Journal of Computational Physics 228 (2009) 1017–1029

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

# A PIC based procedure for the integration of multiple time scale problems in gas discharge physics

C. Soria-Hoyo [a,*], F. Pontiga [b], A. Castellanos [a]

[a] Dpto. Electrónica y Electromagnetismo, Universidad de Sevilla, Facultad de Física, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
[b] Dpto. Física Aplicada II, Universidad de Sevilla. EUAT, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain

ARTICLE INFO

*Article history:*
Received 14 February 2008
Received in revised form 4 September 2008
Accepted 7 October 2008
Available online 17 October 2008

*PACS:*
52.65.Rr
52.80.Hc
52.65.Kj
82.33.Xj
02.70.−c

*Keywords:*
Particle-in-cell method
Glow discharge
Corona discharge
Drift-diffusion approximation
Plasma reactions
Computational technique
Numerical simulation

ABSTRACT

A efficient PIC technique has been implemented to study the development of electrical discharges during long periods of time. Special motivation is provided by electrical pulsations that develop in very short times but whose repetition period is much longer. The method exploits the existence of different time scales in the electrical discharge to implement a long time-step particle pushing technique both at particle and at mesh levels. The development of a train of hundreds of Trichel pulses, which is a prohibitively long computation with a conventional PIC, has been used to test the validity of the method.

## 1. Introduction

Numerical simulation of electrical discharges in gases constitutes a complex task, since the non-linear interplay among ionization, charge density and electric field gives rise to the generation of sharp gradients and propagating shock waves. These facts have motivated the development of different numerical techniques, many of them specially designed to minimize numerical diffusion [1–4]. The pioneer works of Davies et al. [5] used the method of characteristics, which can be easily adapted to the problem of ionization in gases. However, this method required, at least in its original implementation, the interpolation from mesh nodes at each temporal step of the simulation, and did not include any special refinement in order to reduce numerical diffusion. Particularly successful results have been obtained through the use of flux-corrected-transport algorithms with finite differences (FD-FCT). This method is an Eulerian technique that has been optimized to achieve very low numerical diffusion in problems with fronts or shock-like discontinuities [6]. In fact, the Phoenical-FCT method is

[*] Corresponding author. Tel.: +34 954559898; fax: +34 954239434.
*E-mail addresses:* cshoyo@us.es (C. Soria-Hoyo), pontiga@us.es (F. Pontiga), castella@us.es (A. Castellanos).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.10.007

capable of maintaining a steep gradient, without significant distortion, for several thousands of temporal steps (see Fig. 6 in [6]), thus constituting an invaluable tool in the research of corona discharges. Two-dimensional simulation of gaseous discharges have also been also performed using this technique [7,8] and, more recently, the scheme has been successfully extended to two-dimensional and three dimensional geometries by combining FCT with finite element methods [9]. Many other numerical schemes have been applied to the area of gaseous discharges: specially adapted finite-element techniques [10], finite difference Scharfetter–Gummel techniques [4] and, more recently, MUSCL and QUICKEST [11], as well as others techniques [12]. According to the rich bibliography produced, all these methods have been fruitful in fundamental research of transient phenomena in electrical discharges.

Particle-in-cell (PIC) techniques, in spite of being a usual tool in kinetic plasma simulations [13], have not been so widely applied to the simulation of electrical discharges in gases, at least in the drift-diffusion approximation. A modular PIC approach that encompass both fluid and kinetic plasma behavior has been developed by Lapenta et al. [14], who were specially motivated in obtaining the steady-state solutions of DC discharges by solving the time-dependent equations. In a previous work, the authors have also implemented a fluid PIC technique to simulate transient electrical discharges [15]. The proposed implementation included a special treatment of the source/sink terms of particle densities, which improved the precision of the numerical simulation. The PIC technique was then compared with a standard FCT method, and the agreement between both techniques turned out to be remarkable. This is a usual conclusion when comparing PIC with finite difference techniques: for problems that are tractable by finite-difference methods, PIC and FCT give very similar results, but PIC always has a higher computational cost. However, fluid PIC codes, such as FLIP (Fluid-Implicit-Particle) [16], show its potential in specially difficult problems where finite difference methods are not appropriate. In particular, fluid PIC techniques becomes very valuable in the simulation of quasi-stationary states, specially those with large density gradients perpendicular to the flow velocity [17–19].

The goal of this work is to formulate an efficient fluid PIC method to study the development of electrical discharges during long periods of time. PIC methods are particularly adequate for this task, owing to its very low numerical diffusion. The proposed method exploits the existence of very different time scales to implement a long time-step pushing technique [20,21] on computational particles and on virtual node-particles, in a way somewhat inspired in FLIP [16]. The result is a very fast method, capable of simulating electrical pulsations (like Trichel pulses) that usually requires a huge number of computational steps. The proposed technique accelerates PIC calculation to the point of being faster than finite difference techniques.

The paper is organized as follows. In the first section, the basic concepts of the method are reviewed and discussed. In the second section, the long time-step pushing technique is implemented at particle level, and then extended to the grid level in the next section. Finally, in the last section, the problem of the development of a train of Trichel pulses is used to test validity of the method, and a comparison with a conventional PIC method is done.

## 2. Fundamentals of the method

In the drift-diffusion approximation, the density of species in a one-dimensional electrical gas discharge is governed by a set of continuity equations of the form [6]

$$\frac{\partial \rho_i}{\partial t} + \frac{\partial}{\partial x}(\rho_i(x,t)V_i(x,t)) = S_i(x,t), \quad i = 1, 2, \cdots, l, \tag{1}$$

where $\rho_i$, $V_i$ and $S_i$ represent the number density, the velocity and the source–sink term of the species *i*, respectively, and *l* is the total number of species. Typically, this set of equations models the spatio-temporal evolution of an electrical discharge that progresses along a narrow channel in *x* direction.

The conservation equations for the species densities are coupled to Poisson's equation through the charge density,

$$\nabla^2 \phi = -\frac{1}{\varepsilon_0}\sum_{i=1}^{l} q_i \rho_i. \tag{2}$$

where $\phi$ is the electrical potential, $\varepsilon_0$ is the gas permittivity and $q_i$ is the electric charge of particles of the species *i*. Poisson's equation must be solved in three dimensions to account for the finite radial extent of the discharge channel [5,22].

As it is well known, PIC methods replace the continuous particle density $\rho_i(x,t)$ with a set of discrete computational particles or *superparticles*. Using the image of these computational particles as "clouds" of physical particles [23], it was show in [15] (see also [24]) the convenience of considering each species density as the product of a density of *fictitious carriers*, $N_i$, whose number is conserved, and a variable number of physical particles, $\mu_i$,

$$\rho_i(x,t) = N_i(x,t)\mu_i(x,t). \tag{3}$$

With this approach, each equation in (1) splits into two independent equations

$$\frac{\partial N_i}{\partial t} + \frac{\partial}{\partial x}(N_i V_i) = 0, \tag{4}$$

$$\frac{d\mu_i}{dt} = \frac{S_i}{N_i}, \tag{5}$$

where $d/dt$ represents the Lagrangian time derivative $d/dt = \partial/\partial t + \mathbf{V}_i\partial/\partial x$. Some of the physical processes contributing to the source–sink term $S_i$ can be directly proportional to the number density of the species $i$. Therefore, the term $S_i$ can be expressed as

$$S_i(x,t) = \overline{S}_i(x,t) + K_i(x,t)\rho_i(x,t), \tag{6}$$

where $K_i$ is a proportionality constant and $\overline{S}_i$ brings together all the terms that are non-linear in $\rho_i$. The dependence of these functions on $x$ and $t$ may be either explicit or implicit through other variables, like species densities or the electric field. Eq. (5) can then be written as

$$\frac{d\mu_i}{dt} = \frac{\overline{S}_i(x,t)}{N_i(x,t)} + K_i(x,t)\mu_i(x,t). \tag{7}$$

