Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 187 (2003) 550–571
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A new method for coalescing particles in PIC codes

F. Assous *, T. Pougeard Dulimbert, J. Segré

Département de Physique, Théorique et Appliquée, CEA/DAM Ile-de-France, BP12 – F91680 Bruyères-le-Châtel, France

Received 15 October 2001; received in revised form 5 October 2002; accepted 21 January 2003

Abstract

In this paper, a new method for coalescing particles in PIC codes is proposed. This coalescing process conserves the particle and mesh charge and current densities as well, and also the particle energy. Particular attention is devoted to the derivation of a method as general as possible, to be easily extended to other problems. Numerical examples are shown to illustrate the efficiency of the method.

Keywords: Coalescence; PIC codes

1. Introduction

The numerical simulation of charged particle beams or plasma physics phenomena requires methods of solution for the time-dependent Vlasov equation (possibly with source terms), or more generally for an equation modelling the evolution of a plasma, coupled with the Maxwell system, or any kind of equations approximating it, as the Poisson equation, the Darwin equation, etc. In this framework, the particle-in-cell method (PIC) is a widely used method, where the distribution function is approximated in the phase space by a combination of delta functions, hence defining the *macro-particles*. However an intrinsec constraint on the use of PIC codes is the need to control the number and the distribution of the macro-particles.

Why controlling particles in PIC codes?

1. To limit to a "reasonable" level the total number of particles when dealing with source terms: ionization, collisions, re-emission from the boundaries, etc.
2. To enforce a quasi-constant number of particles per cell, for accuracy purpose, when running a PIC code.
   - on a non-uniform mesh or even on an adaptative mesh, required by the modelling of phenomena with large variations of length scales;
   - on a regular mesh when the physics modelled induces large spatial variations in the particle density.

* Corresponding author. Tel.: +33-1-69-26-73-74; fax: +1-69-26-71-06.
E-mail addresses: franck.assous@cea.fr (F. Assous), thierry.pougeard-dulimbert@cea.fr (T. Pougeard Dulimbert), jacques.segre@cea.fr (J. Segré).

0021-9991/03/$ - see front matter © 2003 Published by Elsevier Science B.V.
doi:10.1016/S0021-9991(03)00124-4

3. To reduce the numerical noise by regularly reorganizing the particle distribution.

In order to preserve the physics of the problem, this control has to fulfil some constraints, such as the mass, momentum or energy conservation, or the positivity of the mass and of the energy. In a *particle-in-cell* approach, both of the *particle* and *cell* (or mesh) quantities are concerned by these constraints as well.

Several approaches have been proposed to cope with this problem.

Illner and Rjasanov [3] were concerned with the numerical solution of the spatially homogeneous Boltzmann equation in the velocity space (without space mesh). They proposed an algorithm which consists in reducing N particles into two particles with which weight and velocity are calculated so that the particle mass, momentum, and energy are exactly conserved. An adaptation of this approach is developed in [6] to solve a Vlasov–Poisson model with ionizations terms.

Lapenta and Brackbill, in a first paper [4], developed a one-dimensional method which coalesces two particles into one, or which splits one particle into two, preserving *either* the particle mass and momentum *or* the energy of each component. It can be extended to two or three-dimensional cartesian grids. The use of the spline basis functions allows the conservation of the charge density defined on the grid.

In a following paper [5], the same authors, concerned with fluid simulations, construct another coalescence process which preserves primitive variables on the mesh, but they are not interested in conserving the particle quantities. They coalesce N particles into L with a principle of maximum uniformity, but the conservation on the mesh is not exact in order to enforce the positivity of some of the variables.

In this paper, we propose a new method for coalescing particles in PIC codes. From this process, we deduce a first algorithm of coalescence that conserves – up to an approximation of the same order as a quadrature formula to be chosen – the particle and cell charge and current densities as well, and also the particle energy. We then deduce a second algorithm of coalescence for which the conservations are exact. In both of the cases, the positivity of the mass and of the energy is preserved. Section 2 is devoted to the presentation of the general framework of PIC codes. The principle of a coalescence process is recalled in Section 3, whereas the new coalescence method is introduced in Section 4. Particular attention is devoted to the derivation of a method exposed in a way general enough to be easily extended to other problems. In Section 5, some improvements of our coalescence scheme are described, to address the problems that appear when it is impossible to achieve coalescence. In particular, it leads to introduce a modified algorithm that is exact with regard to the conservations. Some criteria to determine well-adapted quadrature formulae, depending on the applications, and examples are proposed in Section 6, also illustrated by preliminary numerical results. The particular case of the energy conservation is investigated and is treated in Section 7. The resulting global algorithms are supported by numerical examples given in Section 8. Finally, a conclusion is drawn in Section 9.

2. General framework

We consider one or several populations of charged particles (ions, electrons, etc.) moving in a two or three-dimensional bounded domain Ω. The motion of these charged particles can be described in terms of particle distribution functions by the generalized Vlasov equation,

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_\alpha + \frac{q_\alpha}{m_\alpha}(\mathscr{E}(\mathbf{x},t) + \mathbf{v} \times \mathscr{B}(\mathbf{x},t)) \cdot \nabla_{\mathbf{v}} f_\alpha = S, \tag{1}$$

where $f_\alpha(\mathbf{x},\mathbf{v},t)$ denotes the distribution functions. The variables $\mathbf{x}$ and $\mathbf{v}$ are, respectively, the position and the velocity, m_α and q_α denote the mass and the electric charge for each species of particle α, and S is a source term (ionization, collisions, etc.). For the sake of simplicity, we shall consider in the following only

one species of charged particles (with an electric charge q, a mass m, and a distribution function f). However, this work can easily be applied to several species of particles.

The electromagnetic fields $\mathscr{E}(\mathbf{x},t)$ and $\mathscr{B}(\mathbf{x},t)$ are solution of the Maxwell system

$$\frac{\partial \mathscr{E}}{\partial t} - c^2\,\mathbf{curl}\,\mathscr{B} = -\frac{1}{\varepsilon_0}\mathscr{J}, \tag{2}$$

$$\frac{\partial \mathscr{B}}{\partial t} + \mathbf{curl}\,\mathscr{E} = 0, \tag{3}$$

$$\operatorname{div}\mathscr{E} = \frac{1}{\varepsilon_0}\rho, \tag{4}$$

$$\operatorname{div}\mathscr{B} = 0, \tag{5}$$

or of any kind of equations approximating the Maxwell system ((quasi-)electro or magnetostatics, Darwin, etc.).

The charge and current densities ρ and $\mathscr{J}$ are obtained from the solution f of the Vlasov equation according to

$$\rho = q\int_{\mathbb{R}^3} f(\mathbf{x},\mathbf{v},t)\,\mathrm{d}\mathbf{v}, \quad \mathscr{J} = q\int_{\mathbb{R}^3} \mathbf{v} f(\mathbf{x},\mathbf{v},t)\,\mathrm{d}\mathbf{v}. \tag{6}$$

The Vlasov equation is numerically solved by means of a particle method: the distribution function $f(\cdot,\cdot,t)$ is approximated at any time t, by a linear combination of delta functions in the phase space

$$f(\mathbf{x},\mathbf{v},t) \simeq \tilde{f}(\mathbf{x},\mathbf{v},t) = \sum_{k=1}^{N} w_k \delta(\mathbf{x}-\mathbf{x}_k(t))\delta(\mathbf{v}-\mathbf{v}_k(t)), \tag{7}$$

where each term of the sum can be identified with a macro-particle, characterized by its weight w_k, its position $\mathbf{x}_k$ and its velocity $\mathbf{v}_k$, and N is the number of macro-particles. By introducing this particle approximation of f in the expression (6), we obtain a particle approximation of the charge and current densities as follows:

$$\tilde{\rho} = q\int_{\mathbb{R}^3} \tilde{f}(\mathbf{x},\mathbf{v},t)\,\mathrm{d}\mathbf{v} = q\sum_{k=1}^{N} w_k\,\delta(\mathbf{x}-\mathbf{x}_k(t)), \tag{8}$$

$$\tilde{\mathscr{J}} = q\int_{\mathbb{R}^3} \mathbf{v}\tilde{f}(\mathbf{x},\mathbf{v},t)\,\mathrm{d}\mathbf{v} = q\sum_{k=1}^{N} w_k\mathbf{v}_k\,\delta(\mathbf{x}-\mathbf{x}_k(t)), \tag{9}$$

which is a representation of the charge density ρ and the current density $\mathscr{J}$ in terms of delta functions that will be called *particle variables* in the sequel of this paper.

