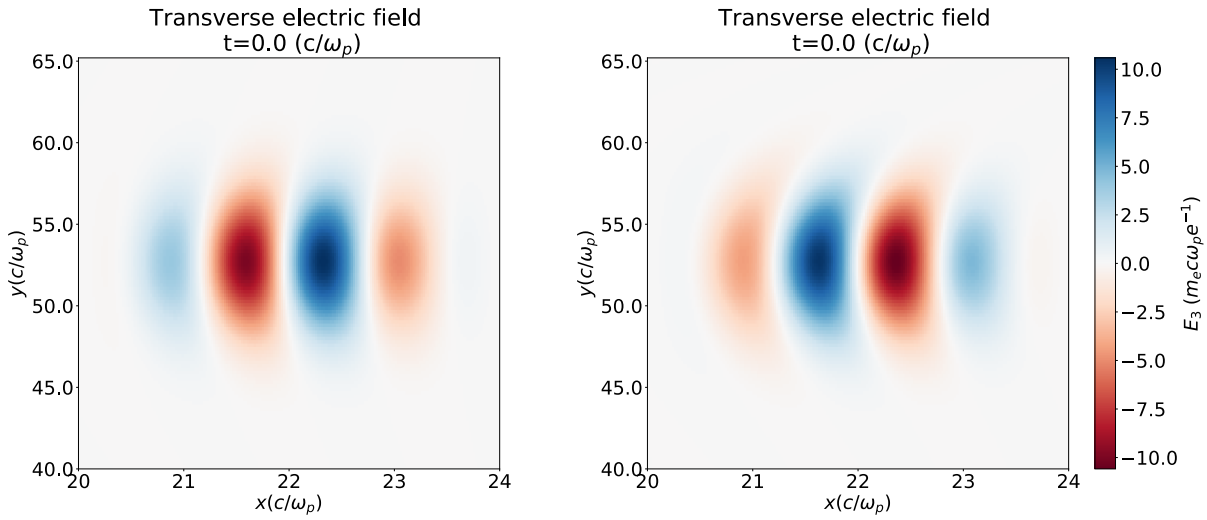


Figure 4.8: Estimated cumulative radiation originated from particles with energy $E > 15m_ec^2$.The upper and lower bound of the colored region represent the maximum and minimum values inside the generation. The mean inside the generation is shown in dark blue. The value for a Gaussian beam is shown in dashed red.

place and account for the majority of the discrepancy between beams.

In this particular example the threshold was selected at $E_t = 15m_ec^2$ and the radiation was integrated from $t = 0$ to $t = 900(1/\omega_p)$. If the energy threshold is selected at $E_t = 0m_ec^2$, the contribution of the particles of the background doesn't allow for much relative difference between lasers (less than 5%). In this case 200 individuals per generation and the best 50 being mutated with a total of 12 generations. The free variables were only associated with the wavefront coefficients (6). The plasma started at x=25.5 c/ω_0. The density was a simple step from n=2n$_0$ to n=n$_0$ at x=30.6. The first 6 Zernike polynomials were used. The length of the pulse was 4 c/ω_p and the laser energy was constant and equal to 500 $c^4m_e^2/e^2$. The focus of the laser pulses was fixed at x=22.5 c/ω_0(start of the plasma). The results of the optimization are presented in Figure 4.8.

The gain in estimated radiation in comparison to a Gaussian pulse is around 30% This shows that the amount of radiation produced is highly dependent on the internal structure of the beam. However, it is possible to note some differences looking at the electric fields at starting time (Figures 4.9a and 4.9b for the Gaussian and the optimized pulses, respectively). The first discrepancy is the overall global phase, that brings the maximum of the electric field forward. The optimized beam also appears to have a slight curvature coming primarily from the even polynomials, as the beam is symmetrical in relation to the x axis.



(a) Gaussian laser

(b) Optimized laser

Figure 4.9: Transverse electric field at the initial time. a) The perfectly Gaussian laser. b) The optimized laser with 6 Zernike polynomials.

Chapter 5

Conclusions

The main conclusion in this work is that genetic algorithms can be used to improve the electron beam quality in the lab, not only to try to make the laser distortion-free but to actively try to distort it in a specific way to tackle several optimization problems.

The k-means algorithm, a machine learning tool used in classification problems, was able to successfully isolate the accelerated electron bunch from the background electrons in several 2D configurations.

