Simulations of Plasmas with Electrostatic PIC Models Using the Finite Element Method

A. C. J. Paes, N. M. Abe, V. A. Serrão, and A. Passaro
Instituto de Estudos Avançados, Centro Técnico Aeroespacial
12231-970, São José dos Campos, São Paulo, Brazil

Received on 20 June, 2002. Revised version received on 10 November, 2002

Particle-in-cell (PIC) methods allow the study of plasma behavior by computing the trajectories of finite-size particles under the action of external and self-consistent electric and magnetic fields defined in a grid of points. In this work, the Finite Element Method (FEM) is used in order to obtain the self-consistent fields. An electrostatic PIC-FEM computational code for simulation of one-dimensional (1D) and two-dimensional (2D) plasmas was developed based on two available and independent codes: the first one a 1D PIC code that uses the Finite Difference Method and the other a FEM code developed at the Instituto de Estudos Avancados (IEAv). The Poisson equation is solved and periodic boundary conditions are used. The ion background that neutralizes the total plasma charge is kept fixed and uniformly distributed in the domain of study. The code is tested by studying the fluctuations of the plasma in thermal equilibrium. In thermal equilibrium a plasma sustain fluctuations of various collective modes of electrostatic oscillations, whose spectral distribution can be analytically obtained by using the fluctuation-dissipation theorem and the Kramers-Kronig relation. In both 1D and 2D cases, there are excellent agreement between the spectral distribution curves predicted theoretically and those obtained by simulation for finite size particles and long wavelengths.

I Introduction

In this work, we use the particle-in-cell (PIC) model, originated from the work of J.M. Dawson [1, 2, 3] in the late fifties, for the simulation of plasmas. In this model, the plasma is represented by thousands of particles (actually macroparticles, since each particle used in the simulation represents thousands of particles of a real experiment). From the positions and velocities of these particles at a certain instant, we calculate the current and charge densities using one of various schemes of particle distribution to a grid of points. The grid spacing Δx must be sufficiently small to resolve the collective behavior of plasmas (typically Δx of the order of the Debye length λ_{De}).

Using these charge and current densities, we calculate the self-consistent electric and magnetic fields via Maxwell's equations. Then, with these fields plus the externally applied ones, we obtain the new particle positions and velocities. Finally, we proceed along this basic cycle with a Δt sufficiently small to resolve the highest frequency of the problem (typically the plasma frequency ω_{pe}).

In the PIC code, Maxwell's equations are usually solved by the Finite Difference Method (FDM) or by Fourier techniques (specially when periodic boundary conditions are involved). In the FDM, the fields are defined in a grid of points, the derivatives are replaced by differences in the values of the field in adjacent grid points, and difference equations on the grid substitute the differential field equations.

Another approach is to apply the Finite Element Method (FEM) instead of FDM to solve the Maxwell's equations. In FEM, the system is divided into subsystems of simple geometry (for example, triangles and/or rectangles for a two-dimensional program), which are called finite elements. Inside each element, the values of the fields are calculated by means of interpolation functions, for example polynomials. The shape of the interpolation in the elements is defined by field values, and sometimes its derivatives, at the nodal points. The field derivatives are obtained by deriving the interpolating functions, and the field equations are approximated by minimizing integral (integro-differential) equations obtained by variational principles or by applying the weighted residual method to the differential equations and boundary conditions.

The FEM has advantages in treating elliptic and parabolic equations when compared with the FDM. The association of boundary conditions to nodal points, mesh refinement, and the adoption of different mesh types (mixed-mesh) become easy in the finite element method. Moreover, the finite element method has advantages of optimization and accuracy [4]. A short description of FEM is presented in section II.

In this work, we evaluate the performance of a computational program that couples the FEM and PIC methods by calculating fluctuations in a Maxwellian plasma, and we deduce theoretical expressions for the spectral distribution of the electric field energy considering finite-size particles. We developed the computational program based on two independent codes. The first one is a 1D PIC modeling software developed by Kruer [5]. This code uses the Finite Difference Method (FDM) in order to compute the self-consistent field. The second one is a 1D and 2D FEM code fully developed at the Instituto de Estudos Avancados (IEAv). In section II, we present the numerical method used. In section III, we present some results of the simulations in the 1D and 2D cases, and deduce expressions that allow us to compare the simulation results with the theoretical ones. Finally, in section IV we present some conclusions.