However the resolution of (approximations of) Maxwell's equations is performed by means of a grid method, namely a finite difference, a finite volume or a finite element method. This requires to define values of ρ and $\mathscr{J}$ at the vertices of the mesh used for the "grid" computation of the electromagnetic fields $\mathscr{E},\mathscr{B}$. Let us denote by $\mathscr{T}$ a mesh that approximates the computational domain Ω. We also introduce $\{a_i\}$ the set of vertices of $\mathscr{T}$ and $\lambda_i,\ 1\leqslant i\leqslant n$ the set of the shape functions of dimension n, related to the field approximation. We define the values of the charge and current densities at the nodes of the mesh by:

$$\rho_i = \frac{\int_\Omega \tilde{\rho}(\mathbf{x})\,\lambda_i(\mathbf{x})\,d\mathbf{x}}{\int_\Omega \lambda_i(\mathbf{x})\,d\mathbf{x}} = \frac{q\sum_{k=1}^{N} w_k \lambda_i(\mathbf{x}_k)}{\int_\Omega \lambda_i(\mathbf{x})\,d\mathbf{x}}, \tag{10}$$

$$\mathscr{J}_i, = \frac{\int_\Omega \tilde{\mathscr{J}}(\mathbf{x})\,\lambda_i(\mathbf{x})\,d\mathbf{x}}{\int_\Omega \lambda_i(\mathbf{x})\,d\mathbf{x}} = \frac{q\sum_{k=1}^{N} w_k \mathbf{v}_k \lambda_i(\mathbf{x}_k)}{\int_\Omega \lambda_i(\mathbf{x})\,d\mathbf{x}}. \tag{11}$$

Since $\sum_i \lambda_i = 1$, this projection procedure (the "assignment") preserves the total charge and the total current of particles.

3. Principle of a local coalescence process

As any other method allowing to control the number of particles, a coalescence process can be *global* or *local*.

A *global* approach consists in replacing the set of all the particles in a domain by another set containing a different number of particles, without changing the grid quantities one wants to conserve.

In a *local* approach, the previous procedure is applied cell by cell. One can choose in which element of the mesh the particles are replaced, while preserving the local mesh and/or particle quantities. Following the same arguments as in [5], we have developed a local coalescence method.

Hence, we want to replace the set of N particles by a set of N' particles. In a coalescence process, $N' < N$. In each element T of the mesh $\mathscr{T}$, the set of N_T particles $(w_k, \mathbf{x}_k, \mathbf{v}_k)_{k=1,N_T}$ are replaced by a set of N'_T particles $(w'_l, \mathbf{x}'_l, \mathbf{v}'_l)_{l=1,N'_T}$. In order to preserve the main features of the distribution function, the coalescence process must meet the particle conditions:

- the mass conservation

$$\sum_{l=1}^{N'_T} w'_l = \sum_{k=1}^{N_T} w_k, \tag{12}$$

- the momentum conservation

$$\sum_{l=1}^{N'_T} w'_l \mathbf{v}'_l = \sum_{k=1}^{N_T} w_k \mathbf{v}_k, \tag{13}$$

- and if it is possible, the energy conservation

$$\sum_{l=1}^{N'_T} w'_l |\mathbf{v}'_l|^2 = \sum_{k=1}^{N_T} w_k |\mathbf{v}_k|^2, \tag{14}$$

with the constraint

$$w'_l > 0, \quad l = 1, \ldots, N'_T. \tag{15}$$

Moreover, the coalescence process must not modify the electromagnetic field calculations. The values of the charge and current densities defined on the mesh (that will be called *mesh variables* in the sequel of this paper) must be unchanged. This implies the following mesh conditions deduced from (10) and (11)

$$\sum_{l=1}^{N'_T} w'_l \lambda_i^T(\mathbf{x}'_l) = \sum_{k=1}^{N_T} w_k \lambda_i^T(\mathbf{x}_k), \tag{16}$$

$$\sum_{l=1}^{N'_T} w'_l \mathbf{v}'_l \lambda_i^T(\mathbf{x}'_l) = \sum_{k=1}^{N_T} w_k \mathbf{v}_k \lambda_i^T(\mathbf{x}_k), \tag{17}$$

for each λ_i^T, where λ_i^T denotes the basis functions λ_i such that $\mathrm{supp}(\lambda_i) \cap T \neq \emptyset$.

4. The new coalescence process

As mentioned in the previous section, the new coalescence process that we developed is local. We call it "process" in the sense that it is a "way of constructing" methods for controlling the number of particles. In this paper, we are interested in coalescence methods, that is, when the number of particles in a cell is too large, they are replaced by a lower number of particles while preserving the particle charge, current and energy, and their contributions to the mesh charge and current.

The principle of the method presented hereafter is also general, since it can be applied to two-dimensional triangle or quadrilateral meshes as well as three-dimensional tetrahedral or hexahedral meshes. Moreover, it can be used with spline basis function, even if special developments are required for using high order splines.

In this paper, the method we present was developed to be implemented in a two-dimensional PIC code, where the field solver uses a P^1 finite element method on an unstructured mesh of triangles [1]. Two examples of algorithms deduced from this method will be exposed.

4.1. Charge and current density conservation

The following sections are devoted to the definition of a first coalescence algorithm that conserves the charge and current density. The energy conservation will be addressed in a forthcoming section.

Let us denote by T a chosen fixed triangle of the mesh, and by $\lambda_i,\ i = 1, 2, 3$ the linear basis functions associated to the nodes of the triangle T (by dropping now the ^{T}). As already stated we want to conserve the mesh variables and the particle variables as well. For doing that, the following property will be helpful:

Proposition 4.1. *We assume that* $\sum_{i=1}^{3} \lambda_i(\mathbf{x}) = 1$. *We then have*

- *If* $\sum_{l=1}^{N'} w'_l \lambda_i(\mathbf{x}'_l) = \sum_{k=1}^{N} w_k \lambda_i(\mathbf{x}_k),\ i = 1, 2, 3,$ *then* $\sum_{l=1}^{N'} w'_l = \sum_{k=1}^{N} w_k$.
- *If* $\sum_{l=1}^{N'} w'_l \mathbf{v}'_l \lambda_i(\mathbf{x}'_l) = \sum_{k=1}^{N} w_k \mathbf{v}_k \lambda_i(\mathbf{x}_k),\ i = 1, 2, 3,$ *then* $\sum_{l=1}^{N'} w'_l \mathbf{v}'_l = \sum_{k=1}^{N} w_k \mathbf{v}_k$.

In others words, this property means that if the local coalescence method conserves the mesh charge and current densities, *it conserves the corresponding* particle variables. *Let us thus consider the mesh quantities. In order to construct the coalescence process, one of the principal elements of our method is the use of the following remark*:

Remark 4.1. Following [7], a particle approximation of a function can be related to a numerical quadrature formula.