The discretization of Eq. (1) or, equivalently (4,5), is made through the use of superparticles, whose interactions are evaluated on a grid. In what follows, the magnitudes related to the grid nodes and computational particles will be designated with indices $g$ and $p$, respectively. In particular, the location of grid nodes will be designated as $X_g$, and the position of computational particles of the species $i$ will be denoted by $x_i^p(t)$. These computational particles can be seen as the natural discretization of the *fictitious carriers*, with density $N_i$, introduced in (3). Therefore, at the grid nodes, $N_i$ will be approximated by the density of computational particles, $n_i^g$, which can be obtained as a weighted average on particle positions [25]

$$N_i(X_g,t) = n_i^g = \frac{1}{\Delta X_g}\sum_p W_{\mathrm{CIC}}^g(x_i^p), \tag{8}$$

where $W_{\mathrm{CIC}}^g(x_i^p)$ represents the weighting function (or assignment function) for the particle position $x_i^p$ and grid node $g$, and $\Delta X_g = (X_{g+1} - X_{g-1})/2$ is taken as the cell width corresponding to the grid node $X_g$. For consistency, the weighting function must satisfy the following normalization condition

$$\sum_g W_{\mathrm{CIC}}^g(x_i^p) = 1. \tag{9}$$

Different types of assignment functions can be used for the interpolation between grid nodes and particles [23]. Here, the subscript CIC indicates that a cloud-in-cell interpolation scheme has been chosen.

Moreover, each computational particle has a certain number of associated physical particles, $\mu_i^p(t)$. The density of the species $i$ at $X_g$ can then be approximated as

$$\rho_i(X_g,t) = \frac{1}{\Delta X_g}\sum_p \mu_i^p(t) W_{\mathrm{CIC}}^g(x_i^p). \tag{10}$$

Using this notation, the equations governing the temporal evolution of $x_i^p(t)$ and $\mu_i^p(t)$ can be written as

$$\frac{dx_i^p}{dt} = v_i(x_i^p,t) \tag{11}$$

$$\frac{d\mu_i^p}{dt} = \sigma_i(x_i^p,t) + k_i(x_i^p,t)\mu_i^p(t), \tag{12}$$

where $\sigma_i$, $v_i$ and $k_i$ are the discretized versions of $\overline{S}_i/N_i$, $V_i$ and $K_i$, respectively. At the grid nodes, the values of $\sigma_i$, $v_i$ and $k_i$ are equal to the corresponding of $\overline{S}_i/N_i$, $V_i$ and $K_i$, which are functions of the species densities and of the electric field. Elsewhere they must be interpolated from their values at grid nodes using the weighting functions. Particularly, $\sigma_i(x_i^p,t)$ is obtained as the interpolation of the ratio $\overline{S}_i/N_i$, and not as the ratio of the interpolated values of $\overline{S}_i$ and $N_i$, independently. Moreover, due to the differentiated treatment of proportional and non-proportional source–sink terms, only $K_i$ (and not $K_i\rho_i/N_i$) is interpolated from the grid to the particles for source–sink terms that are proportional to the species density. In contrast, for non-proportional source–sink terms, it is $\overline{S}_i/N_i$ that must be interpolated to the computational particles. As it was shown by Soria et al. [15], this differentiated treatment of proportional and non-proportional source–sink terms has as a result a more precise numerical simulation.

Eqs. (11) and (12) constitute a set of ordinary differential equations which can be integrated in time using a finite difference method, like the Runge–Kutta method. The temporal step for this integration is limited by both accuracy and stability considerations:

1. The displacement of each computational particle during one temporal step must be smaller than the grid spacing, in order to integrate the particle trajectory with the resolution allowed by the grid.
2. The temporal step must be shorter than the characteristic times associated with the source–sink terms (ionization, attachment, etc.).
3. The temporal step must be shorter than any other characteristic physical time, particularly, the charge relaxation time.

In most of circumstances, the main limitation to the temporal step comes from the first consideration applied to the fastest particles, that is, the electrons in the case of plasmas or electrical discharges. Various techniques have been proposed to mitigate this limitation. For example, subcycling reduces computational time by performing a smaller number of longer temporal steps for ions than for electrons [13]. Other techniques have focused on integrating analytically the electron evolution under certain approximations, and using the obtained expressions to advance particles over a larger temporal step [21,20]. For example, Smolsky [20] has simulated the motion of relativistic particles in electromagnetic fields using the analytical solutions of the equations of motion in homogenous or quasi-homogenous electric and magnetic fields. Usually, these semi-analytical techniques must evaluate lengthy expressions for each particle. Therefore, they will only be efficient if the saving in the number of temporal steps compensates for their higher computational cost.

The method proposed in this paper is also a semi-analytical technique, and it is formulated for stationary electric fields. Particle by particle integration is reduced to a minimum, and most of required operations are performed on virtual node-particles following a procedure inspired in FLIP algorithm [16]. However, in the present case, the evolution equations cannot be fully solved using these virtual node-particles, since there are several types of particles interacting with the others through the source–sink terms.

## 3. Implementation of the long time-step on superparticles

In this section, we address the problem of implementing the long time-step particle pushing technique at the particle level. Therefore, the Lagrangian Eqs. (11) and (12) will be integrated in time assuming a stationary, but inhomogeneous, electric field. Grid values of the electric field have been obtained by interpolating the electrical charge of computational particles to the grid nodes using a CIC assignment scheme, and then integrating Poisson's equation numerically [23,13].

The equation governing the evolution of $\mu_i^p$ can be written as a function of particle coordinates or, more conveniently, in terms of *natural coordinates* [16], $\gamma_i^p(t)$,

$$\frac{d\mu_i^p}{d\gamma_i^p} = \frac{\Delta X_{g+1/2}}{v_i(x_i^p,t)}[\sigma_i(x_i^p,t) + k_i(x_i^p,t)\mu_i(x_i^p,t)], \tag{13}$$

where $\gamma_i^p$ is defined as

$$x_i^p = X_g + \gamma_i^p \Delta X_{g+1/2}, \quad X_g \leqslant x_i^p < X_{g+1}, \tag{14}$$

with $\Delta X_{g+1/2} = X_{g+1} - X_g$.

Integration of particle positions is accomplished by using an energy conserving algorithm [23], as momentum-conserving algorithms are known to be unstable for this type of problems [15]. In the energy conserving algorithm, all computational particles of the species *i* lying inside the interval $X_g \leqslant x_i^p < X_{g+1}$ have an identical velocity: the one corresponding to the electric field at the midpoint between grid nodes, $v_i^{g+1/2} = V_i(E_{g+1/2})$, where $E_{g+1/2} = E(X_{g+1/2})$ and $X_{g+1/2} = (X_{g+1} + X_g)/2$. This particle velocity interpolation is equivalent to a nearest-grid-point (NGP) interpolation from a grid whose grid nodes are located at the midpoints of the original grid.