II Numerical model

In particle simulation, we attempt to study the behavior of plasmas by following the motion of a great number of particles. Since it is too expensive to compute the self-consistent motion of the plasma by calculating the force on each particle due to every pair interaction, a spatial grid is introduced on which the charge and current density are accumulated; electric and magnetic fields are stored in the grid as well.

In this work, we adopt an electrostatic model, i.e., only electric fields are considered. The basic scheme of this model is very simple. First, we accumulate the charge density on the grid from particle positions and velocities. Second, we solve Poisson's equation on the grid to find the electric potential and the electric field. Finally, we interpolate the electric field on the grid to the particles position and use the interpolated force on Newton's equations of motion to determine new particle positions and velocities. Then, we repeat this cycle as many time steps as necessary to study our system.

Numerous algorithms are available for the charge accumulation process in the spatial grid, such as NGP (nearest grid point) and PIC (particle-in-cell) [6]. In this work, we are using the PIC modeling. In this method, a point charge at x_i is assigned to its nearest grid points by using a linear interpolation $W(x)$. Particle pushing is done with the second-order accurate leap-frog technique.

The electrostatic field generated by a certain charge distribution can be obtained from the solution of Poisson's equation:

$$\nabla^2 \, \Phi(\vec{r}) \; = \; \rho(\vec{r}) \, , \tag{1}$$

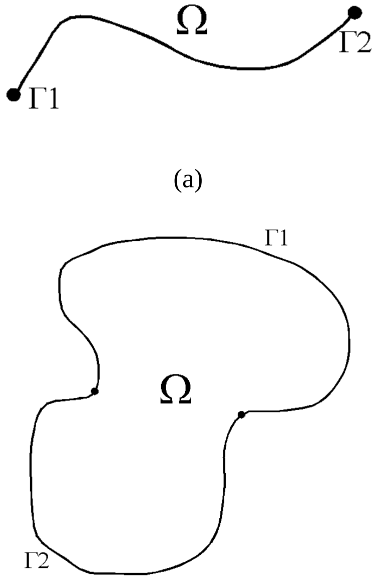
with boundary conditions:

$$\Phi \; = \; \phi_s \, , \qquad \text{in } \Gamma_1, \tag{2}$$

and

$$\frac{\partial \Phi}{\partial n} = f_s \qquad \text{in } \Gamma_2 \tag{3}$$

where $\Gamma \;=\; \Gamma_1 + \Gamma_2$ is the surface that delimits the domain of study Ω. Fig. 1 shows the domain and delimiting surfaces for both a one-dimensional (1D) and a two-dimensional (2D) domain.



(a)

(b)

Figure 1. Limiting surfaces Γ_1 and Γ_2 for (a) a one-dimensional domain and (b) a two-dimensional domain.

An approximated solution ϕ for the differential equation (1) can be obtained by applying the Weighted Residual Method (WRM). A complete description of this method can be obtained in several basic literature on the FEM [7, 8]. In WRM, the integral of the residue I over the entire domain Ω is imposed equal to zero:

$$I = \int\limits_{\Omega} P_1 R_1 d\Omega + \int\limits_{\Gamma_1} P_2 R_2 dS + \int\limits_{\Gamma_2} P_3 R_3 dS = 0 \tag{4}$$

with:

$$R_1 = \nabla^2 \, \phi(\vec{r}) \; - \; \rho(\vec{r}) \tag{5}$$

$$R_2 = \phi \; - \; \phi_s \tag{6}$$

$$R_3 = \frac{\partial \phi}{\partial n} - f_s . \tag{7}$$

The time independent weighting functions, P_1, P_2, and P_3, are chosen arbitrarily. Only one condition must be imposed on these weighting functions: they must be sufficiently regular in order to obtain finite integrals.

By applying the first Green's identity to equation 4, results:

$$\int\limits_{\Omega} \vec{\nabla} P_1 \vec{\nabla} \phi \; d\Omega - \int\limits_{\Omega} P_1 \rho \; d\Omega + \int\limits_{\Gamma_1} P_1 \vec{n} \cdot \vec{\nabla} \phi \; dS + \int\limits_{\Gamma_2} P_1 \vec{n} \cdot \vec{\nabla} \phi \; dS +$$

$$\int\limits_{\Gamma_1} P_2\left(\phi-\phi_s\right) dS+\int\limits_{\Gamma_2} P_3\left(f_s-\frac{\partial \phi}{\partial n}\right) dS=0. \tag{8}$$