Indeed, let $g(\mathbf{x})$ be a given function, and consider a N-points quadrature formula on T as follows:

$$\int_T g(\mathbf{x})\,\mathrm{d}\mathbf{x} \simeq \sum_{k=1}^{N} \alpha_k g(\boldsymbol{\xi}_k), \tag{18}$$

where $\boldsymbol{\xi}_k$ and α_k denote the quadrature points and weights. Using (18), we have

$$\int_T f(\mathbf{x})\psi(\mathbf{x})\,\mathrm{d}\mathbf{x} \simeq \sum_{k=1}^{N} \alpha_k f(\boldsymbol{\xi}_k)\psi(\boldsymbol{\xi}_k) \quad \forall \psi \text{ a regular test function},$$

which defines a "measure" approximation (see [7] for a precise definition) of f as

$$f(x) \simeq \sum_{k=1}^{N} w_k \delta(\mathbf{x} - \boldsymbol{\xi}_k),$$

where the weights w_k are given by

$$w_k = \alpha_k f(\boldsymbol{\xi}_k).$$

4.2. Construction of the position and the weight of the new particles

The construction of the position and weight of the new particles can now be described in three steps:

1. Find a function g defined on T that satisfies:

$$\int_T g(\mathbf{x})\lambda_i(\mathbf{x})\,\mathrm{d}\mathbf{x} = \sum_{k=1}^{N} w_k \lambda_i(\mathbf{x}_k) \stackrel{\text{def}}{=} Q_i, \quad i = 1,2,3, \tag{19}$$

where N, w_k and $\mathbf{x}_k$ are the number, the weights and the positions of the particles to be replaced.

2. Choose on the triangle an accurate N'-points quadrature formula characterized by (x'_l, α_l), $l = 1, \ldots, N'$, its quadrature points and weights. N' and x'_l will be the number and the positions of the new particles, defined in the triangle T.
3. Define the weights of the new particles by applying the quadrature formula, namely

$$\int_T g(\mathbf{x})\lambda_i(\mathbf{x})\,\mathrm{d}x \simeq \sum_{l=1}^{N'} \alpha_l g(\mathbf{x}'_l)\lambda_i(\mathbf{x}'_l), \quad i = 1,2,3,$$

so we have

$$w'_l = \alpha_l g(\mathbf{x}'_l).$$

With this choice, the mesh charge density conservation property is ensured, up to a precision level depending on the quadrature formula, that is

$$\sum_{l=1}^{N'} w'_l \lambda_i(\mathbf{x}'_l) \simeq \sum_{k=1}^{N} w_k \lambda_i(\mathbf{x}_k), \quad i = 1,2,3,$$

and consequently, by using Proposition 4.1, the particle charge density is also conserved, with the same precision.

Remark 4.2. It can be easily proved that this approximation is *of the same order as* the quadrature formula approximation applied to $\int_T g(\mathbf{x})\lambda_i(\mathbf{x})\,\mathrm{d}\mathbf{x}$. In a forthcoming section (see Section 5), a modified approach that leads to an exact method of coalescence will be also introduced.

It remains now to choose a well-adapted function $g(\mathbf{x})$. Obviously, there is several different possible choices. As $g(\mathbf{x})$ must satisfy three conditions defined by (19), it is natural to introduce three unknowns in its definition, so we choose

$$g(\mathbf{x}) = \sum_{j=1}^{3} g_j \lambda_j(\mathbf{x}),$$

where the three coefficients g_j are the solutions of the 3×3 linear system

$$\sum_{j=1}^{3} g_j \int_T \lambda_j(\mathbf{x})\lambda_i(\mathbf{x})\,\mathrm{d}x = Q_i, \quad i = 1,2,3 \tag{20}$$

$$\Rightarrow g_j = \frac{3}{|T|}\left(3Q_j - \sum_{i\neq j} Q_i\right), \quad j = 1,2,3, \tag{21}$$

where $|T|$ denotes the area of the triangle T, and

$$w'_l = \alpha_l \sum_{j=1}^{3} g_j \lambda_j(\mathbf{x}'_l). \tag{22}$$

This solution does not necessarily ensure the positivity of the particle weights w'_l, which is a condition that *must* be fulfiled. To ensure this positivity, a sufficient condition consists in choosing a quadrature formula with positive weights ($\alpha_l > 0\ \forall l$) and to make sure that $g(\mathbf{x}) > 0\ \forall \mathbf{x} \in T$. This yields the following positivity condition:

$$3Q_j - \sum_{i\neq j} Q_i > 0, \quad j = 1,2,3. \tag{23}$$

It can exist some cases where the positivity conditions are not fulfiled. Indeed, the quantities $(Q_j)_{j=1,2,3}$ depend on the initial repartition of the particles in the triangle before coalescence. Typical situations which do not meet these criteria are depicted on Fig. 1, where particles with the same weights are considered in a triangle with a local numerotation of its nodes.

In the first case (Fig. 1(a)), most of the particles have clustered together near the node 3, we then have $Q_3 \gg Q_1$ and $Q_3 \gg Q_2$. In the second illustration (see Fig. 1(b)), the particles have clustered together near the edge (2-3). In such a situation, $Q_1 \ll Q_2$ and $Q_1 \ll Q_3$. In both situations, the relations (23) are violated which prevents the coalescence from being performed.

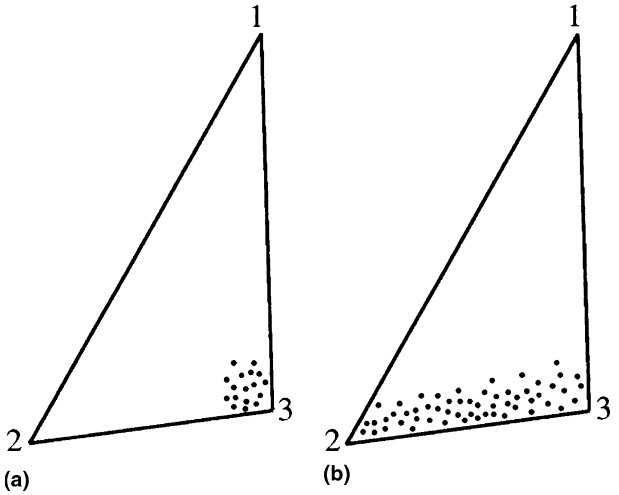

Fig. 1. Two cases which do not meet the positivity conditions.

In order to enlarge the range of application of the coalescence, a modification of this method will be proposed (see Section 5) that allows us to improve the process by relaxing the positivity conditions (23).

4.3. Construction of the velocities of the new particles

The velocities of the new particles are constructed component by component in the same way as the positions and the weights. This construction can be divided into three steps. For each component ξ:

1. Find a function h^ξ defined on the triangle T which satisfies

$$\int_T h^\xi(\mathbf{x})\lambda_i(\mathbf{x})\,d\mathbf{x} = \sum_{k=1}^{N} w_k v_k^\xi \lambda_i(\mathbf{x}_k) \stackrel{\text{def}}{=} I_i^\xi, \quad i = 1,2,3. \tag{24}$$

2. Use the same quadrature formula as the one used for the charge conservation.
3. Define the velocity of the new particles with the quadrature formula

$$\int_T h^\xi(\mathbf{x})\lambda_i(\mathbf{x})\,d\mathbf{x} \simeq \sum_{l=1}^{N'} \alpha_l h^\xi(\mathbf{x}'_l)\lambda_i(\mathbf{x}'_l), \quad i = 1,2,3.$$

So, using the weight definition established when dealing with the charge density conservation

$$w'_l = \alpha_l g(\mathbf{x}'_l),$$

we define

$$v_l'^\xi = \frac{h^\xi(\mathbf{x}'_l)}{g(\mathbf{x}'_l)}, \tag{25}$$

which leads to

$$\sum_{l=1}^{N'} w'_l v_l'^\xi \lambda_i(\mathbf{x}'_l) \simeq \sum_{k=1}^{N} w_k v_k^\xi \lambda_i(\mathbf{x}_k), \quad i = 1,2,3,$$

i.e., the mesh current density is conserved, up to an approximation of the same order as the quadrature formula we will choose. Using again Proposition 4.1, the particle current density is conserved as well, with the same precision.

