Simulation of plume-plasma expansion with one-dimensional Particle-in-Cell

C A Gonzalez¹, J A Arteaga¹, Y H Gomez², J Osorio², J A Jaramillo² and H Riascos¹
¹ Grupo Plasma Láser y Aplicaciones, Universidad Tecnológica de Pereira A.A 097, Pereira, Colombia
² Grupo Sirius, Universidad Tecnológica de Pereira A.A 097, Pereira, Colombia
E-mail: caangonzalez@utp.edu.co

Abstract. In this work we present the analysis of the dynamic of the expansion of Al Plasma produced by Nd:YAG laser (1064 nm, 500 mJ, 9 ms, 10 Hz) in vacuum. To study the Coulomb interaction between the particles of the initial states of the plasma expansion, we use the one dimensional Particle-in-Cell method (PIC) and finite difference method. We considered an ideal model, that is, we assume that the plasma is in a local thermal equilibrium, the ablated particles have a fixed temperature and a constant evaporation flux (J) from the aluminium surface. To obtain more accurate results we use high computing exploiting the parallelization of this kind of algorithms. The mean velocity and particles densities are determined for different times of the expansion.

1. Introduction

Studies of plasma produced by pulsed laser have been taking great importance due to its multiple applications in different scientific and industrial fields. One of its most important applications is the deposition of thin films by laser ablation commonly called Pulsed Laser Deposition (PLD) [1, 2, 3, 4].

PLD involves the incidence of high-energy laser in a solid that absorbs its radiation. This energy is transmitted to the ejected species in the form of kinetic energy [5, 6]; Ablation process is generated when the kinetic energy absorbed by the species is greater than its binding energy in the material. The result of this process is a cloud of particles composed of ions, electrons and neutrals, called plasma plume. In this cloud there is a collective interaction between the ablated particles through the fields generated from these particles. The representation of this interaction is a complex and nonlinear problem, which includes the use of a wide range of physical concepts. Research about pulsed laser deposition is made through theoretical and experimental models.

Many important details of the plasma are highly difficult to obtain experimentally. Therefore the numerical simulation plays an important role in understanding the phenomena that occurs there. In this way the theoretical models can be checked and additionally, get more detailed information about what happened in the process.

In this paper we used the PIC computational method for studying the one-dimensional initial stage of the plasma plume expansion in vacuum. It was accomplished by analyzing the change of kinetic momentum produced by the Coulomb interaction between the plasma particles. To
make the results more approximate to those predicted by analytical models it is required to define a finer discrete model. However, the use of fine discrete models increases exponentially the algorithms simulation times. It is for this reason that is required to use a strategy of high performance computing, where we can take advantage of the latest parallelization platforms such as GPUs.

Here we present the acceleration of ions and electrons of Al plume plasma in vacuum as function of interaction time between laser pulse and the target using PIC method and high performance, assuming Coulomb interaction for species in plasma only. It was determined that some instabilities were generated while the expansion plume occurred.

2. Model

The study of the expanding plasma plume in vacuum consists of two stages. An initial stage where the particles are ejected from the solid surface, during the interaction between laser pulse and the solid in a finite time $t_0$. And then, a second stage, where no additional particles are ejected allowing for the Coulomb interaction between particles in the cloud.

Particles emitted in the initial stage are principally composed of ions (charge +e, mass M) and electrons (charge -e, mass m). These particles were ejected in the normal direction to the surface ($x$ - axis) through a constant evaporation flux $J$. The generated cloud of ions and electrons as a whole has a neutral electric charge, because the evaporation flows of both particles are assumed to be equal and constant. In addition, these particles are ejected from the surface with a thermal energy $kT_0$ that depends on both the material and the characteristics of the laser, in our case we assumed that they reach thermal equilibrium.