The fifth integral can be eliminated by imposing that the solution satisfies identically the Dirichlet boundary condition in Γ_1. As the weighting functions are arbitrary, P_1 is chosen equal to zero in Γ_1, resulting:

$$\int\limits_{\Omega} \overrightarrow{\nabla} P_1 \overrightarrow{\nabla} \phi \, d\Omega-\int\limits_{\Omega} P_1 \rho \, d\Omega+\int\limits_{\Gamma_2} P_3 f_s dS+\int\limits_{\Gamma_2}\left(P_1-P_3\right) \frac{\partial \phi}{\partial n} dS=0. \tag{9}$$

Without loss of generality, P_3 is chosen equal to P_1, in order to eliminate the fourth integral in (9). Additionally, in this work the Neumann condition in Γ_2 is always homogeneous, e.g., $f_s=0$ (there is no electric current applied in the domain of study).

The resulting integral equation:

$$\int\limits_{\Omega} \overrightarrow{\nabla} P_1 \overrightarrow{\nabla} \phi \, d\Omega-\int\limits_{\Omega} P_1 \rho \, d\Omega=0 \tag{10}$$

represents the weak formulation for the Poisson equation [7, 8]. We apply the FEM in order to evaluate Eq.(10) .

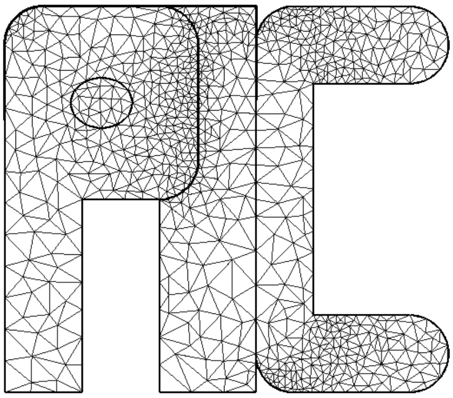


Figure 2. A triangular finite element mesh for an arbitrary domain.

In the FEM, the domain Ω is subdivided into elements of simple geometry, such as, lines and curves in 1D simulations, triangles and rectangles for 2D simulations, tetrahedra, parallelepipeds, and prisms for three-dimensional (3D) simulations. Each element defines a subdomain named Ω^γ. The Fig. 2 illustrates a triangular mesh for an arbitrary domain of study. These elements of simple geometry, named finite elements, must obey the following rules in order to constitute a consistent mesh:

- the sum of the area of each element results in the domain area $\left(\bigcup\limits_{\gamma=1 \ldots NE} \Omega^\gamma=\Omega\right)$; and

- the elements do not intersect each other $\left(\bigcap\limits_{\gamma=1 \ldots NE} \Omega^\gamma=0\right)$. NE is the number of elements in which the domain is decomposed.

The finite element mesh follows all boundaries of the geometric model even for a complex geometry. Due to this characteristic, the attribution of boundary conditions to the mesh points is easier than in the FDM, because it is not necessary to treat mesh points that do not coincide with the geometric lines.

Inside each element, the approximated value of the dependent variable is calculated from a finite set of linearly independent base functions (compact support). This set depends on the family and approximation order of the used elements. Fig. 3 shows two examples of the triangular elements family and the base function set for each case. The functions L$_1$, L$_2$, and L$_3$ are given by:

$$L_i=\frac{a_i+b_i x+c_i y}{2 \Delta}$$

where Δ is the area of the triangle and the coefficients a, b, and c are obtained from the coordinates of the vertices of the triangle:

$$\begin{aligned} a_i&=\varepsilon_{ijk}\left(x_j y_k-x_k y_j\right) \\ b_i&=\varepsilon_{ijk}\left(y_j-y_k\right) \\ c_i&=\varepsilon_{ijk}\left(x_k-x_j\right) \end{aligned}$$

ε_{ijk} is the three-dimensional Levi-Civitta symbol [9]. The indices i, j, and k vary from 1 to 3.

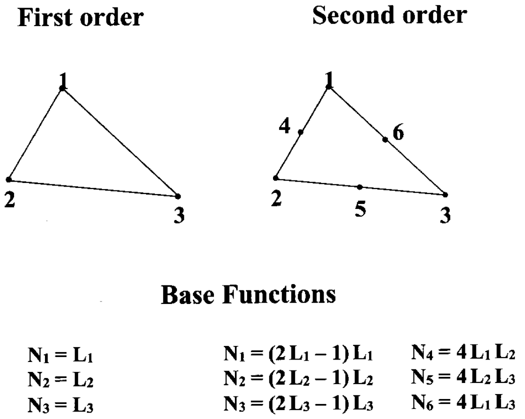


Figure 3. Triangular element family – elements of first and second order.