Áfter a successful benchmark in a known 1D case, the quality of the beam in 1D increased by over 30% across generations. In 2D, the best score for energy efficiency, acceleration distance and radiation production was always originated from lasers that were not perfectly Gaussian, which can open a new research area. In particular, the number of electrons in a determined energy range was increased by 20% when comparing the best score for both the optimized and Gaussian beam. If the comparison is done for the exact iteration where the optimized beam obtain the highest number of particles in that range, the difference reaches 100%. This corresponds to an experiment where a detector is put in a specific spatial location and beam properties are only measured once. When minimizing the acceleration distance, changing the laser wavefront allowed for a 10% reduction, something that no experimental set up has measured so far. The estimated radiation from an electron beam was increased by 30% when comparing to a Gaussian laser.

5.1 Contributions

Two posters for the EPS 2018, along with an article for the proceedings[30, 31]. Part of this work (namely subsection 4.3.1) was used for a presentation for an European Research Council (ERC) grant. A jupyter notebook for ZPIC was created to study the effect of the length of a flat laser on a plasma. This notebook was used in one lab rotation of the APPLAuSE doctoral programme.

5.2 Future Work

In the simulations that were presented here the value of the ratio k_{perp}/k_{long} was taken very conservatively so that the final laser pulse wouldn't be so much different from the Gaussian pulse. With the relaxation of this parameter and using the results collected thus far it is likely that some extra improvement may be had in the variables already discussed.

Other cases where it is expected that changes in the wave front may produce measurable improvements in the final electron beam include

- Transverse emittance
- Electron angular profile
- Radiation shape (getting a specific signature in the detector)
- Electron spatial profile (from laser pulses with orbital angular momentum)
- Point $(\vec{r}, t)$ to inject electrons with determined momentum

OSIRIS is the next step, where the optimization can move to a most sophisticated code. In OSIRIS 3D simulations will be carried for the exact same cases shown here and possibly some other. This will require a hardware upgrade from the Accelerates cluster here at IST or the use of other supercomputers so the simulation time is sustainable..

Other machine learning algorithms, like neural networks, could be used to optimize internal OSIRIS processes and further reduce computational time.

Bibliography

[1] N. A. Solyak. Gradient limitations in room temperature and superconducting acceleration structures. *AIP Conference Proceedings*, 1086(1):365–372, 2009. doi: 10.1063/1.3080933. URL http://aip.scitation.org/doi/abs/10.1063/1.3080933.

[2] T. Tajima and J. Dawson. Laser electron accelerator. *Physical Review Letters*, 43(4):267–270, 1979. doi: 10.1103/physrevlett.43.267.

[3] E. Esarey, C. B. Schroeder, and W. P. Leemans. Physics of laser-driven plasma-based electron accelerators. *Rev. Mod. Phys.*, 81:1229–1285, Aug 2009. doi: 10.1103/RevModPhys.81.1229. URL https://link.aps.org/doi/10.1103/RevModPhys.81.1229.

[4] P. Chen, J. M. Dawson, R. W. Huff, and T. Katsouleas. Acceleration of electrons by the interaction of a bunched electron beam with a plasma. *Phys. Rev. Lett.*, 54:693–696, Feb 1985. doi: 10.1103/PhysRevLett.54.693. URL https://link.aps.org/doi/10.1103/PhysRevLett.54.693.

[5] A. Caldwell, K. Lotov, A. Pukhov, and F. Simon. Proton-driven plasma-wakefield acceleration. *Nature Physics*, 5(5):363–367, 2009.

[6] J. Rosenzweig, B. Breizman, T. Katsouleas, and J. Su. Acceleration and focusing of electrons in two-dimensional nonlinear plasma wake fields. *Physical Review A*, 44(10):R6189, 1991.

[7] W. Lu, C. Huang, M. Zhou, W. Mori, and T. Katsouleas. Nonlinear theory for relativistic plasma wakefields in the blowout regime. *Physical review letters*, 96(16):165002, 2006.

[8] C. Geddes, C. Toth, J. Van Tilborg, E. Esarey, C. Schroeder, D. Bruhwiler, C. Nieter, J. Cary, and W. Leemans. High-quality electron beams from a laser wakefield accelerator using plasma-channel guiding. *Nature*, 431(7008):538–541, 2004.