In the second stage it is assumed that the particles are ejected with initial velocities that satisfy a half-space Maxwell distribution function

$$f_i(x = 0, v_x, t) = J \left( \frac{M}{kT_0} \right) \exp \left( -\frac{Mv_x^2}{2kT_0} \right)$$

where $J = n_0v_0$ is the flow of ions or electrons immediately after the incidence of the laser, $k$ is Boltzmann’s constant and $v_x$ is the velocity perpendicular to the surface. Immediately after the expulsion of particles, the expansion process of the cloud begins due to the Coulomb interaction between particles. This interaction is based on the potential difference generated by the charge distribution in certain regions of the cloud described by the equation:

$$\nabla^2 \phi = -e \frac{n_i - n_e}{\epsilon_0}$$

where $n_{i,e}$ represents the density of particles of both species. The fundamental basis for describing the evolution of collisionless plasma is given by the Vlasov equation:

$$\partial_t f + v_x \partial_x f + \frac{eE}{M} \partial_{v_x} f = 0.$$ (3)

This equation determines the time evolution of a statistical system described by a distribution function $f$. In this way it achieves the kinetic model of particles in phase space $(x,v)$. From (3) it is possible to obtain different conservation laws, i.e. the conservation of the particle numbers (zero order momentum) [7].

3. Simulation

3.1. PIC description

The PIC method is a technique that allows the tracking of particles in a space. In order to develop a computational model, a discretization of the space is needed as well as the definition
of the initial values of the system. Later, the forces are calculated by using those initial values, and the next position of each particle in the discretized space. Using next state we can close the loop and make the calculations during the time in order to complete the path of the movement of each particle.

First, a charge is assigned to the group of particles in a cell, called 'superparticle'. Since the charge is not uniformly distributed, a potential difference can be generated between the neighboring cells that is assumed to be a field of zero order weight\[8, 9\], that is, the n-th cell particles are affected from the difference of charge between both the n − 1 and n + 1 cell, no matter the real position of the center of the charge of the n − th cell as it is shown in figure 1. Once the charge density at each cell is known, the Poisson equation is solved to find the potential difference between the cell and its neighbors, subsequently the electric field due to the potential difference given by the equation: \( E_n = -\nabla \phi_n \) is evaluated.

By knowing the n-th cell’s electric field, its effect on each particle of that cell is weighted. Then, their new position and velocity are found, according to Newton’s second law, for each discretized time step \( \Delta t \). The summary of the PICs method is shown in figure 2.

### 3.2. Discrete space

Since the equations describing the system are differential equations, the most appropriate method to solve them in the discrete space is the one-dimensional finite difference method \[10, 11\], this is,

\[
\frac{\partial^2 \phi}{\partial x^2} \approx \frac{\phi_{n+1} - 2\phi_n + 2\phi_{n-1}}{\Delta x^2}
\]

where \( n \) represents the \( n \)-th cell. To integrate the equations of motion we use the leap – frog algorithm which creates a coherent calculation cycle, given by the following formulas \[12\]:

\[
x_{n+1} = x_n + v_{n+1/2} \Delta t
\]

\[
v_{n+3/2} = v_{n+1/2} + \frac{eE_n}{M} \Delta t
\]

### 3.3. Stability criteria

To apply the PIC method in the expansion of the plume, the simulation space must be divided into cells smaller than the Debye length \( \Delta x \ll x_0 \), this in order to neglect the collisions among particles \[13\].

The time \( t_0 \) is the time that the particles emitted take to go through a Debye length with an ejection speed \( v_0 \), \( (t_0 = x_0/v_0) \). According to these conditions \( \Delta t \) should be less than the
plasma wave period (\(\Delta t \ll 2\pi /\omega_e\)) to satisfy the Courant criteria where \(\omega_e\) is the electron plasma frequency.

For dimensionless physical quantities, all the equations that govern the system were normalized using \(x_0\), \(t_0\), ie, \(\Delta x/x_0\), \(\Delta t/t_0\). For the distribution function, the kinetic energy of each particle is normalized with the average energy \(kT_0\). Finally, the normalized field takes the form \(E/(v_0 M/\varepsilon t_0)\).