The derivation of these expressions for triangular elements and for elements of other families is presented in [7].

In general, the shape of the interpolation functions in the element is defined by the values of the dependent variables and, sometimes, their derivatives, in some points of the element, the nodal points. The dependent variables, and, when necessary, their derivatives, are named nodal variables. Thus, one or more nodal variables are computed in each nodal point. The number of nodal variables depends on the formulation adopted for the solution of a given problem. In the formulation adopted in this work, there is only one nodal variable: the electric potential.

In this way, the scalar potential in a element γ is written as a sum of the base functions $N_i^{(\gamma)}$:

$$\phi^{(\gamma)}(\vec{r}_j) \;=\; N_i^{(\gamma)}(\vec{r}_j)\phi_i^{(\gamma)}, i=1,....n_0 \qquad (11)$$

where $\phi_i^{(\gamma)}$ are the potential values on the nodal point i and n_0 is the number of nodal points of the element. Notice that we are using the summation convention. The base functions have the following properties on the nodal points:

$$N_i^{(\gamma)}(\vec{r}_j) \;=\; \delta_{ij} \qquad (12)$$

where δ_{ij} is the Kronecker's delta. This means that the base functions assume the value one when the index of the nodal point, j, coincides with the index of the base function, i, and is zero in any other nodal point. The base functions for a determined element are valid only in the limits of this element.

In the FEM, we also let the integro-differential equation (10) be equal to zero for each element, which allows us to write them individually:

$$\int_{\Omega^\gamma} \overrightarrow{\nabla} P_1^{(\gamma)}.\overrightarrow{\nabla}\phi^{(\gamma)}\, d\Omega - \int_{\Omega^\gamma} P_1^{(\gamma)}\rho^{(\gamma)}\, d\Omega \;=0 \qquad (13)$$

where Ω^γ corresponds to the element domain.

In order to solve the residual equation, we have only to choose convenient weighting functions. In the finite elements approximation, inside each element the number of unknowns is equal to that of nodal variables. Consequently, we must choose the same number of weighting functions. The weighting functions inside each element coincide with the base functions used to describe the dependent variables (Galerkin's method):

$$P_i^{(\gamma)} \;=\; N_i^{(\gamma)} \qquad (14)$$

resulting for the residue equation in each element:

$$\left(\int_{\Omega^\gamma} \overrightarrow{\nabla} N_i^{(\gamma)}.\overrightarrow{\nabla} N_j^{(\gamma)}\, d\Omega\right)\phi_j - \int_{\Omega^\gamma} N_i^{(\gamma)}\rho^{(\gamma)}\, d\Omega \;=0 \qquad (15)$$

with i = 1,2,..., n_0.

A matricial notation can be used to represent the integral equation set (15):

$$[K]\{\phi\}^T \;=\; \{b\}^T \qquad (16)$$

where:

$$[K] = \int_{\Omega^\gamma} [gradN]^T\, [gradN]\; d\Omega \qquad (17)$$

$$\{\phi\} = \{\;\phi_1 \quad \phi_2 \quad \phi_3 \quad \cdots \quad \phi_{n_0}\;\} \qquad (18)$$

$$\{b\} = \int_{\Omega^\gamma} \{N\}\,\rho^{(\gamma)}\, d\Omega \qquad (19)$$

$$\{N\} = \{\;N_1 \quad N_2 \quad N_3 \quad \cdots \quad N_{n_0}\;\} \qquad (20)$$

$$[gradN] = \begin{bmatrix} \frac{\partial\{N\}}{\partial x_1} \\ \frac{\partial\{N\}}{\partial x_2} \\ \cdots \end{bmatrix}, \qquad (21)$$

and x_i stands for the ith coordinate in a k-dimensional space.

For example, for the one-dimensional case, we use a linear element in first order approximation (two nodal points, which correspond to the vertices of the line element), with linear interpolation functions given by:

$$N_1(x) \;=\; \frac{x_{1_2} - x}{l} \qquad (22)$$

$$N_2(x) \;=\; \frac{x - x_{1_1}}{l} \qquad (23)$$

in which

x_{1_1} : is the coordinate of the left nodal point,

x_{1_2} : is the coordinate of the right nodal point,

l : is the length of the element,

x : is the coordinate inside the element,

and the following matrices for the element:

$$[grad\,N] \;=\; [\frac{\partial\{N\}}{\partial x}] \;=\; \frac{1}{l}\,[-1\;\;1], \qquad (24)$$

$$[K] \;=\; \frac{1}{l^2}\int \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} d\,\Omega \;=\; \frac{1}{l}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \qquad (25)$$

$$\{b\} \;=\; \int \{N\}\rho\, d\Omega \;=\; \frac{l\rho}{2}\,[1\;\;1]\,. \qquad (26)$$