On the contrary, source and sink terms are usually evaluated at the grid nodes, and then they are interpolated to the computational particles. However, as indicated before, the physical processes contributing to the source and sink terms are of different nature and, in particular, some of them are directly proportional to the particle density, with a constant of proportionality that only depends on the particle velocity. For these last processes, a more accurate evaluation will result if it is performed at the same location than the particle velocity, that is, at the midpoint between grid nodes. Therefore, functions $\bar{\sigma}_i(x_i^p)$ and $k_i(x_i^p)$ are defined as

$$\sigma_i(x_i^p,t) = W_{\mathrm{CIC}}^{g}(x_i^p)\sigma_i^g + W_{\mathrm{CIC}}^{g+1}(x_i^p)\sigma_i^{g+1}, \tag{15}$$

$$k_i(x_i^p,t) = W_{\mathrm{NGP}}^{g+1/2}(x_i^p)k_i^{g+1/2}, \tag{16}$$

where $k_i^{g+1/2} = K_i(E_{g+1/2})$ and $W_{\mathrm{CIC}}^{g}(x_i^p)$ and $W_{\mathrm{NGP}}^{g+1/2}(x_i^p)$ denote the CIC and NGP weighting functions, respectively, [23]. The function $\sigma_i^g$ is defined as

$$\sigma_i^g = \frac{\overline{S}_i(X_g)}{N_i(X_g)}. \tag{17}$$

In terms of natural coordinates, CIC weighting reduces to

$$W_{\mathrm{CIC}}^{g+1}(x_i^p) = \gamma_i^p, \tag{18}$$

$$W_{\mathrm{CIC}}^{g}(x_i^p) = 1 - \gamma_i^p, \tag{19}$$

and NGP weithing to

$$W_{\mathrm{NGP}}^{g+1/2}(x_i^p) = \begin{cases} 1, & \text{if } X_g \leqslant x_i^p < X_{g+1}, \\ 0, & \text{in any other case}. \end{cases} \tag{20}$$

Therefore, Eq. (13) becomes

$$\frac{d\mu_i^p}{d\gamma_i^p} = \Delta t_i^{g+1/2}\left[\sigma_i^{g+1}\gamma_i^p + \sigma_i^g(1-\gamma_i^p) + \mu_i^p k_i^{g+1/2}\right], \tag{21}$$

where

$$\Delta t_i^{g+1/2} = \frac{\Delta X_{g+1/2}}{v_i^{g+1/2}}. \tag{22}$$

Eq. (21) can be integrated from $\gamma_0$ to $\gamma_i^p$, yielding

$$\mu_i^p(\gamma_i^p) = \left[\mu_i^p(\gamma_0) + D_i^g + F_i^g\gamma_0\right]\exp[C_i^g(\gamma_i^p - \gamma_0)] - \left(D_i^g + F_i^g\gamma_i^p\right), \tag{23}$$

where

$$A_i^g = \sigma_i^g\Delta t_{g+1/2}, \tag{24}$$

$$B_i^g = (\sigma_i^{g+1} - \sigma_i^g)\Delta t_{g+1/2}, \tag{25}$$

$$C_i^g = k_i^{g+1/2}\Delta t_{g+1/2}, \tag{26}$$

$$D_i^g = \frac{1}{C_i^g}\left(A_i^g + \frac{B_i^g}{C_i^g}\right), \tag{27}$$

$$F_i^g = \frac{B_i^g}{C_i^g}. \tag{28}$$

The application of Eq. (23) to a long time-step particle pushing, along which the particle may cross several cells, requires that both $k_i$ and $\bar{\sigma}_i$ be constants. Since $k_i$ and $\bar{\sigma}_i$ are implicit functions of time through their dependence on electric field, the temporal variation of the electric field during the long time-step should be negligible. Therefore, the long time-step must be necessarily shorter that the charge relaxation time. However, the first two limitations of the temporal step cited in Section 2 have been eliminated, since both $x_i^p$ and $\mu_i^p$ can be integrated semi-analytically. If other characteristic physical times, in addition to the charge relaxation time, were present, the long time-step will be limited by the shortest among them.

The interaction between species during the long time-step imposes additional restrictions to the use of Eq. (23), since $k_i$ and, particularly, $\bar{\sigma}_i$, may depend on the densities of other species different from $i$. Fortunately, in gas discharges and plasmas in general, the most important processes are caused by the interaction of electrons with the background gas, whose density may be taken as a constant (see, for example, the Appendix in [26]). These processes give rise to the generation of new electrons and also to the rest of species. Thus, the source term of a species $i$, different from electrons, can frequently be written as $S_i(x,t) = \kappa_i(x,t)\rho_e(x,t)$, where $\kappa_i$ is a proportionality constant (that may depend on the electric field) and $\rho_e$ is the electron density. These physical considerations suggest restricting the application of Eq. (23) to electrons, and then obtaining the evolution of the rest of species from them.

At a particle level, a superelectron $p$ contributes to the variation of the number of physical particles associated to the species $i$ at a rate $\kappa_i(x_e^p,t)\mu_e^p(t)$, where the index $e$ has been selected for electrons and $i \neq e$. Therefore, instead of solving Eq. (21), the rate $\kappa_i(x_e^p,t)\mu_e^p(t)$ can be assigned to the grid nodes adjacent to the superelectron by using a CIC assignment,

$$\frac{d(\Delta_p M_i^g)}{d\gamma_e^p} = \Delta t_e^{g+1/2}\kappa_i(x_e^p,t)\mu_e^p(t)W_{\mathrm{CIC}}^g(x_e^p), \tag{29}$$

$$\frac{d(\Delta_p M_i^{g+1})}{d\gamma_e^p} = \Delta t_e^{g+1/2}\kappa_i(x_e^p,t)\mu_e^p(t)W_{\mathrm{CIC}}^{g+1}(x_e^p), \tag{30}$$

where $\Delta_p M_i^g$ denotes the variation of the number of physical particles of the species $i$ induced by the electron $p$ and assigned to the grid node $g$. Integration of Eq. (29) gives

$$\Delta_p M_i^g = \Delta t_e^{g+1/2}\kappa_i^{g+1/2}[I_1(\gamma_0,\gamma_e^p) - I_1(\gamma_0,\gamma_0)], \tag{31}$$

$$\Delta_p M_i^{g+1} = \Delta t_e^{g+1/2}\kappa_i^{g+1/2}[I_2(\gamma_0,\gamma_e^p) - I_2(\gamma_0,\gamma_0)], \tag{32}$$

where $\kappa_i^{g+1/2}$ is defined analogously to $k_i^{g+1/2}$ and functions $I_0$, $I_1$ and $I_2$ are expressed as

$$I_0(\gamma_0,\gamma) = \frac{1}{C_e^g}(D_e^g + \mu_e^p(\gamma_0) + F_e^g\gamma_0)\exp[C_e^g(\gamma - \gamma_0)] - \frac{\gamma}{2}(2D_e^g + F_e^g\gamma), \tag{33}$$

$$I_2(\gamma_0,\gamma) = \frac{1}{(C_e^g)^2}(D_e^g + \mu_e^p(\gamma_0) + F_e^g\gamma_0)(C_e^g\gamma - 1)\exp[C_e^g(\gamma - \gamma_0)] - \frac{\gamma^2}{6}(3D_e^g + 2F_e^g\gamma), \tag{34}$$

$$I_1(\gamma_0,\gamma) = I_0(\gamma_0,\gamma) - I_2(\gamma_0,\gamma). \tag{35}$$

The total amount of physical particles of the species $i$ left at the node $g$ by electrons are obtained by adding all the contributions,

$$\Delta M_i^g = \sum_p \Delta_p M_i^g. \tag{36}$$

This amount must then be redistributed among all the superparticles of the species $i$ using the CIC assignment scheme.