[9] K. Yee. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *IEEE Transactions on Antennas and Propagation*, 14(3):302–307, May 1966. ISSN 0018-926X. doi: 10.1109/TAP.1966.1138693.

[10] W. P. Leemans, A. J. Gonsalves, H.-S. Mao, K. Nakamura, C. Benedetti, C. B. Schroeder, C. 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. *Phys. Rev. Lett.*, 113:245002, Dec 2014. doi: 10.1103/PhysRevLett.113.245002. URL https://link.aps.org/doi/10.1103/PhysRevLett.113.245002.

[11] I. Blumenfeld, C. E. Clayton, F.-J. Decker, M. J. Hogan, C. Huang, R. Ischebeck, R. Iverson, C. Joshi, T. Katsouleas, N. Kirby, et al. Energy doubling of 42 gev electrons in a metre-scale plasma wakefield accelerator. *Nature*, 445(7129):741–744, 2007.

[12] M. Litos, E. Adli, W. An, C. Clarke, C. Clayton, S. Corde, J. Delahaye, R. England, A. Fisher, J. Frederico, et al. High-efficiency acceleration of an electron beam in a plasma wakefield accelerator. *Nature*, 515(7525):92–95, 2014.

[13] P. A. W. et al. Horizon 2020 eupraxia design study. *Journal of Physics: Conference Series*, 874(1): 012029, 2017. URL http://stacks.iop.org/1742-6596/874/i=1/a=012029.

[14] D. Zibar, H. Wymeersch, and I. Lyubomirsky. Machine learning under the spotlight. *Nature Photonics*, 11(12):749, 2017.

[15] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 609–616, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. URL http://doi.acm.org/10.1145/1553374.1553453.

[16] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[17] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676): 354, 2017.

[18] J. Zhang, Z. h. Zhan, Y. Lin, N. Chen, Y. j. Gong, J. h. Zhong, H. S. H. Chung, Y. Li, and Y. h. Shi. Evolutionary computation meets machine learning: A survey. *IEEE Computational Intelligence Magazine*, 6(4):68–75, Nov 2011. ISSN 1556-603X. doi: 10.1109/MCI.2011.942584.

[19] D. S. Weile and E. Michielssen. Genetic algorithm optimization applied to electromagnetics: a review. *IEEE Transactions on Antennas and Propagation*, 45(3):343–353, Mar 1997. ISSN 0018-926X. doi: 10.1109/8.558650.

[20] A. Mihailescu. A new approach to theoretical investigations of high harmonics generation by means of fs laser interaction with overdense plasma layers. combining particle-in-cell simulations with machine learning. *Journal of Instrumentation*, 11(12):C12004, 2016. URL http://stacks.iop.org/1748-0221/11/i=12/a=C12004.

[21] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85 – 117, 2015. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2014.09.003. URL http://www.sciencedirect.com/science/article/pii/S0893608014002135.

[22] Z.-H. He, B. Hou, V. Lebailly, J. Nees, K. Krushelnick, and A. Thomas. Coherent control of plasma dynamics. *Nature communications*, 6, 2015.

[23] A. Martinez de la Ossa, Z. Hu, M. J. V. Streeter, T. J. Mehrling, O. Kononenko, B. Sheeran, and J. Osterhoff. Optimizing density down-ramp injection for beam-driven plasma wakefield accelerators. *Phys. Rev. Accel. Beams*, 20:091301, Sep 2017. doi: 10.1103/PhysRevAccelBeams.20.091301. URL https://link.aps.org/doi/10.1103/PhysRevAccelBeams.20.091301.

[24] D. Guénot, D. Gustas, A. Vernier, B. Beaurepaire, F. Böhle, M. Bocoum, M. Lozano, A. Jullien, R. Lopez-Martens, A. Lifschitz, and J. Faure. Relativistic electron beams driven by kHz single-cycle light pulses. *Nature Photonics*, 11(5):293–296, 2017. ISSN 17494893. doi: 10.1038/nphoton.2017.46. URL http://dx.doi.org/10.1038/nphoton.2017.46.

[25] D. Gustas, D. Guénot, A. Vernier, S. Dutt, F. Böhle, R. Lopez-Martens, A. Lifschitz, and J. Faure. High-charge relativistic electron bunches from a kHz laser-plasma accelerator. *Physical Review Accelerators and Beams*, 21(1):1–7, 2018. ISSN 24699888. doi: 10.1103/PhysRevAccelBeams.21.013401.