After the calculation of the matrices of each element, we assemble the global system, i.e., the matricial system including all the nodal points. Because to each nodal point is associated a unique identification number in the system, the process of assemblage is simple. Each finite element matrix connects the set of nodal points of the element, and hence the values calculated for an element of the matrix can be added directly in the global matrix in the positions indicated by the number of the nodal points. An elegant mathematical treatment of the global matrix assemblage is presented in [8]. The global matricial system is given by:

$$[K_G]\{\phi_G\}^T \;=\; \{b_G\}^T \qquad (27)$$

where $[K_G]$ is a symmetric and sparse matrix, of order NP x NP (NP is the total number of nodal points of the grid). The vectors $\{\phi_G\}$ and $\{b_G\}$ have dimension NP.

The electric potential on the grid is obtained by solving (27). In this work, we adopted the ICCG (Incomplete Cholesky Conjugate Gradient) method [10] to solve the system, because the matrix $[K_G]$ is sparse, real, and positive definite.

III Results

In this section, we show the results of simulations obtained with the developed electrostatic PIC-FEM (1D and 2D) numerical code. We applied the code to the study of fluctuations in a stable plasma.

Even in a Maxwellian plasma, there are fluctuations. Plasma waves are Cerenkov-emitted and Landau-damped. The balance between emission and absorption leads to thermal-level field fluctuations.

In spite of Vlasov theory and particle models simulate collisionless plasmas, electric field fluctuations do not emerge from a Vlasov model because these fluctuations are connected in a basic way with the discreteness of the plasma. Consequently, particle models have to be used to calculate such fluctuations.

In order to obtain the electric field energy fluctuation spectrum theoretically, the test-particle picture is used [11]. In this model, it is considered the motion of a test charge in the Vlasov fluid (plasma described by the Vlasov equation). After some time, the test particle polarizes the plasma, that is, the particle get "dressed". This means that in a plasma, the Coulomb potential (potential of interaction between particles in vacuum) is modified because of collective effects, each negative charge attracts a cloud of positive charges and vice versa. From this potential and the corresponding density fluctuation, the spectral distribution E_k^2 of the electric field can be obtained (k stands for k-space, or Fourier space).

At present, experiments are being conducted to observe the process of particle "dressing" in a non-equilibrium plasma. This process has a predicted time scale of the order of $(\omega_{pe}/2\pi)^{-1}$, which corresponds to 70 femtoseconds for a plasma with density of 2 10^{18} cm^{-3}. The precision of the existing models can be verified by using pulsed lasers lasting 27 femtoseconds to probe the plasma [12, 13]. However, for a Maxwellian plasma there is a well established result for the spectral distribution of the averaged longitudinal electric field energy given by [14]:

$$\left\langle \frac{E_k^2 L}{8\pi} \right\rangle = \frac{T_e}{2} \, \frac{1}{1 + k^2\lambda_{De}^2} \tag{28}$$

for a system of length L and transverse area equal to one. T_e is the electron temperature in energy units.

This result can also be analytically obtained using the fluctuation-dissipation theorem and the Kramers-Kronig's relation [4].

To compare this result deduced for a test charge (point electron) with the results of our simulations, we must remember that we are simulating the physics of a plasma of finite-size macroparticles. The finite-size effects are unavoidably associated with the interpolation function used to relate quantities to the grid. The macroparticle effect is due to the fact that each simulation particle represents thousands of electrons in a real experiment.

It can be shown that in k-space the charge density for a finite-size particle simply equals the charge density for a point particle system multiplied by a shape factor $S(k)$. Thus, we can rewrite most of the plasma theory for the finite size particle system by replacing the charge q by $qS(k)$ [4].

However, here, there is an additional difficulty since there are really two sources for the "finite-sizeness" of the particles. One is the above mentioned use of an interpolation function $W(x)$ to relate particle quantities to the grid. This can never be avoided in a PIC code. A second possible source is the use of various kinds of filter functions $S(x)$, either in real space or k-space. The effective particle shape is then the convolution of these two functions $\int W(x)S(x-x')dx$, or in k-space, $W(k)S(k)$ [15].