3.4. Simulation details
Since all particles in the system get an initial velocity greater than zero, the boundary condition \(\phi(x = 0) = 0\) is considered in order to ensure that all the particles that return to the surface are reflected [14]. When the simulation space \(x_0\) is overflowed by the particles \((x_p > x_0)\), these are re-injected at, \(x_p = x_0 - x_p\) in order to satisfy the conservation of the charge.

As a consequence of the previous conditions, the electric potential must be periodic \(\phi(x) = \phi(x + x_0)\). According to previous authors [15], the approximate parameters to simulate Al plasma produced by Nd: YAG laser (1064nm, 500mJ, 9ms, 10Hz) are presented in table 1. The initial velocity for the ions given by \(v_{0,i} = \sqrt{2kT_0/\pi M}\); in our case determined a temperature \(T_0 = 10000 K\) and for electrons \(v_{0,e} \approx 7v_{0,i}\).

3.5. Algorithm optimization
In order to improve the performance of the algorithm, a data dependence analysis between functions must be made. In this case, the force particle function and electric field grid function are independent of each other, thus allowing for the parallel execution of both processes in different CPU cores. That is why the high performance tools, based on floating point operations, known as GPUs, were used in the present work. GPUs act as coprocessors to the CPU’s data bus using a high-transfer [16]. In order to further reduce simulation times and take the best advantage of the GPU, the use of OpenCL framework from the KRONOS consortium [17] was used. This framework allows the creation of calculus kernels using a programming language like C, C++, or FORTRAN and it’s also costless.

| Table 1. Simulation parameters for Al plasma. |
|-----------------------------------------------|
| \(N_p\) | \(N_g\) | \(\Delta x/x_0\) | \(\Delta t/t_0\) |
| 1x10^5  | 5x10^3  | 5x10^{-3} | 1x10^{-5} |

**Figure 2.** Basic cycle of the Particle In Cell simulation.
Parallelized and sequential algorithm was run on a computer with the following characteristics: GTX580 Video Card that has 512 processing cores, 16 GB of RAM Intel Core i7 processor, 2 solid state drives and a motherboard P6T7 Axus SuperComputer.

Figure 3 shows the linear behavior of both processes where the GPU’s efficiency is evidenced, having an improvement of 16x. Also, the execution time is doubled at each increment of emission times shown for the two algorithms.

4. Results and discussions
To investigate the dynamics of expansion of the plasma plume, we construct the phase space of the system, which consists of identifying each of the particles according to its velocity \(v_p\) and position \(x_p\) in an instant of time \(t\). Figure 4 shows the phase space \((x_p,v_p)\) system for the emission time \(t_0\), where the black dots indicate the presence of an ion, and the blue ones the presence of an electron. From figure 4 it is observed that the electrons achieve greater distance and speed compared with ions, this due to the ratio of their masses \((m \ll M)\). In the region where both species coincide in position, it shows clearly the oscillation of the electrons around the ions, the electrons are trapped in the potential that they generate. After this region, a certain number of electrons that exceed their kinetic energy and thus manage to move freely \((mv_e = cte)\) can be found. As for the dynamics of ions it is obvious the linear collective behavior, that can be known thoroughly studying its density. To find the particle density we turn to the solution of the expression, \(n(x,t) = \int f(x,v_x,t)dv_x\) divided into intervals of equal length, the simulation space; in our case we took ten points for each \(x_0\) \((x/x_0 = 10)\) as shown in Equation(1), and

![Figure 3](image-url)

**Figure 3.** Execution times for the sequential and parallelized algorithms.

![Figure 4](image-url)

**Figure 4.** Ion phase-space (black) and electron phase space (blue).
finally, we integrated the velocity of each particle that was found at each interval. Figure 5 shows that for an emission time \( t/t_0 = 1 \), the ion density decreases exponentially for \( x > 0.5x_0 \), just as the two fluid model predicts. As long as the post-emission time \( t/t_0 > 1 \) increases, density is distributed homogeneously in a larger space and, therefore showing the plume expansion. On the other hand, the electron density in a small region coincides with the ion density, according with the quasi-neutral plasma condition plasma.