[26] B. Beaurepaire, A. Lifschitz, and J. Faure. Electron acceleration in sub-relativistic wakefields driven by few-cycle laser pulses. *New Journal of Physics*, 16, 2014. ISSN 13672630. doi: 10.1088/1367-2630/16/2/023023.

[27] B. Beaurepaire, A. Vernier, M. Bocoum, F. Buehle, A. Jullien, J. P. Rousseau, T. Lefrou, D. Douillet, G. Iaquaniello, R. Lopez-Martens, A. Lifschitz, and J. Faure. Effect of the laser wave front in a laser-plasma accelerator. *Physical Review X*, 5(3):1–7, 2015. ISSN 21603308. doi: 10.1103/PhysRevX.5.031012.

[28] R. R. Shannon, J. C. Wyant, and R. Kingslake. *Applied optics and optical engineering / edited by Rudolf Kingslake*. Academic Press New York, 1965. ISBN 0124086098 0124086101.

[29] W. Lu, M. Tzoufras, C. Joshi, F. S. Tsung, W. B. Mori, J. Vieira, R. A. Fonseca, and L. O. Silva. Generating multi-GeV electron bunches using single stage laser wakefield acceleration in a 3D nonlinear regime. *Physical Review Special Topics - Accelerators and Beams*, 10(6):1–12, 2007. ISSN 10984402. doi: 10.1103/PhysRevSTAB.10.061301.

[30] B.Malaca, J.Vieira, and R. Fonseca. Machine learning controlled laser wakefield acceleration simulations. In *45th EPS Conference on Plasma Physics*, 2018.

[31] R. Calado, B. Malaca, M. Pardal, A. Helm, W. B. Mori, V. K. Decyk, J. Vieira, L. O. Silva, and R. A. Fonseca. The zpic educational code suite. In *45th EPS Conference on Plasma Physics*, 2018.

Appendix A

Classes Developed

Some new classes were developed in this work so that the study of the internal properties of the beam could be done.

A.1 1D classes

A.1.1 Laser with fixed energy

Before this work, the a_0 of the laser was the main variable that could be used in the input deck. A simple laser beam could be placed in the simulation as shown in the listing below

```
t_emf_laser laser = {
        .type = GAUSSIAN,
        .start = 24.0,
        .fwhm  = 4.0,
        .a0 = 3.0,
        .omega0 = 4.0,
        .W0 = 6.0,
        .focus = 30.0,
        .axis = 25.6,
        .polarization = M_PI_2
};

sim_add_laser( sim, &laser );
```

Because the energy of the laser was the variable that was required to remain constant, the class `t_emf_full_envelope` was created. This class allows for selection of a chirp, an envelope and the energy.

```
 t_emf_full_envelope env = {
        .start=29.5,              //start of the beam (in plasma coordinates)
```

```
        .length=9,                  //maximum length of the beam
        .N=e_Np,                    //number of points in the envelope
        .interpolation=SPLINE,      //can be spline or linear
        .omega0=__OMEGA__,          //laser frequency at the start
        .chirp_N=__N_CHIRP__,       //number of chirp parameters (w=w0+c0*z+c1*z^2+...)
        .freq_limit=__FREQ_LIMIT__, //if the frequency is bigger at
                                    //any point the new parameters will be rejected
        .chirp_values=chirp,        //pointer with c0,c1,...
        .energy=50,                 //laser energy
        .values=values2,            //values of the longitudinal profile (between 0 and 1)
        .polarization=M_PI_2        //laser polarization
};
```

A.2 2D

A.2.1 Laser with changed wave front

Before, only gaussian laser pulses were available on ZPIC. In 2D the biggest change was the creation of the `t_emf_laser_zernike` class, which allows for wave front tailoring.

```
t_emf_laser_zernike laser = {
        .type = GAUSSIAN,
        .start = 24.0,
        .fwhm  = 2.0,
        .a0 = 1.0,
        .omega0 = 4.0,
        .energy = 500.0,
        .W0 = 4.0,
        .focus = 20.0,
        .axis = 25.6,
        .polarization = M_PI_2,
        .aperture=4*0.5, //aperture (where you calculate the ratio)
        .N_zernike=6, //number of coefficients
        .zernike_coef=values2, //vector with 6 coefficients
        .ratio=0.1 //maximum initial ratio calculated at the apperture
};
```