The considerations above imply that we have to modify the term $k^2\lambda_{De}^2$ in Eq. (28) in order to take into account finite-size effects. But macroparticle effects also lead to alteration of Eq.(28). This modification comes from the fact that the temperature used in the formula must not be that of the electron, because in the statistical basis of Eq. (28), what matters is the freedom of kinetic motion. In the simulation, the free random motion is not assigned to an electron of energy $\frac{1}{2}m_e v_e^2$ (where m_e is the electron mass, and v_e its velocity), but to a macroscopic particle of energy $\frac{1}{2}m_p v_p^2$ (where m_p is the macroparticle mass and v_p its velocity). Therefore, we should interpret T_e as $T_p = < \frac{1}{2}\, m_p v_p^2 >$, with T_p the macroparticle temperature (in units of energy).

Based on all the above considerations, the theoretical expression for the spectral distribution of the averaged longitudinal electric field energy for a plasma of finite-size macroparticle can be written:

$$\left\langle \frac{E_k^2 L}{8\pi} \right\rangle = \frac{T_p}{2} \left(1 + \frac{k^2\lambda_{De}^2}{S^2(k)W^2(k)} \right)^{-1} . \tag{29}$$

In this work, we compare the results of our simulations with this expression.

We know all terms of Eq. (29) but one. The unknown term is $S(k)$ (since $W(k)$ is obtained from $W(x)$, the prescribed interpolation function), that is, the filtering term.

In order to determine this filter, let us consider a simple problem, that is, the normalized one-dimensional Gauss' law $\nabla_x E_x = \rho$. In the theoretical case, applying a Fourier transform to this equation we obtain

$$E_{xk} = \frac{\rho_k}{ik_x} \tag{30}$$

To determine $S(k)$, let us look at what happens in particle models that use Fourier transform methods to calculate the electric field, because in this case they use a prescribed $S(k)$ given by $e^{-\frac{k^2a^2}{2}}$(with a of the order of λ_{De}) filtering out the short-wavelength ($ka > 1$) modes, lowering the effective collision frequency, and thus allowing realistic simulations with fewer particles. In this case they have for the one-dimensional Poisson equation:

$$E_{xk} = \frac{\rho_k S(k)}{ik} = \frac{\rho_k e^{-\frac{k^2a^2}{2}}}{ik} \tag{31}$$

with the wavenumber k=k_x.

In our case, applying the finite element formulation for a Fourier mode, we have from the one-dimensional Poisson's equation

$$E_{xk} = \frac{\Delta x \; sin(k\Delta x)\rho_k}{i4sin(\frac{k\Delta x}{2})}. \tag{32}$$

Comparing this expression with Eqs. (30) and (31) permits us to determine $S(k)$ for our case, which is given by (for the normalized case $\Delta x = 1$):

$$S(k) = \frac{sin(k)\,k}{4sin(\frac{k}{2})} \tag{33}$$

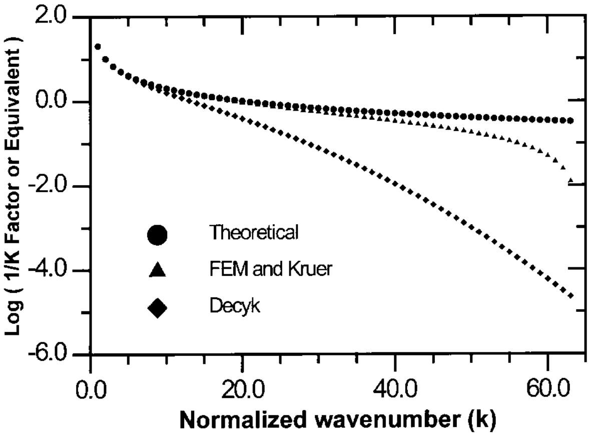


Figure 4. The filtering effect of the factor that multiplies ρ_k (Eq.32) for the finite element method (similar to the finite difference case [5]) and for the Fourier transform method [16], compared to the theoretical result.

In Fig. 4, we present the terms multiplying ρ_k (in fact the log_{10} of these terms) in the one-dimensional Poisson's equation for the theoretical model, for the finite element method (similar in this case to the finite difference case referred to as Kruer's case [5]), and for the Fourier transform method (referred to as Decyk's case [16]). The figure shows the filtering effect mentioned above, since the terms associated with FEM and Fourier transform decrease with k when compared with the theoretical result. Actually, the deduced $S(k)$, Eq. (33), acts as an unintentionally applied filter.