Following the same arguments as in the previous case, we choose for the function h^ξ:

$$h^\xi(\mathbf{x}) = \sum_{j=1}^{3} h_j^\xi \lambda_j(\mathbf{x}).$$

By inverting this system, one obtains

$$h_j^\xi = \frac{3}{|T|}\left(3I_j^\xi - \sum_{i\neq j} I_i^\xi\right), \quad j = 1,2,3. \tag{26}$$

Remark that, in this case, there is no sign condition for the coefficients h_j^ξ.

Remark 4.3. At first sight, it would seem in the new particle velocities (25), that the denominator $g(\mathbf{x}'_l) \stackrel{\text{def}}{=} \sum_{j=1}^{3} g_j \lambda_j(\mathbf{x}'_l)$ may be quite small, which could lead to create particles with very large velocities:

Consider for instance a new particle created near a cell vertex, then all the λ_j except one would be small. If the g_j corresponding to the finite λ_j is also tiny, then the sum in the denominator can be also small. Moreover, there is no a priori reason why the numerator must also be small.

In practice, however, this situation does not occur by choosing quadrature formulae with integration points "not too close" to the cell vertices. In particular, "good" quadrature formulae are such that all their integration points are different from the mesh vertices (see examples in Section 6).

To conclude this section, let us summarize the main features of the new coalescence method

- N particles in the current triangle are replaced by N' particles, due to the choice of the quadrature formula, located in the same triangle.
- The locations of the new particles are defined as the integration points of the chosen quadrature formula.
- The weights of the new particles are defined by

$$w'_l = \alpha_l \sum_{j=1}^{3} g_j \lambda_j(\mathbf{x}'_l)$$

- and the velocity of the new particles are

$$v'^{\xi}_l = \frac{\sum_{j=1}^{3} h^{\xi}_j \lambda_j(\mathbf{x}'_l)}{\sum_{j=1}^{3} g_j \lambda_j(\mathbf{x}'_l)},$$

where

$$g_j = \frac{3}{|T|}\left(3Q_j - \sum_{i\neq j} Q_i\right), \quad j = 1,2,3 \tag{27}$$

and

$$h^{\xi}_j = \frac{3}{|T|}\left(3I_j - \sum_{i\neq j} I_i\right), \quad j = 1,2,3. \tag{28}$$

The algorithm of the coalescence process is then, for every triangle T:

```
If N > N_max, the maximum number of particles per triangle
  computation of g_j
  if g_j > 0,  j = 1,2,3
     coalescence
  endif
endif
```

5. How to enhance the coalescence process

5.1. Relaxing the positivity condition (23)

The impossibility to achieve coalescence with positive weights when the Q_j are strongly non-uniform (see (23)) is mainly caused by the particle created on the "central points" of the used quadrature formula. Indeed, their contributions to the three nodes of the triangle T are almost equal which prevents the quantities on these nodes from being very different. An obvious remedy consists in choosing other functions

g and h^{ξ}, which will eliminate these central particles. Obviously, there are a lot of possible choices. We describe now the one we have chosen, following with these new functions exactly the same methodology as developed in Section 4.2.

Assume first that finer triangles $t_j,\ 1 \leqslant j \leqslant 4$ are defined by dividing each triangle T into four triangles of equal surface, locally numbered as shown in Fig. 2. We then define the functions $\mu_j,\ j = 1, 2, 3$, the support of which being restricted to each triangle t_j such that

$$\mu_j(\mathbf{x}) = 2\lambda_j(\mathbf{x}) - 1, \quad \mathbf{x} \in t_j, \tag{29}$$

$$\mu_j(\mathbf{x}) = 0, \quad \mathbf{x} \in t_i,\ i \neq j, \tag{30}$$

where $\lambda_j(\mathbf{x})$ are the shape functions previously introduced (see for instance (10) and (11)). The function g that must satisfy the three conditions (19) is now defined as

$$g(\mathbf{x}) = \sum_{j=1}^{3} g_j \mu_j(\mathbf{x})$$

and therefore the coefficients g_j are solutions of the linear system

$$\sum_{j=1}^{3} g_j \int_T \mu_j(\mathbf{x})\lambda_i(\mathbf{x})\,\mathrm{d}x = Q_i, \quad i = 1, 2, 3, \tag{31}$$

from which one can deduce the new positivity conditions as

$$7Q_j - \sum_{i \neq j} Q_i > 0, \quad j = 1, 2, 3,$$

which are far less restrictive than the previous ones (23).

In this modified coalescence process, the initial N particles per triangle are replaced with N'' particles per triangle, and the corresponding weights are thus equal to

$$w'_l = \alpha_l \sum_{j=1}^{N''} g_j \mu_j(\mathbf{x}'_l).$$

Actually, when we use a N'-points quadrature formula, N'' of these points (see Fig. 2) are located *outside* the central sub-triangle t_4. Because we have

$$\mathrm{supp}(\mu_j) \cap t_4 = \emptyset,$$

this choice of g eliminates the $N' - N''$ central particles.

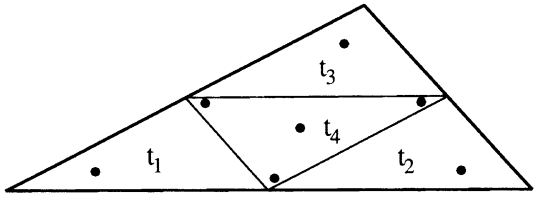

Fig. 2. Definition of the finer triangles t_j.

For the velocities, we also use the same μ_j and we set

$$h^{\xi}(\mathbf{x}) = \sum_{j=1}^{3} h_j^{\xi} \mu_j(\mathbf{x}).$$

The straight use of this method has, however, shown results with a quite poor accuracy on the conservation of the mesh variables (and also on the particle variables), with a relative error of $\simeq 10^{-2}$, due to the fact that (see Remark 4.2) the integral

$$\int_T \mu_j(\mathbf{x})\lambda_i(\mathbf{x})\,\mathrm{d}\mathbf{x} \quad \forall i,j \tag{32}$$

that appears in Eq. (31) is poorly approximated by

$$\sum_{l=1}^{N''} \alpha_l \mu_j(\mathbf{x}'_l)\lambda_i(\mathbf{x}'_l) \quad \forall i,j, \tag{33}$$

because the function $\mu_j(\mathbf{s})\lambda_i(\mathbf{s})$, for $\mathbf{s} \in T$, is far from a polynomial. The forthcoming section is devoted to a solution of this difficulty.

5.2. An exact coalescence process

A remedy to overcome this poor approximation is now to replace the g_j by a new set of coefficients γ_j, which will be obtained by using consistently the approximation (33) rather than the exact value of (32). More precisely, we define the coefficients γ_j as the solutions of the system

$$\sum_j \gamma_j \sum_{l=1}^{N''} \alpha_l \mu_j(\mathbf{x}'_l)\lambda_i(\mathbf{x}'_l) = Q_i, \quad i = 1,2,3, \tag{34}$$

and we choose the weights as

$$w'_l = \alpha_l \sum_{j=1}^{3} \gamma_j \mu_j(\mathbf{x}'_l), \quad l = 1,2,3. \tag{35}$$

With this choice of coefficients, we have made better than just improve the approximation. Indeed, one can prove the following property

Property 5.1. *When using the function* $g(\mathbf{x}) = \sum_{j=1}^{3} \gamma_j \mu_j(\mathbf{x})$, *with the* γ_j *solving the system* (34), *the coalescence process conserves the mesh charge exactly*.