The evolution of the mean velocity of ions \( u(x, t) = \frac{1}{n(x, t)} \int v_x f(x, v_x, t) dv_x \) as these were far from the surface was found, as shown in figure 6, by the same method that the figure 5 was done, with the distribution function \( f(x, v_x, t) \) we have resolved the equation.

Figure 6 shows the ions acceleration as they move away from the surface, this acceleration may be caused by the potential difference between these ions. As shown in figure 5, the ion density decreases exponentially down to zero, after this, the ions do not experience a change in their momentum. Therefore, the distance to the surface grows linearly with the momentum they have.

Unfortunately our results do not agree with those of other authors [15, 18], probably due to the fact that the previous results are related to certain factors, such as initial conditions, boundary conditions, and other factors inherent to the computational model. On the other hand, the consideration of the field weight in our case was zero-order, which has major implications on the final results, since the potential of a cell affects not only the neighbor cells but also the entire

Figure 5. Ion density vs ablated surface distance at different times.

Figure 6. Ion mean velocity vs. distance to wall for different times of the expansion.
system itself.

5. Conclusions

We have calculated the density for ions and electrons of a plasma plume in expansion finding that these densities decrease exponentially down to zero and then the movement was not change. While their mean velocities increase linearly. For an evaporation time larger than $t_0$ two stream instabilities were generated. An important aspect of this work was to have made the acceleration of the code using high performance computing with the use of graphic accelerator cards, as the results were obtained in a short time in comparison to the simulation of a sequential code. This acceleration enabled us to make changes to our source code (initial conditions) without having to wait long.

References

[1] Chrisey D and Hubler G 1994 Pulsed laser deposition of thin films Wiley-interscience publication (J. Wiley) ISBN 9780471592181
[2] Eason R 2007 Pulsed laser deposition of thin films: applications-led growth of functional materials (Wiley-Interscience) ISBN 9780471447092
[3] Riascos H, Neidhardt J, Randnóczi G, Emmerlich J, Zambrano G, Hultman L and Prieto P 2006 Thin Solid Films 497 1–6
[4] Franco L, Pérez J and Riascos H 2008 Microelectronics Journal 39 1363–1365 ISSN 0026-2692
[5] GBekefi 1976 Principle of Laser Plasmas
[6] Miller J and Haglund R 1998 Laser ablation and desorption Experimental methods in the physical sciences (Academic Press) ISBN 9780124759756
[7] Kruer W 2003 The physics of laser plasma interactions Frontiers in Physics (Westview Press) ISBN 9780813340838
[8] Hockney R and Eastwood J 1981 Computer simulation using particles (McGraw-Hill International Book Co.) ISBN 9780070291089
[9] Birdsall C and Langdon A 1991 Plasma physics via computer simulation Adam Hilger series on plasma physics (Adam Hilger) ISBN 9780750301176
[10] Henrici P 1962 Discrete variable methods in ordinary differential equations (Wiley)
[11] Appert K, Suzuki S, Vaclavik J and Villard L 1987 Computer Physics Reports 6 335–349 ISSN 0167-7977
[12] Cartwright K, Verboncoeur J and Birdsall C 2000 Journal of Computational Physics 162 483–513
[13] Chen F 1984 Introduction to Plasma Physics and Controlled Fusion: Plasma physics Introduction to Plasma Physics and Controlled Fusion (Plenum Press) ISBN 9780306413322
[14] Felske H, Schneider R and Weisse A 2008 Computational many-particle physics Lecture notes in physics (Springer) ISBN 9783540456850
[15] Nedelea T and Urbassek H M 2002 Phys. Plasmas 9 3209–16
[16] Macedonia M 2003 IEEE Computer 36 106–8
[17] Munshi A, Gaster B, Mattson T, Fung J and Ginsburg D 2011 OpenCL Programming Guide OpenGL Series (Addison Wesley Professional) ISBN 9780321749642
[18] Ellegaard O, Nedelea T, Schou J and Urbassek H 2002 Applied Surface Science 197–198 229–238 ISSN 0169-4332