Since the long time-step particle pushing technique is only applied to electrons, the long time-step will be limited by the transit time of ions across a computational cell, which is significantly longer than that of electrons. In addition, the long time-step must be shorter than the charge relaxation time in order to ensure that the electric field is quasi-stationary.

## 4. Implementation of the long time-step on node-particles

The algorithm proposed in the previous section may not be computationally efficient if it is applied individually to each superelectron. Of course, part of the coefficients appearing in expressions (23) and (24)–(28) can be precalculated, thus obtaining a certain optimization. However, the fact that the algorithm requires a number of operations proportional to both the number of superparticles and the number of cells crossed by the superparticles during its evolution implies a relatively high computational cost. Therefore, a fundamental change in the algorithm is needed in order to significantly reduce the computational time.

In this section, the temporal evolution of electrons will be partially described in terms of a set of virtual particles located on the grid nodes at the beginning of each long time-step. Thus, the technique is based on the Lagrangian evolution of these virtual particles, which will be denoted as *node-particles*. In this aspect, the method resembles FLIP [16], but a stationary grid is here required to evaluate the interaction between species. After the long time-step, the position and the number of electrons associated to each superelectron will be updated with the information of node-particles.

In 1D problems, the temporal evolution of a PIC-particles partially overlaps the evolution of other superparticles. This fact is not useful in normal simulations, where each superparticle advances a fraction of the computational cell during one temporal step. However, when a long time-step particle pushing technique is being applied, the existence of such overlapping may be exploited to achieve a significant reduction in the computational time. Consider a superelectron, with position $x_e^p$, located between grid nodes $X_g$ and $X_{g+1}$. The temporal evolution of the number of electrons associated to the superparticle from time $t_1$ to time $t_2$ can then be expressed as

$$\int_{t_1}^{t_2} \frac{d\mu_e^p}{dt} dt = \int_{t_1}^{t_1-\Delta t_e^p} \frac{d\mu_e^p}{dt} dt + \int_{t_1-\Delta t_e^p}^{t_2-\Delta t_e^p} \frac{d\mu_e^p}{dt} dt + \int_{t_2-\Delta t_e^p}^{t_2} \frac{d\mu_e^p}{dt} dt, \tag{37}$$

where $\Delta t_e^p$ is the time required for the superelectron to reach the node $X_g$,

$$\Delta t_e^p = \frac{x_e^p - X_g}{v_e^{g+1/2}} = \gamma_e^p \Delta t_{g+1/2}. \tag{38}$$

Therefore, the first integral in the right-hand-side of Eq. (37) can be interpreted as the evolution of the superelectron backward in time until it reaches the node $X_g$. On the contrary, the second integral corresponds to the evolution of the superelectron forward in time during a net time interval of value $t_2 - t_1$. Finally, the third integral further advances the evolution of the superelectron for the time interval $\Delta t_e^p$, in order to compensate for the first integration backward in time.

Clearly, the second integral will be similar for all superelectrons situated between nodes $X_g$ and $X_{g+1}$ provided that $d\mu_e^p/dt$ is invariant to time translations,

$$\int_{t_1-\Delta t_e^p}^{t_2-\Delta t_e^p} \frac{d\mu_e^p}{dt} dt = \int_{t_1}^{t_2} \frac{d\mu_e^p}{dt} dt, \tag{39}$$

since each superelectron is characterized by a different value of $\Delta t_e^p$. Therefore, the integral could be computed by applying the long time-step particle pushing technique to a node-particle initially located at $X_g$. At the beginning of the long time-step, the node particle would be assigned a total amount of physical electrons given by

$$M_e^g = \sum_p \mu_e^p (t_1 - \Delta t_e^p) \tag{40}$$

where the sum is extended to all the superelectrons transported to the node $g$. After the long time-step, the individual evolution of each superelectron must be recovered from the node-particle.

Application of Eqs. (23) and (31, 32) to node-particles must be analyzed in detail, as it must be shown that they can be added at PIC-particle level to obtain the correct evolution of node-particles. In the case of source terms that are proportional to the electron density, these equations become linear in $\mu_e^p$ and, consequently, they can be confidently added at superparticle level. The evolution of the node-particle will be given by

$$M_e^g(\gamma_e^p) = M_e^g(\gamma_0) \exp[C_e^g(\gamma_e^p - \gamma_0)], \tag{41}$$

where $\gamma_0 = 0$ at its initial position, the node $X_g$. In addition, the number of physical particles of the species $i$ (different from electrons) is incremented on the grid according to (31,32), where

$$I_0(\gamma_0,\gamma)=\frac{M_e^g(\gamma_0)}{C_e^g}\exp[(C_e^g(\gamma-\gamma_0)] \tag{42}$$

$$I_2(\gamma_0,\gamma)=\frac{M_e^g(\gamma_0)}{(C_e^g)^2}(C_e^g\gamma-1)\exp[C_e^g(\gamma-\gamma_0)]. \tag{43}$$

This procedure can be iteratively applied to a node-particle crossing several cells. After the long time-step, the number of physical electrons associated to each superelectron is extracted from the node-particle as

$$\mu_e^p(t_2-\Delta t_e^p)=\frac{M_e^g(\gamma_e^p)}{M_e^g(\gamma_0)}\mu_e^p(t_1-\Delta t_e^p). \tag{44}$$

The case of source terms that are not proportional to the electron density is not so obvious. Consider the application of Eq. (23) and the form of coefficients $A_e^g$ to $F_e^g$. From the definitions of $n_e^g$ and $\sigma_e^g$ given by Eqs. (8) and (17), it is

$$\sigma_e^g=\frac{\Delta X_g\overline{S}_e(X_g)}{\sum\limits_p W_{\mathrm{CIC}}^g(x_e^p)}. \tag{45}$$

For PIC-electrons transported to the node $g$, the denominator in Eq. (45) reduces to the number of superelectrons at the node, since $W_{\mathrm{CIC}}^g(X_g)=1$. Additionally, the value of the denominator remains constant during the evolution of the node-particle, because $W_{\mathrm{CIC}}^g$ is a constant of motion [16]. Therefore, apparently, coefficients $A_e^g$, $B_e^g$, $D_e^g$ and $F_e^g$ are only affected by a constant (the number of superelectrons per node-particle) and the evolution of each individual superelectron could be extracted from the node-particle after the long time-step.

However, two important facts were omitted in the previous discussion. Firstly, the sum in the denominator of Eq. (45) is altered by the transport of superelectrons to the node $g$. This is not an essential error, as it can be fixed by selecting other forms of transporting PIC-particles to nodes. One possibility is splitting each superelectron $p$ in two superparticles, with numbers of physical electrons $W_{\mathrm{CIC}}^g(x_e^p)\mu_e^p$ and $W_{\mathrm{CIC}}^{g+1}(x_e^p)\mu_e^p$, and displacing the first of these superparticles to the node $g$ and the second one to the node $g+1$. This procedure has the advantage of preserving the density of superelectrons on the grid, although at the cost of duplicating the number of PIC-electrons. Secondly, Eq. (45) involves a coupling between particles, through the term $\Delta X_g$, which changes during the evolution of node-particles. Consequently, in the case of non-proportional source terms, while it is still possible to calculate the evolution of superelectrons using node-particles, it is no longer possible to implement a long time-step particle pushing technique.