Proof 5.1. Using (35), one has that

$$\sum_l w'_l \lambda_i(\mathbf{x}'_l) = \sum_l \alpha_l \sum_j \gamma_j \mu_j(\mathbf{x}'_l)\lambda_i(\mathbf{x}'_l) = \sum_j \gamma_j \sum_l \alpha_l \mu_j(\mathbf{x}'_l)\lambda_i(\mathbf{x}'_l),$$

which, by construction equal to Q_i, implies

$$\sum_l w'_l \lambda_i(\mathbf{x}'_l) = \sum_k w_k \lambda_i(\mathbf{x}_k). \qquad \square$$

Remark 5.1. This method is illustrated here with the special case treated in Section 5.1. It is, however, a general procedure that consists in replacing the system of the form (20) by a system similar to (34). We apply also this technique to the conservation of the momentum. Thus we have defined an alternate coalescence process which conserves exactly the mesh and the particle mass and momentum.

6. Quadrature formulae and applications

6.1. Choice of quadrature formulae

The choice of a quadrature formula depends strongly on the physical applications one wants to model.

Hence, if we are concerned by kinetic PIC of plasma physics, often with a lot of particles per cell (tens, hundreds, or even thousands), we have to use a quadrature formula with also a "sufficiently large" number of integration points per cell (some tens), in order to exchange a set of particles for another that is "comparable". Otherwise the alteration of the particle distribution function, that plays an important role in kinetic PIC (see [5]), would lead to a significative error on the results.

On the contrary, in fluid applications or in most of kinetic modellings of particle beams, where the number of particles per cell is often around a few tens, a coalescence method that reduces the number of particles in a cell to some units can be sufficient.

Moreover, in all the cases, the quadrature formula has to fulfil the following requirements:

1. The weights have to be positive, because of the particle approximation.
2. The order of approximation has to be sufficient, because it is directly related to the precision of the first method of coalescence (see Remark 4.2), but not for the exact method of Section 5.
3. The integration points must be different from the mesh vertices, as explained in Remark 4.3.

Trying to take into account all these parameters, the approach we propose consists in using "family" of quadrature methods, in which the number of integration points is a parameter. One can then choose the number of particles per cell after coalescence one wants.

The first illustration we used in the sequel of this paper is a 7-points Gauss–Hammer quadrature formula (see Fig. 3). It is a sixth-order quadrature formula with positive weights, and the quadrature is therefore exact for $\int_T \lambda_j(\mathbf{x})\lambda_i(\mathbf{x})\,d\mathbf{x}$ (see for instance [8]). Moreover, these 7 points are located *in* the triangle, which allows the coalescence to remain local. For more details on the numerical values of the weights and of the positions of these points, the reader can refer (for instance) to [8]. This quadrature formula is obviously well designed for fluid applications.

For kinetic PIC situations, among the other possible choices, we propose to use the Gauss–Radau formula, as depicted in Fig. 4, even if the accuracy order is not optimal with regards to the number of integration points. This number of integration points (9, 16, 25, ...) – that is the number of particles per cell after coalescence – can be chosen as large as we want, while the weights are always positive, that allows to treat easily kinetic PIC problems. Finally, remark that the choice of any admissible quadrature formula can be a way to control a priori the number of particles after coalescence.

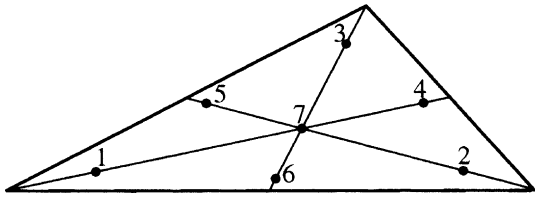

Fig. 3. Location of the 7 points in the Gauss–Hammer quadrature formula.

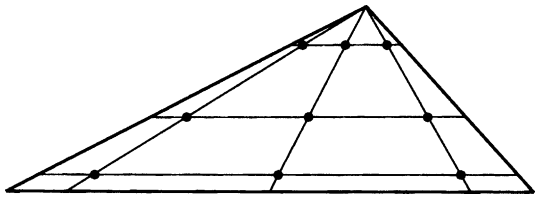

Fig. 4. Location of the integration points in the 9-points Gauss–Radau quadrature formula.

6.2. Numerical test for the charge and current density conservation

As an illustration, we present some results obtained by using the algorithm constructed from the 7-points Gauss-Hammer formula. We consider a mesh made up with 512 triangles. On every triangle, 30 particles are created. Their positions are chosen at random uniformly on the triangle and their velocities according to a Maxwellian distribution.

The results after coalescence show a relative error on the mesh and on the particle charge and current densities conservation of 10^{-8} for a single precision computation. The number of particles before coalescence is 15,360, the number after is 4389 and there are 35 triangles where the coalescence is forbidden.

Remark 6.1. Following Remark 4.2, the accuracy of this process is only related to the number of digits retained to define the positions and the coefficients of the quadrature formula, because the integration of the functions $\lambda_i\lambda_j$ is exact with the 7-points Gauss–Hammer quadrature formula.

To improve the efficiency of the coalescence, we use now, on the same example as previously, the modified coalescence algorithm, that is with only three particles when the tests associated to the seven particles fail. In this case, the positivity condition (23) is relaxed, and we obtain using the exact scheme of Section 5.2

$$C_{\mathrm{p}}Q_i - \sum_{i\neq j} Q_j > 0, \quad i = 1,2,3,$$

where the constant $C_{\mathrm{p}} \simeq 8.87$. We obtain the following conclusions: The number of particles before coalescence is 15,360, the number after is 3444 and there is no triangle where the coalescence is impossible.

Remark 6.2. Because three particles *in each* element of a mesh is not sufficient to ensure a non-noisy particle method, even in a fluid application, this improvement is used only if the general approach fails. As shown by the numerical results, it occurs in only 35 triangles compared to 478, that is less than 7%, so that the precision of the particle method is not significantly worse when using this technique for relaxing the positivity condition.

7. Energy conservation

At first sight, it could appear not necessary to ensure the conservation of the particle energy, this quantity being not required in the global coupled model. However, the following numerical test (see Fig. 5) shows that, without particle energy conservation, the time variation of the computed electric potential in an electron beam ionizing a background gas can be very noisy. More generally, it is well known that conserving the particle energy is of prime importance when modelling plasma physics phenomena. In these conditions, we have now to investigate the energy conservation.

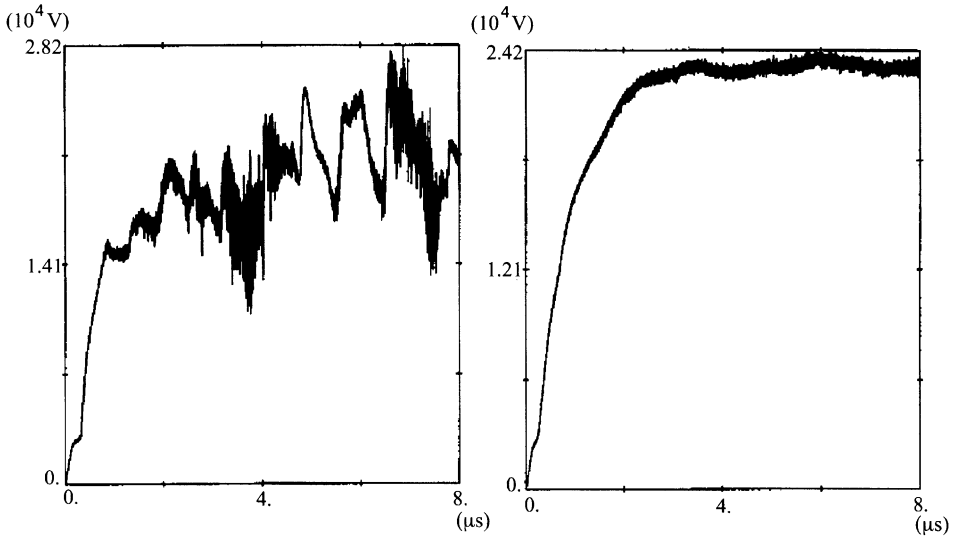

Fig. 5. Computed electric potential with and without particle energy conservation.