In the 1D simulations, we considered periodic boundary conditions with immobile ions as charge neutralizing background. With these boundary conditions, the electrons that leave the system from one side are placed at the other side. We used a grid of 50 points, N = 5000 particles, grid spacing $\Delta x = \lambda_{De}$, thermal velocity $v_t = \lambda_{De}\omega_{pe}$, time step $\Delta t = 0.2\omega_{pe}^{-1}$, and we run the code until $100\omega_{pe}^{-1}$.

In the 1D case, we used in the PIC-FEM code a linear interpolation function, whose Fourier transform is given by $W(k) = (\sin{(k/2)} \,/\, (k/2))^2$. Introducing this value of $W(k)$ and $S(k)$ from Eq. (33) into expression (29), normalizing, and rearranging the terms, we obtain:

$$c_k = \frac{E(k)^2 N}{v_{tx}^2} = \left(1 + \frac{\lambda_{De}^2 k^2}{(\frac{sink}{2})^2(\frac{sin(\frac{k}{2})}{\frac{k}{2}})^2}\right)^{-1} \tag{34}$$

where v_{tx}^2 is the normalized square of the thermal velocity of the particles.

The time average of the values $E(k)^2$ are obtained directly from the computational simulation, and these values can be compared with the values given by the expression (34).

In Fig. 5, we present the results obtained from the simulation (actually, log_{10} of the results) as a function of k, and the theoretical curve $log_{10}c_k$. We observe an excellent agreement, except for values of k near zero (very long wavelengths). In this region, there is some deviation from the thermal equilibrium. Presumably, this is due to the fact that modes with longer wavelengths are less susceptible to attenuation, and for this reason, they do not reach the equilibrium as fast as modes with smaller wavelengths. Besides, we note that in the simulation the time average is taken over a finite time interval, differently of what occurs in theory.

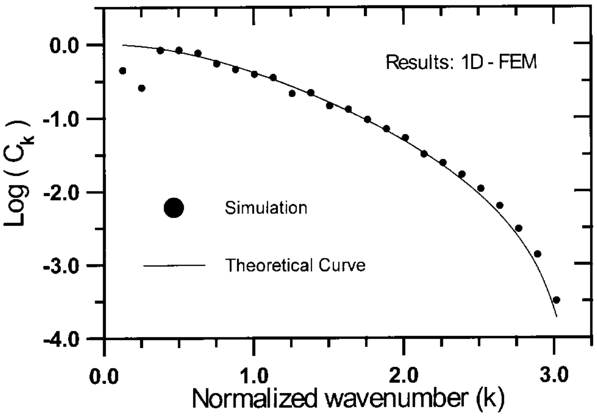


Figure 5. The theoretical spectral distribution of the electric field energy (solid curve) compared to the 1D PIC-FEM simulation (dots).

In the 2D simulations, we also considered periodic boundary conditions with immobile ions as a charge neutralizing background. Equally, in this case, electrons that leave from the sides of the rectangular grid are placed in the other side. We used a grid of 64 points in the x-direction and 128 points in the y-direction. We used $N = 294,916$ particles, grid spacings $\Delta x = \Delta y = \lambda_{De}$, thermal velocities of $v_t = \lambda_{De}\omega_{pe}$ in both directions, $\Delta t = 0.2\omega_{pe}^{-1}$, and we run the code until $325\omega_{pe}^{-1}$.

In the PIC-FEM code, we used linear interpolation functions in the x and y directions. In this case, the expression from the electric field average energy of the fluctuations depends on k_x and k_y and is given by

$$c\,(k_x,k_y) = \frac{(E_x^2(k_x,k_y) + (E_y^2(k_x,k_y))N}{(v_{tx}^2 + v_{ty}^2)} =$$

$$\left(1 + \frac{4(sin^2(\frac{k_x}{2}) + sin^2(\frac{k_y}{2}))^2\lambda_{De}^2}{(sin^2(k_x) + sin^2(k_y))W_1^2(k_x)W_2^2(k_y)}\right)^{-1} \quad (35)$$

where $W_1(k) = W_2(k) = (\sin{(k/2)} \,/\, (k/2))^2$.

In the simulation we used a constant value for k_y (normalized) equal to $2\pi\,(61/128)$, a value arbitrarily chosen among the possible values of k_y.

In Fig. 6, we present the results of the PIC-FEM code for the 2D simulations adopting a constant k_y. There is a good agreement with the theoretical results. The same observations made in the 1D case, for the region of k_x near zero remain valid. We also note greater deviation from the theoretical curve, than in 1D case, fact also observed with Fourier transform particle models [17].