A review of the computational cycle would be as follows:

1. Interpolate the PIC-particles to the grid to obtain the species densities on the grid nodes,
2. Calculate the characteristic times.
3. If the characteristic times are much longer than the transit time of electrons across a computational cell, apply the long time-step particle pushing technique:
   (a) Advance superelectrons backwards in time to the nearest node using Eq. (23), compute Eqs. (31) and (32), and save the delay $\Delta t_e^p$ as given by Eq. (38).
   (b) Generate node-particles from superelectrons at every grid node, advance node-particles a time interval $t_2-t_1$ using Eq. (41), and compute Eqs. (31) and (32).
   (c) Extract superelectrons from the node-particles, advance superelectrons a time interval $\Delta t_e^p$ using Eq. (23), and compute Eqs. (31) and (32).
   (d) Advance the superparticles of rest of species (different from electrons) a time interval $t_2-t_1$. During this phase, the total amount of physical particles generated by superelectrons and node-particles during their evolution and left on the nodes (Eq. (36)) must be taken into account.
4. Else, if the characteristic times are comparable to the transit time of electrons across a computational cell, integrate according to a traditional PIC [15].

## 5. Numerical test: development of Trichel pulses in pure oxygen

The pulsations known as Trichel pulses have been selected to test the suitability of the proposed technique for the simulation of physical phenomena with different temporal scales. These pulses appear when a sharp electrode is subjected to a negative DC voltage with value close to the onset of corona discharge.

In this work, Trichel pulses will be modelled using a simplified version of R. Morrow's one-dimensional model of corona discharge [6]. The discharge is assumed to develop between a sphere, with radius $R=2$ mm, and a grounded plane at a distance $d=2$ cm. The sphere is subjected to a negative high voltage, $\phi_0=-2350$ V, and the discharge channel is modelled as cylinder of radius $r=0.8$ mm with uniform properties in each normal section. The equations governing this problem are

$$\frac{\partial \rho_e}{\partial t} + \frac{\partial}{\partial x}(V_e \rho_e) = (\alpha - \eta)|V_e|\rho_e, \tag{46}$$

$$\frac{\partial \rho_+}{\partial t} + \frac{\partial}{\partial x}(V_+ \rho_+) = \alpha|V_e|\rho_e, \tag{47}$$

$$\frac{\partial \rho_-}{\partial t} + \frac{\partial}{\partial x}(V_- \rho_-) = \eta|V_e|\rho_e, \tag{48}$$

together with Poisson's equation

$$\nabla^2 \phi = -|q_e| \frac{-\rho_e + \rho_+ - \rho_-}{\varepsilon_0}, \tag{49}$$

where subscripts $e$, $+$ and $-$ refer to electrons, positive ions and negative ions, respectively, and $\alpha$ and $\eta$ denote the coefficients of ionization and attachment per unit length. These coefficients, as well as the velocities of electrons and ions, are assumed to be function of the local electric field [27], and their values are taken from [6]. The coordinate $x$ is taken along the symmetry axis of the discharge channel, with $x = 0$ on the sphere (the cathode) and $x = d$ at the plane (the anode).

Poisson's equation is solved as the sum of the Laplacian electric field plus the electric field caused by the space charge present in the discharge channel. This last electric field is calculated by the method of disks [5]. Boundary conditions for ions express the fact that there are neither negative ions on the cathode nor positive ions on the anode,

$$\rho_-(0,t) = 0, \tag{50}$$

$$\rho_+(d,t) = 0. \tag{51}$$

However, electrons are emitted from the cathode by secondary emission, as the result of the impact of positive ions,

$$\rho_e(0,t) = \Gamma \frac{\rho_+(0,t)|V_+(0,t)|}{|v_e(0,t)|}, \tag{52}$$

where $\Gamma$ is the secondary-emission coefficient and its value has been taken from [6]. Finally, the current intensity in the external circuit is calculated by Sato's equation [28],

$$I = \frac{\pi r^2 |q_e|}{\phi_0} \int_0^d (\rho_+ V_+ - \rho\_V_- - \rho_e V_e) E_L dx, \tag{53}$$

where $E_L$ is the Laplacian electric field.

### 5.1. Results and discussion

The PIC simulation was run on a non-uniform grid with 218 nodes that has a very fine constant resolution in the vicinity of the cathode and becomes coarser as it approaches the anode. An initial Gaussian distribution of seed electrons was set in the proximity the cathode, in order to trigger the electrical discharge under the effect of the applied electric field. The development of the electrical discharge was simulated for 1 ms. This interval of time is sufficiently long to have a reasonable number of Trichel pulses in a stable pulsating regime.

The results of the numerical simulation corresponding to the total current intensity are shown in Fig. 1. The amplitude of the first Trichel pulse is significantly higher than that of subsequent pulses because it starts its development in a region void of space-charge. On the contrary, the amplitude of pulses within the subsequent train of pulses are all of smaller and similar magnitude. The elapsed time between the first and the second Trichel pulse is also much longer than between any other two pulses in the train of pulses. This is a consequence of the high density of charge generated by the first pulse, which shields the Laplacian electric field and inhibits the development of the second Trichel pulse. Only after the space charge has been substantially removed the second pulse will appear. In the train of pulses, the charge generated by each pulse is much smaller, and the interval of time between pulses are correspondingly shorter.

Fig. 2 depicts the evolution of the relevant characteristic times during the development of six consecutive Trichel pulses. Clearly, the shortest time is the one associated with the transit time of electrons across a computational cell, $\tau_e = \Delta X_{g+1/2}/|V_e|$. The characteristic temporal scale of ions, $\tau_i = \Delta X_{g+1/2}/|V_+| \sim \Delta X_{g+1/2}/|V_-|$, is two orders of magnitude higher than that of electrons, due to their lower mobilities. The source–sink terms of particle densities introduce two additional characteristic times, associated to the processes of ionization, $\tau_\alpha = (\alpha|V_e|)^{-1}$, and electron attachment, $\tau_\eta = (\eta|V_e|)^{-1}$. Finally, the longest temporal scale is the one corresponding to the charge relaxation time,

$$\tau_r = \frac{\epsilon}{|q_e|(K_e \rho_e + K_+ \rho_+ + K_- \rho_-)}, \tag{54}$$

where $K$ denotes the mobility of charged carriers. This last scale represents the time interval required for a significative variation of the electric field.

In conventional PIC simulations, the transit time of electrons across the computational cells (the shortest temporal scale) constitutes an upper limit for the computational time-step. Since $\tau_e \sim 10^{-11}$ s, the numerical simulation of the discharge development during 1 ms would therefore require more than $10^8$ computational cycles. However, in the present PIC