We first consider ϵ, the particle energy before coalescence, defined by

$$\epsilon = \sum_{k=1}^{N} w_k |\mathbf{v}_k|^2,$$

and also introduce ϵ' the particle energy after coalescence, as

$$\epsilon' = \sum_{l=1}^{N'} w'_l |\mathbf{v}'_l|^2.$$

The coalescence processes that we have constructed for ρ and $\mathscr{J}$ do not satisfy a priori the particle energy conservation, that is

$$\epsilon = \epsilon'.$$

To remedy this problem, two approaches have been tried:

1. introduce a correction in the ρ and $\mathscr{J}$ coalescence scheme in order to achieve the energy conservation;
2. construct a global new coalescence scheme which satisfies the energy conservation.

The following two sections are devoted to the presentation of these two approaches.

7.1. An energy correction

This correction is inspired by the Illner and Rjasanov [3] method of coalescence.

Let us consider one of the particles obtained after a coalescence process, characterized by $(w'_{l_0}, \mathbf{x}'_{l_0}, \mathbf{v}'_{l_0})$. The principle of this approach consists in splitting it into two particles $(w'_{l_1}, \mathbf{x}'_{l_1}, \mathbf{v}'_{l_1})$ and $(w'_{l_2}, \mathbf{x}'_{l_2}, \mathbf{v}'_{l_2})$, with the same positions, and with their weights and velocities chosen in such a way that the initial mass and momentum are conserved. Moreover, the lack of energy due to the coalescence process, i.e., $\Delta\epsilon \stackrel{\text{def}}{=} \epsilon - \epsilon'$, must be compensated by a well adapted choice of the characteristics of the new particles. These conditions can be summarized with the following equations (dropping the $'$ and the $_l$):

$$\mathbf{x}_1 = \mathbf{x}_2 = \mathbf{x}_0, \tag{36}$$

$$w_1 + w_2 = w_0, \tag{37}$$

$$w_1\mathbf{v}_1 + w_2\mathbf{v}_2 = w_0\mathbf{v}_0, \tag{38}$$

$$w_1|\mathbf{v}_1|^2 + w_2|\mathbf{v}_2|^2 = w_0|\mathbf{v}_0|^2 + \Delta\epsilon. \tag{39}$$

This system obviously is not necessarily well posed, the existence and uniqueness of solution being not ensured. Following [3], we can fix several degrees of freedom by choosing

$$w_1 = w_2 = \frac{w_0}{2}, \tag{40}$$

$$\mathbf{v}_1 = \mathbf{v}_0 - \beta\mathbf{u}, \quad \mathbf{v}_2 = \mathbf{v}_0 + \beta\mathbf{u}, \tag{41}$$

where β is a parameter to determine and $\mathbf{u} \in S^2$, the unit sphere of $\mathbb{R}^2$. With (40) and (41), the mass and momentum conservations (37) and (38) are satisfied. Now *if and only if* the variation of the energy is positive, i.e.,

$$\Delta\epsilon > 0, \tag{42}$$

the energy conservation relation is fulfiled by taking

$$\beta = \sqrt{\frac{\Delta\epsilon}{w_0}}$$

as the solution of (39).

Remark 7.1. The unit vector $\mathbf{u}$ being an additional degree of freedom, we choose arbitrarily to take $\mathbf{u} = \bar{\mathbf{v}}/|\bar{\mathbf{v}}|$, with $\bar{\mathbf{v}} \stackrel{\text{def}}{=} \sum_l w'_l\mathbf{v}'_l / \sum_l w'_l$.

It remains now to choose which particle to split. In order to obtain resulting particles with velocities not too different from the main velocity, we have to minimize β and therefore, we split the particle with the maximum weight w_0. When dealing with a coalescing process with few particles after coalescence, we can split all of them and then make use of the energy conservation scheme to increase the number of particles after coalescence.

A drawback of this method is the positivity condition (42) on the sign of the energy variation that can be easily tested but never be controlled. Another drawback is that, if $\Delta\epsilon$ is "too important", the velocities of the split particles may be largely out of the range of the velocities of the other particles, leading to a particle distribution function very different from the initial one. For these reasons, it is worthwhile to study now an alternate method, that is described as follows.

7.2. A coalescence scheme satisfying the energy conservation

The principle of this approach is no more to *correct* the energy after performing the coalescence, but to propose a *modification* of the global coalescence scheme, that yields velocities after coalescence verifying the particle conservation of the energy, that is

$$\sum_l w'_l |\mathbf{v}'_l|^2 = \sum_k w_k |\mathbf{v}_k|^2,$$

while preserving the mesh conservation of the momentum

$$\sum_l w'_l \mathbf{v}'_l \lambda_i(x'_l) = \sum_k w_k \mathbf{v}_k \lambda_i(x_k),$$

the mesh conservation of the mass being unchanged. For doing this, we start from the conservative coalescence scheme described in Section 5.2. We add in the definition of the velocities a new degree of freedom β that allows us to enforce the energy conservation. Hence, using the same notations as in Section 4.3, we define for each component ξ

$$h^\xi(\mathbf{x}) = \sum_{j=1}^{3} h_j^\xi \lambda_j(\mathbf{x}) + \beta u_\xi v(\mathbf{x}),$$

where u_ξ is the ξth component of an arbitrary unit vector $\mathbf{u}$, which can be chosen as in Remark 7.1, and $v(\mathbf{x})$ is a function locally defined on every element T. Then in the same spirit as (34), the momentum conservation (for the ξth component) reads

$$\sum_{l=1}^{N'} \alpha_l \left(\sum_{j=1}^{3} h_j^\xi \lambda_j(\mathbf{x}'_l) + \beta u_\xi v(\mathbf{x}'_l) \right) \lambda_i(\mathbf{x}'_l) = I_i^\xi, \quad i = 1,2,3. \tag{43}$$

The unknowns h_j^ξ and β can be determined by supplementing the system (43) with a second degree equation for the energy conservation. An easy way to simplify this rather complex system is to introduce a pointwise definition of the function v such as

$$\sum_{l=1}^{N'} \alpha_l v(\mathbf{x}'_l) \lambda_i(\mathbf{x}'_l) = 0, \quad i = 1,2,3. \tag{44}$$

Doing this, the equations of the momentum and energy conservation are no more coupled, so that the h_j^ξ are the solutions of a system similar to (34). With this choice, the velocities of the particles after coalescence can be defined as (see Section 4.3)

$$\widetilde{v_l'^\xi} = \alpha_l \frac{\sum_{j=1}^{3} h_j^\xi \lambda_j(\mathbf{x}'_l) + \beta u_\xi v(\mathbf{x}'_l)}{w'_l}$$

that can be written as follows:

$$\widetilde{v_l'^\xi} = v_l'^\xi + \beta u_\xi \frac{\alpha_l}{w'_l} v(\mathbf{x}'_l). \tag{45}$$

Remark 7.2. With (45), this method can be viewed as a perturbation of the initial coalescence process in which the velocities of the new particles are corrected by an additive term in order to conserve the energy.