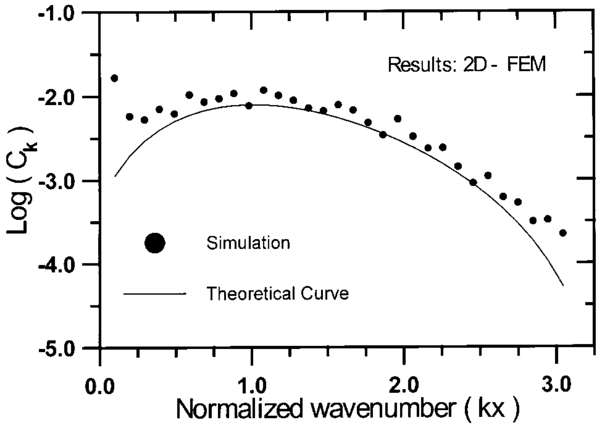


Figure 6. The theoretical spectral distribution of the electric field energy (solid curve) compared to the 2D PIC-FEM simulation (dots).

IV Summary

In this work, a 1D and 2D electrostatic computational code was developed in order to study problems that involve bounded collisionless plasmas. The code is based on the coupling of two numerical techniques, the FEM and the PIC modelings. The FEM is largely used in engineering and, in this work, it substitutes the FDM, traditionally applied in plasma physics. One of the advantages of the FEM is the handling of arbitrary boundaries, which can be accurately represented by an unstructured mesh. This feature allows the simulation of devices that present a complex geometry. The boundary condition handling advantages were not explored in this work.

The computations were done considering a plasma in thermal equilibrium and periodic boundary conditions in a rectangular domain, i.e., the plasma is in an open domain.

The computer model is tested by calculating the fluctuation spectrum of plasma in equilibrium. To make this comparison, we developed also a theoretical expression for the fluctuation spectrum of plasmas of finite-size macroscopic particles. We obtained an excellent agreement between the results of our simulations and the expressions developed.

Acknowledgements

This work was supported by FAPESP (grant n. 99/12468-8). The authors acknowledge Dr. V. Decyk for fruitful discussions.

References

[1] J.M. Dawson and A.T. Lin, *Particle Simulation*, Handbook of Plasma Physics - V.2 -(Elsevier Science Publishers 1984)

[2] J.M. Dawson, Rev. of Mod. Phys. **55**, 2 (1983)

[3] R. Hockney and J. Eastwood, *Computer Simulation Using Particles* (McGraw-Hill, New York, 1981).

[4] T. Tajima, *Computational Plasma Physics: With Applications to Fusion and Astrophysics*. (Addison-Wesley Publishing Company, 1989)

[5] W.L. Kruer,*The Physics of Laser Plasma Interactions*, (Addison-Wesley Publishing Company, 1988)

[6] C.K. Birdsall and A.B. Langdon, *Plasma Physics via Computer Simulation* (McGraw-Hill, New York, 1985)

[7] O.C. Zienkiewicz and R.L. Taylor, *The Finite Element Method*, 4th edition, V.1 and 2(McGraw-Hill, London)

[8] P.P. Silvester and R.L. Ferrari, *Finite Elements for Electrical Engineers*, 2th edition (Cambridge University Press, 1990)

[9] G. Arfken, *Mathematical Methods for Physicists*, 2nd edition, (Academic Press, 1970)

[10] O. Axelsson,*Iterative Solution Methods* (Cambridge University Press, 1996)

[11] N.A. Krall and A.W. Trivelpiece, *Principles of Plasma Physics* (McGraw-Hill 1973)

[12] R. Huber, F. Tauser, A. Brodschelm, M. Bichler, G. Abstreiter, and A. Leitenstorfer, Nature, **414**, 286 (2001)

[13] H. Haug, Nature, **414**, 261 (2001)

[14] N. Rostoker, Nucl. Fus.**1**, 101 (1961)

[15] V.K. Decyk, Comp. Phys. Rep.**4**, 245 (1986)

[16] V.K. Decyk, F.S. Tsung, P.C. Liewer, P.M. Lyster, and R.D. Ferraro,“Particle Simulation on Distributed Memory Parallel Computers”, IPFR-UCLA Technical Report, PPG-1446, July 1992.

[17] A.C.J. Paes, N.M. Abe, V.A. Serrão, and A.Passaro, “Electrostatic Models for Plasma Simulation: Coupling ‘Particle-in-Cell’ to the Finite Element Method”, IEAv/CTA Technical Report (In Portuguese) CTA/IEAv-EFA/NT-001/2001