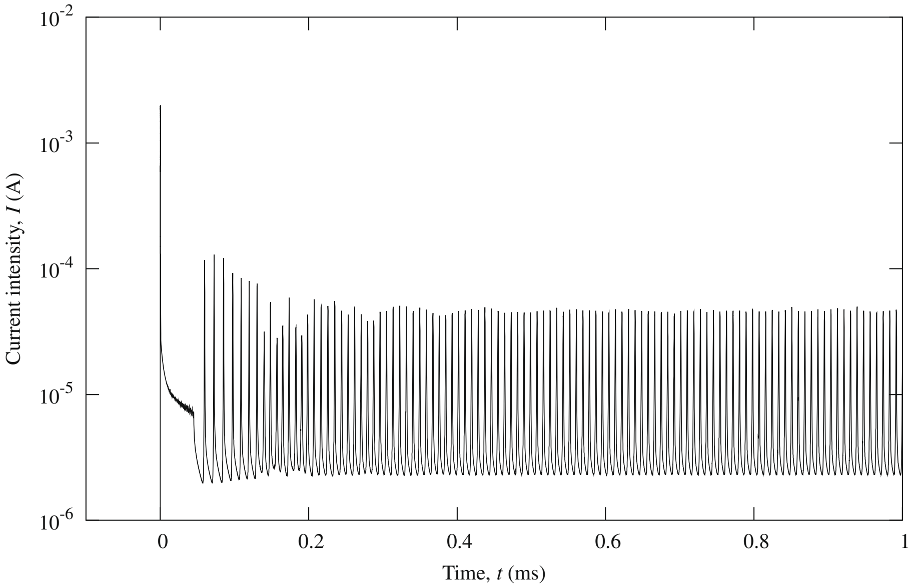

**Fig. 1.** Total current intensity versus time corresponding to train of Trichel pulses for $\phi_0 = -2350$ V.

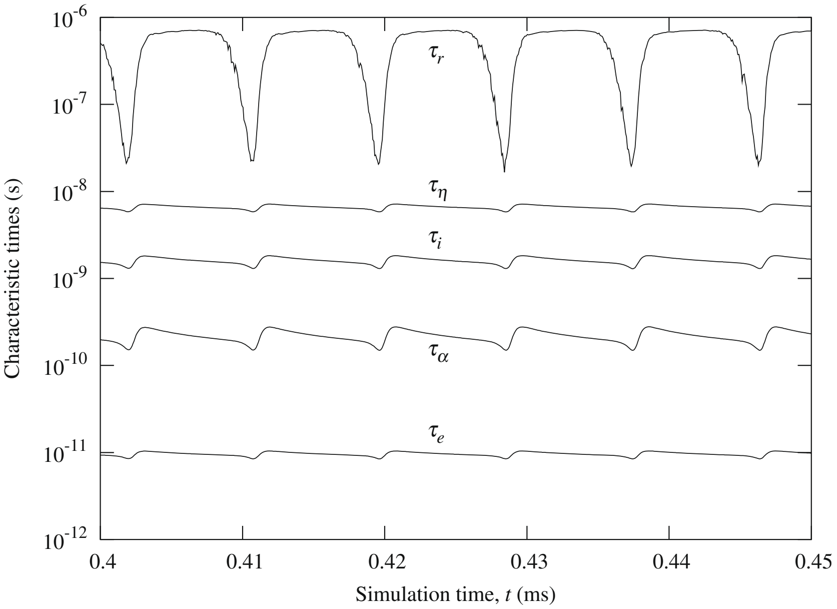

**Fig. 2.** Temporal evolution of the characteristic times during the development of several Trichel pulses. $\tau_r$: charge relaxation time, $\tau_\alpha$: ionization time, $\tau_\eta$: electron attachment time, $\tau_e$ and $\tau_i$: transit time of electrons and ions across a computational cell, respectively.

technique, the computational time-step is only limited by the charge relaxation time and the transit time of ions, which is, as previously indicated, more than two order of magnitude longer than the transit time of electrons. Since the transit time of ions is also shorter than the charge relaxation time, the electric field will be quasi-stationary during the computational time-step. Therefore, the position of superparticles and the number of physical particles associated to them can be integrated by applying the procedure explained in Sections 3 and 4.

### 5.2. Comparison with a conventional PIC method

In this section, the results obtained by using the proposed PIC technique will be compared with those obtained by means of a standard PIC method. However, the numerical simulation will be restricted to a single pulse ($\sim 10\,\mu$s), since the simulation of the complete train of Trichel pulses using a conventional PIC technique would require a huge computational time. Therefore, the spatial distribution of species densities predicted by the new PIC technique, at $t = 440$ μs, has been taken as the initial condition to start the numerical simulation with the standard PIC method.

The total current intensity obtained from the two techniques is first compared in Fig. 3. In Fig. 4, the separated contributions of electrons (left), positive ions (center) and negative ions (right) to the total current intensity is also shown. Clearly, the agreement between the two techniques is remarkable.

A more stringent test would be to compare the spatial distribution of electrons and ions at a certain instant of time. This comparison is performed in Fig. 5, where the spatial distribution of species are depicted at the times corresponding to the peak value of the total current intensity (left) and at the end of the Trichel pulse (right). Again, the agreement between the

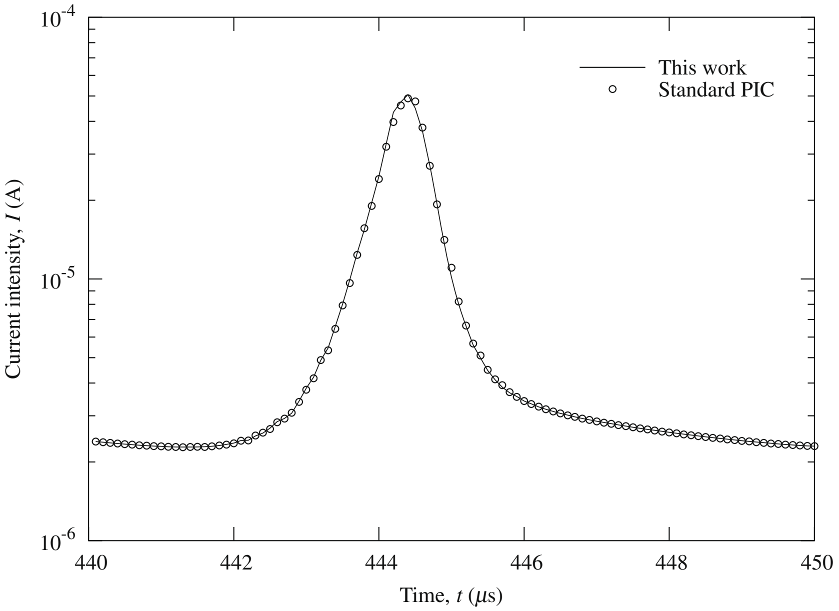

**Fig. 3.** Total current intensity versus time for a single Trichel pulse according to the proposed PIC technique and a conventional PIC method.

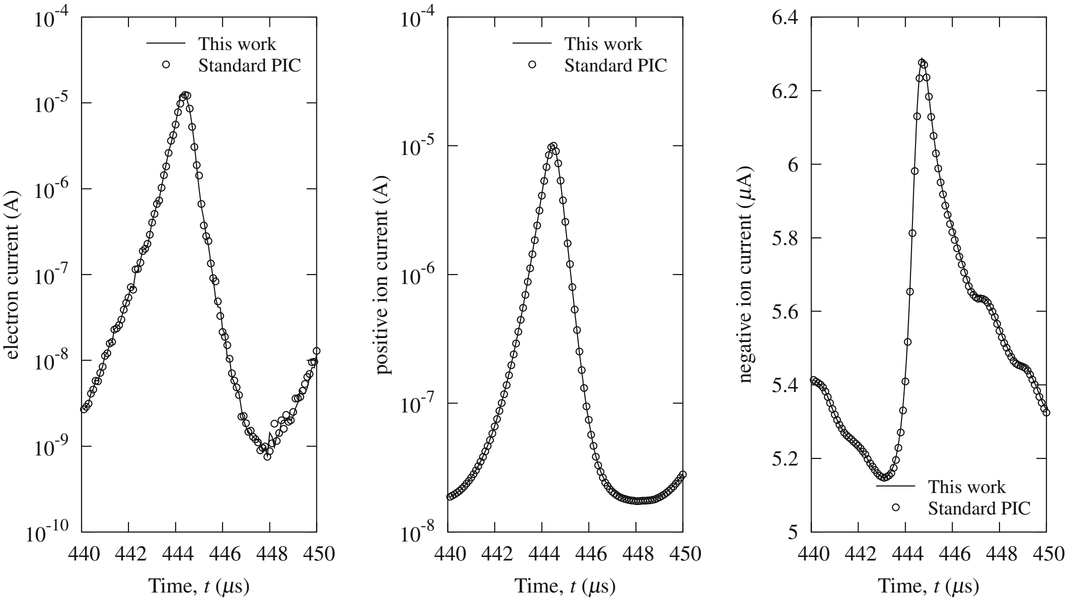

**Fig. 4.** Contributions of electrons (left), positive ions (center) and negative ions (right) to the total current intensity of a single Trichel pulse according to the proposed PIC technique and a conventional PIC method.