The energy conservation equation

$$\sum_{l=1}^{N'} w'_l \sum_\xi \left(\widetilde{v_l'^\xi} \right)^2 = \sum_{k=1}^{N} w_k \sum_\xi \left(v_k^\xi \right)^2 = \epsilon,$$

together with (45), allows us to characterize β as a solution of the second degree equation

$$\sum_{l=1}^{N'} \frac{\alpha_l^2}{w'_l} v(\mathbf{x}'_l)^2 \beta^2 + \sum_{\xi} \sum_{l=1}^{N'} v_l'^{\xi} \alpha_l v(\mathbf{x}'_l) \beta - \Delta\epsilon = 0, \tag{46}$$

where $\Delta\epsilon$ is the initial energy conservation error, as defined in Section 7.1. The conservation of the energy is therefore enforced as the discriminant of (46) δ_β is non-negative, which is a condition weaker than the condition $\Delta\epsilon > 0$, required by the first energy conservation scheme.

It remains now to construct $v(\mathbf{x}'_l)$ satisfying the conditions (44), that is nothing but orthogonality conditions (in a space of dimension N') between the vector $\boldsymbol{v}$ of components $v(\mathbf{x}'_l)$, and each of the three vectors $\boldsymbol{\Lambda}_i$ of components $\alpha_l \lambda_i(\mathbf{x}'_l)$. It is achieved through a Gram–Schmidt algorithm.

Remark 7.3. This method of energy conservation can obviously be applied in a similar way when the "central particles" are eliminated.

In practice, this second method of energy conservation should be preferred, first because of the larger domain of application, but more importantly since it far less modifies the initial particle distribution, as it will be shown in the following section. Remark also that, in both of the methods, no coalescence is performed when the conditions ($\Delta\epsilon > 0$ or $\delta_\beta \geqslant 0$) are not fulfiled, even if in practice, these situations almost never occur. Nevertheless, this limitation of the method allows us to prevent in all cases some numerical heating or cooling of the system.

It remains now to evaluate the performances and the precision of this coalescence scheme in some examples of numerical applications. This is the aim of the last section.

8. Numerical results

We define a test inspired from Forrester in [2, Chapter 3, p. 46]. We consider a plane diode and prescribed fixed values of the potential on the electrodes, i.e., $V = 0$ on the anode and $V = -V_a$ on the cathode. The diode is filled with ions which are generated at a uniform rate per unit volume. Following Forrester, we consider the "critical rate" which ensures that the electric field $\mathscr{E}$ vanishes on the anode.

This one-dimensional situation is treated as a two-dimensional test case. At each time-step, macro-particle are created in every element of the mesh, in order to take into account the generation process. The mesh is made up with 1000 triangles and we run the computation during 1000 time-steps which is enough to obtain a stationary solution.

In a first case, we model the generation of ions by creating at each time-step *one* macro-particle in every element of the mesh. We then compare the results obtained without coalescence and those with a 7-points coalescence scheme. The coalescence is performed in a mesh element as soon as the number of particles becomes greater than 20. We obtain:

- number of particles without coalescence: 35,588;
- number of particles with coalescence: 14,715;
- number of situations where the coalescence is impossible, due to the weights positivity constraint: 0;
- number of situations where the coalescence is impossible, due to the energy conservation constraint: 0.4% of the total number of coalescence attempts.

Fig. 6 represents the ion current intensities through the cathode (without and with coalescence) and Fig. 7 the respective charge density profiles between the electrodes. In both cases the mean values of these quantities are the same with and without coalescence, but the results are noisier in the first case. Two reasons explain that the coalescence induces noise:

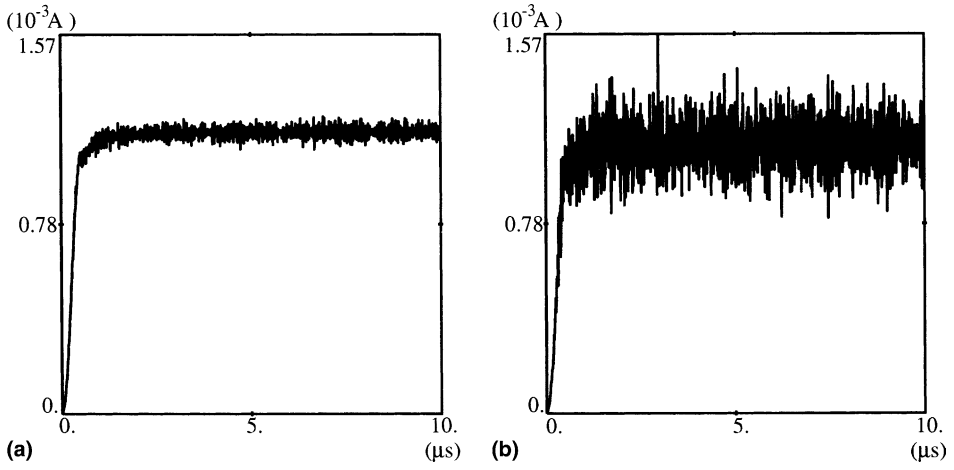

Fig. 6. Ion current intensity through the cathode.

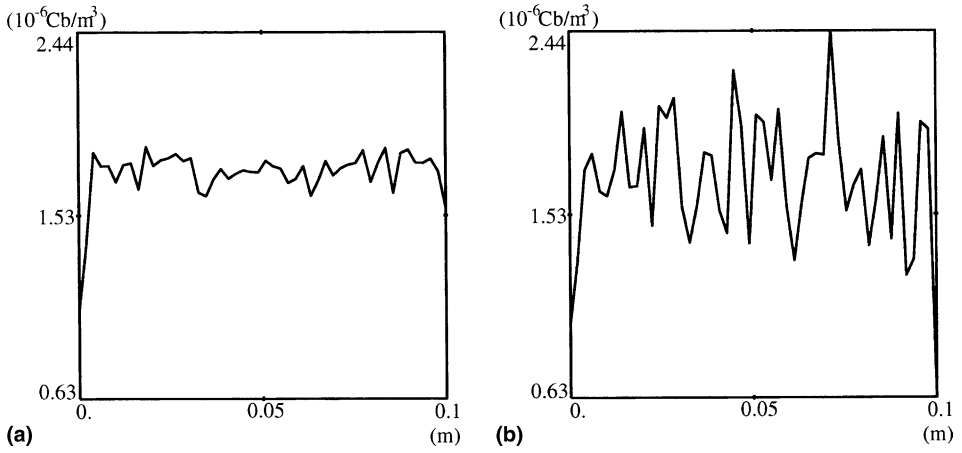

Fig. 7. Charge density profile between the electrodes.

1. the noise increases when the number of particles decreases,
2. the weights of the whole set of particles are less uniform after several coalescence steps than without coalescence.

In order to determine the relative parts of both reasons, we perform a test without coalescence, but with half of the particles (more precisely 17,565) that is *comparable* to the with-coalescing simulation. Fig. 8 shows the density and the current profiles in this case. We can conclude that the non-uniformity of the particle weights is the major cause of the noise induced by the coalescence.

This noise increase is, however, without any consequence on the field calculations, as it is shown in Fig. 9, where we note that the potential distribution is exactly the same with (Fig. 9(a)) or without (Fig. 9(b)) coalescence.