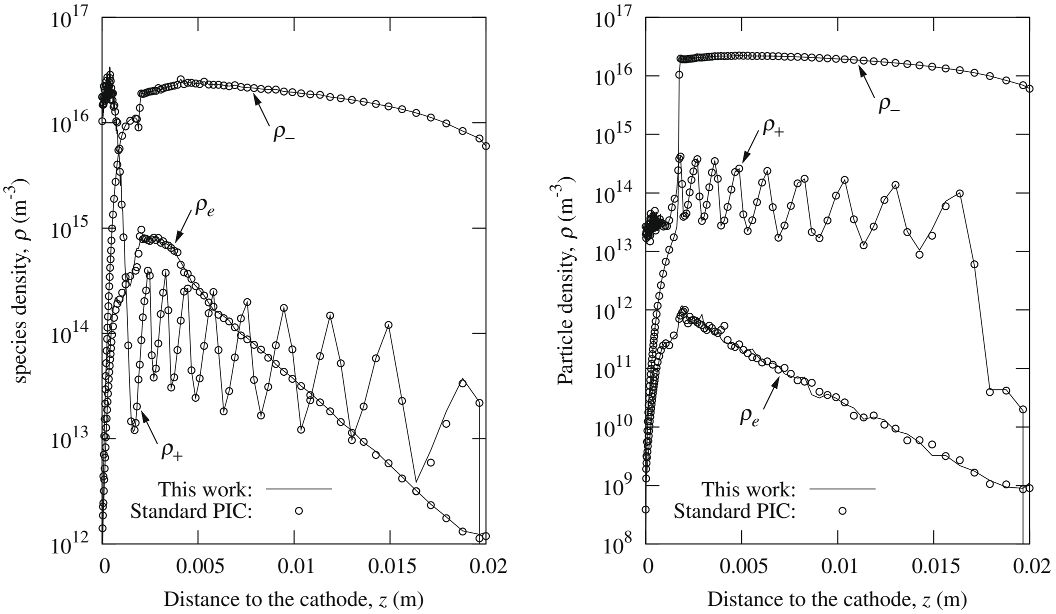

**Fig. 5.** Spatial distribution of electrons and ions at the time of maximum of current intensity (left) and at the end of the Trichel pulse (right) according to the proposed PIC technique and a conventional PIC method.

two techniques is excellent. The oscillations exhibited by the positive ion density are originated by the raise and fall of electron density with each Trichel pulse. The trigger of a Trichel pulse entails an augmentation of the electron density everywhere in the discharge gap (cf. Fig. 5-left and Fig. 5-right). Additionally, in the vicinity of the anode, the electric field intensity is sufficient to ionize the background oxygen, thus generating positive ions. These ions will drift towards the anode impelled by the electric field, but the migration time of ions is roughly 1/10th of the period of Trichel pulses. Therefore, about ten peaks of positive ions density will be always present in the space between electrodes.

Finally, the spatial distribution of the electric field, according to the two numerical techniques, are compared in Fig. 6 at the same two instants of time previously considered. In the proximity of the anode, the sharp drop of electric field produces a

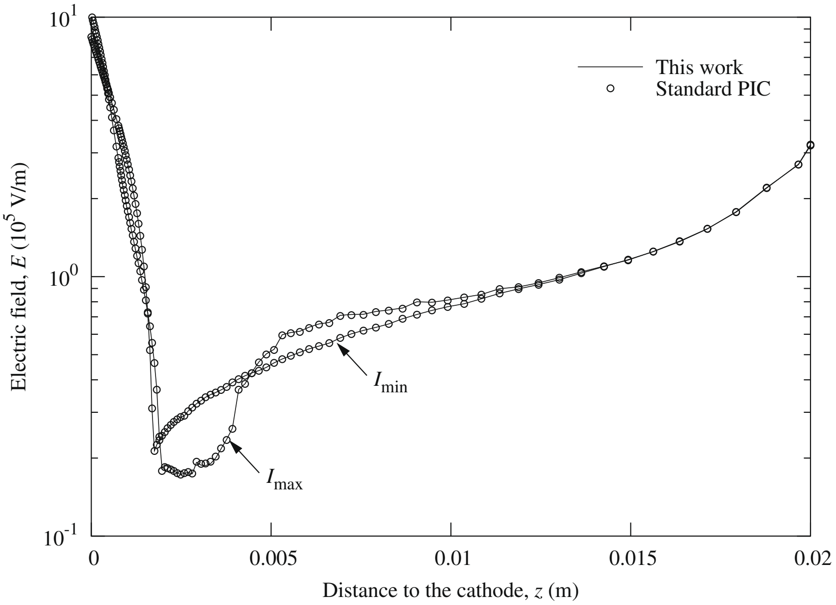

**Fig. 6.** Spatial distribution of electric field at the time of maximum of current intensity and at the end of the Trichel pulse according to the proposed PIC technique and a conventional PIC method.

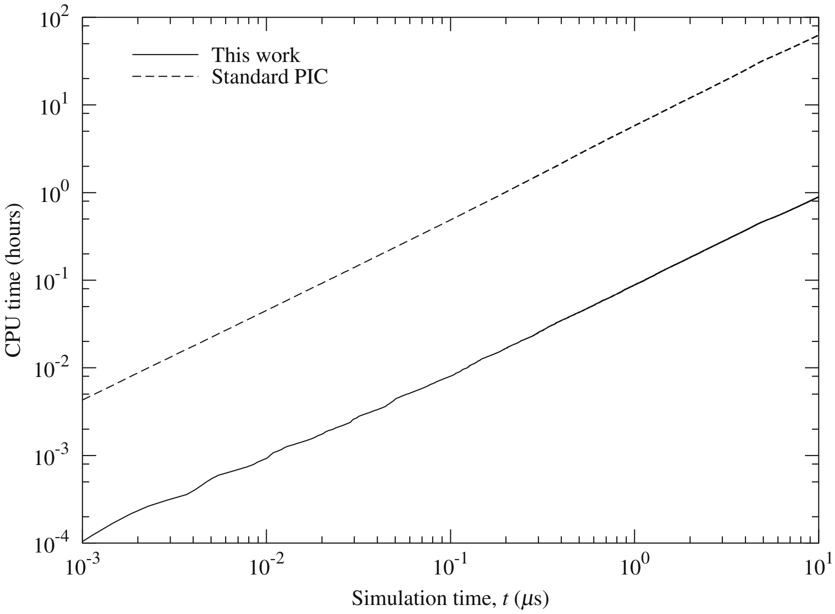

**Fig. 7.** Elapsed CPU time versus simulation time corresponding to the numerical simulation of a single Trichel pulse using the proposed PIC technique and a conventional PIC method.

large gradient of velocity, which favors the accumulation superelectrons at the minimum of electric field. On the other hand, since positive ions are moving in a growing velocity field, they are separated one from the another. Therefore, the superparticle density of positive ions becomes very low and may even fall under one superparticle per cell. To avoid these problems, superparticles are coalesced and split conveniently to maintain a reasonable density of superparticles everywhere in integration domain. Very sophisticated techniques have been developed for splitting and coalescing particles [29–31]. In this work, a simplified version of procedure proposed by Lapenta and Brackbill [29] has been applied to maintain the fluctuations of particles densities to a reasonable degree.

The advantage of the proposed PIC technique over the standard PIC method resides in its superior computational efficiency. This fact can be readily appreciated in Fig. 7, where the elapsed CPU time is depicted as a function of the simulation time. The elapsed CPU time increases almost linearly with the simulation time but, according to these results, the PIC implementation proposed in this work is about 70 times faster than the conventional PIC technique. The CPU time represented in Fig. 7 only corresponds to the time spent in the routines that integrate the particle positions and the number of physical particles associated to the computational particles.

## 6. Conclusions

A fluid PIC method has been proposed for the integration of continuity equations using a long time-step particle pushing procedure. In gas discharges, electrons are usually responsible for the generation of the rest species. Therefore, the use of the long time-step can be limited to superelectrons, and then deduce from them the evolution of the other species. The computational time can be significantly reduced with the introduction of node-particles that allow the implementation of the long time-step particle pushing technique at grid level. In the proposed method, the time-step is only limited by the charge relaxation time and the transit time of ions across the computational cells, which are much longer that transit time of electrons. The validity of the method has been checked with the simulation of a train of Trichel pulses lasting 1 ms. The results obtained with the proposed method are in excellent agreement with the predictions of a conventional PIC technique, but the new method is nearly 70 times faster than the standard PIC method.

## Acknowledgements

This research has been supported by Spanish *Ministerio de Ciencia y Tecnología* (Contract FIS2006-03645) and *Junta de Andalucía* (Contract FQM 421).

## References

[1] R. Morrow, Numerical solution of hyperbolic equations for electron drift in strongly non-uniform, Journal of Computational Physics 43 (1981) 1–15.
[2] A.J. Davies, Discharge simulation, IEE Proceedings A 133 (4) (1986) 217–240.

[3] E.E. Kunhardt, C. Wu, Towards a more accurate flux corrected transport algorithm, Journal of Computational Physics 68 (1987) 127–150.

[4] A.A. Kulikovsky, A more accurate Scharfetter–Gummel algorithm of electron transport for semiconductor and gas discharge simulation, Journal of Computational Physics 119 (1995) 149–155.

[5] A.J. Davies, C.J. Evans, F. Llewellyn Jones, Electrical breakdown of gases: the spatio-temporal growth of ionization in fields distorted, in: Proceedings of the Royal Society of London Series A-Mathematical and 281, 1964, pp. 164–183.

[6] R. Morrow, Theory of negative corona in oxygen, Physical Review A 32 (3) (1985) 1799–1809.

[7] S.K. Dhali, P.F. Williams, Two-dimensional studies of streamers in gases, Journal of Applied Physics 62 (12) (1987) 4696–4707.

[8] P. Vitello, B. Penetrante, J.N. Bardsley, Simulation of negative streamer dynamics in nitrogen, Physical Review E 49 (6) (1994) 5574–5598.

[9] G.E. Georghiou, R. Morrow, A.C. Metaxas, An improved finite-element flux-corrected transport algorithm, Journal of Computational Physics 148 (1999) 605–620.

[10] M. Yousfi, A. Poinsignon, A. Hamani, Finite element method for conservation equations in electrical gas discharge areas, Journal of Computational Physics 113 (1994) 268–278.

[11] A. Bourdon, K. Hassonuni, E. Marode, J. Paillol, P. Ségur, Comparison between some numerical schemes usually adopted for the numerical modelling of the highly convective streamer propagation phenomena, in: XV International Conference on Gas Discharges and Their Applications, 2004, pp. 311–314.

[12] K. Ramakrishna, I.M. Cohen, P.S. Ayyaswamy, Numerical methods for two-dimensional analysis of electrical breakdown in a non-uniform gap, Journal of Computational Physics 104 (1993) 173–184.

[13] C.K. Birdsall, A.B. Langdon, Plasma Physics Via Computer Simulation, IOP Publishing Ltd., Bristol, UK, 1991.

[14] G. Lapenta, F. Iinoya, J.U. Brackbill, Particle-in-cell simulation of glow discharges in complex geometries, IEEE Transactions on Plasma Science 23 (4) (1995) 769–779.

[15] C. Soria, F. Pontiga, A. Castellanos, Particle-in-cell simulation of electrical gas discharges, Journal of Computational Physics 171 (2001) 47–78.

[16] J.U. Brackbill, H.M. Ruppel, FLIP: A method for adaptively zoned particle-in-cell calculations of fluid flows in two, Journal of Computational Physics 65 (1986) 314–343.

[17] P.A. Vázquez, G.E. Georghiou, A. Castellanos, Characterization of injection instabilities in electrohydrodynamics by numerical modelling: comparison of particle in cell and flux corrected transport methods for electroconvection between two plates, Journal of Physics D: Applied Physics 39 (13) (2006) 2754–2763.

[18] A. Castellanos, A.T. Pérez, P. Atten, Charge diffusion versus coulomb repulsion in finite amplitude electroconvection, in: Conference Record of the 1989 IEEE Industry Applications Society Annual Meeting, vol. 2, 2005, pp. 2112–2117.

[19] R. Chicón, A. Castellanos, A. Martín, Numerical modelling of coulomb-driven convection in insulating liquids, Journal of Fluid Mechanics 344 (1997) 43–66.

[20] M. Smolsky, Long time-step particle pushing in drift approximation without orbit averaging, Journal of Computational Physics 145 (1998) 41–60.

[21] C.E. Rathmann, J.L. Vomvoridis, J. Denavit, Long-time-scale simulation of resonant particle effects in langmuir and whistler waves, Journal of Computational Physics 26 (1978) 408–442.

[22] R. Morrow, J.J. Lowke, Streamer propagation in air, Journal of Physics D: Applied Physics 30 (1997) 614–627.

[23] R.W. Hockney, J.W. Eastwood, Computer Simulation using Particles, Adam Hilger, Bristol, 1988.

[24] P.J. O'Rourke, J.U. Brackbill, B. Larrouturou, On particle-grid interpolation and calculating chemistry in particle-in-cell methods, Journal of Computational Physics 109 (1993) 37–52.

[25] J.U. Brackbill, The ringing instability in particle-in-cell calculations or low-speed flow, Journal of Computational Physics 75 (1988) 469–492.

[26] C. Soria, F. Pontiga, A. Castellanos, Plasma chemical and electrical modelling of a negative DC corona in pure oxygen, Plasma Sources Science and Technology 13 (2004) 95–107.

[27] J.J. Lowke, D.K. Davies, Properties of electric discharges sustained by a uniform source of ionization, Journal of Applied Physics 48 (1977) 4991–5000.

[28] N. Sato, Discharge current induced by the motion of charged particles, Journal of Physics D: Applied Physics 13 (1980) L3–L6.

[29] G. Lapenta, J.U. Brackbill, Dynamic and selective control of the simulation of the number of particle in kinetic plasma, Journal of Computational Physics 115 (1994) 213–227.

[30] G. Lapenta, J.U. Brackbill, Control of the number of particles in fluid and mhd particle-in-cell methods, Computer Physics Communications 87 (1995) 139–154.

[31] F. Assous, T. Pougeard Dulimbert, J. Segré, A new method for coalescing particles in pic codes, Journal of Computational Physics 187 (2) (2003) 550–571.