We can lessen the coalescence induced noise by defining a new strategy which will decrease the discrepancy in the particle weights. One way to achieve this is to perform the coalescence process *all* N_c

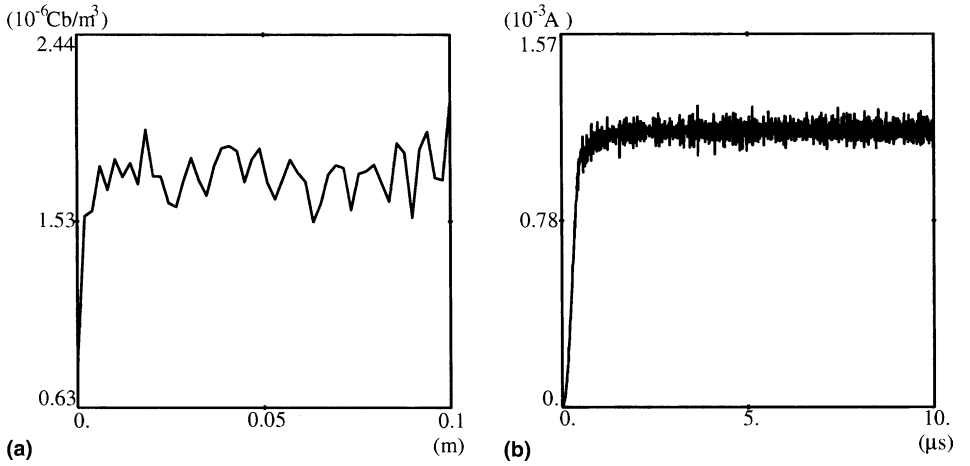

Fig. 8. Charge density profile (a) and ion current intensity (b) without coalescnece and with a reduced number of particles.

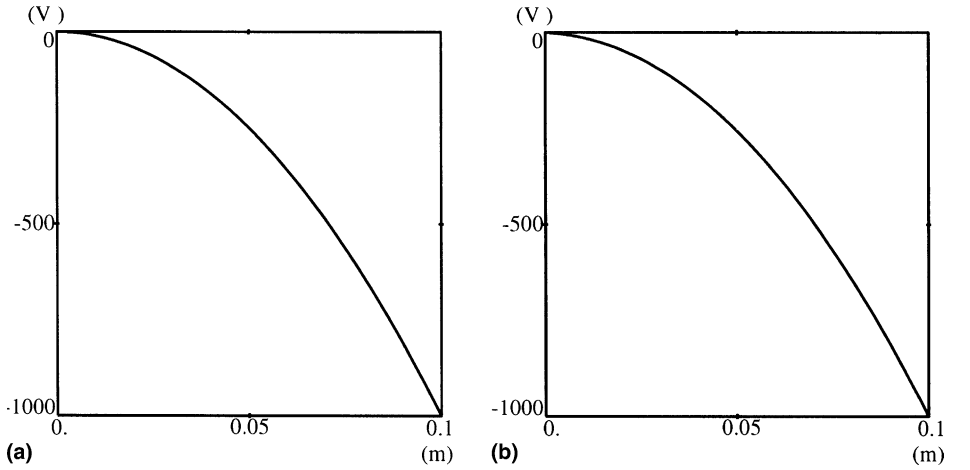

Fig. 9. Potential distribution with and without coalescence.

time-steps (N_c denotes a number of time-steps where the coalescence is not performed) *on every* mesh element rather than coalescing every time-step on those elements where the number of particles is greater than a given maximum number. This method can therefore be viewed as a *global* coalescence based on a *local* procedure.

We apply this technique on the same test case as previously, with $N_c = 50$. The total number of particles in the computation domain varies from 7800 to 31,700. We obtained a potential curve identical to the one obtained in Fig. 9, but the density profile and the current intensity (see Fig. 10) are now far less noisy than before. Compared to the initial results without coalescence the remaining extra noise can now be assigned to the lower total number of particles.

We now consider another test case intended to measure the effect of coalescence on the particle distribution function. We use the same mesh as previously, and we approximate a Maxwellian distribution by

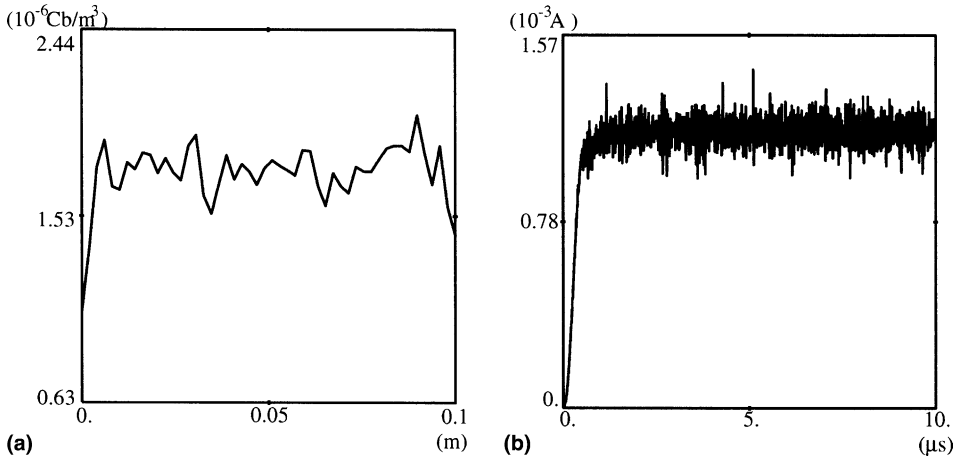

Fig. 10. Charge density profile (a) and ion current intensity (b) with the new strategy.

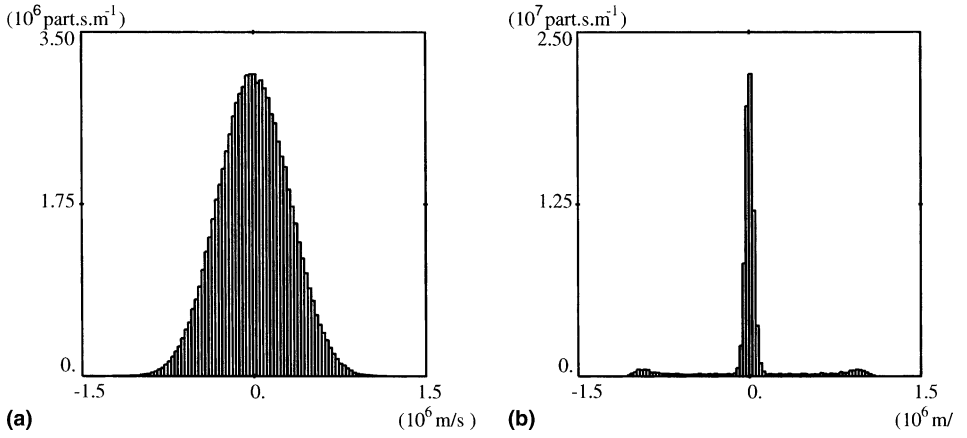

Fig. 11. Velocity distribution function without coalescence (a) and after coalescing with the first energy conservation scheme (b).

injecting 250 particles per cell. We then perform coalescence in the whole domain with a 25-particle scheme, that means with 25 particles in each cell after a coalescence step.

We present in Fig. 11(a) the initial distribution function, and in Fig. 11(b) the one obtained after coalescing with the first energy conservation scheme, described in Section 7.1 (note that the scales are different). We remark that this method introduces a bias corresponding to the particles used to correct the energy.

Figs. 12(a) and (b) show, respectively, the same initial distribution function and the one obtained after coalescing with the second energy conservation scheme (see Section 7.2). Here, the coalescence does not deteriorate the main characteristics of the distribution function.

For all the cases, the numerical relative precision obtained on a double precision machine (CRAY T90) for the conservation of mass, momentum and energy is about 10^{-14} which corresponds to the round off error.

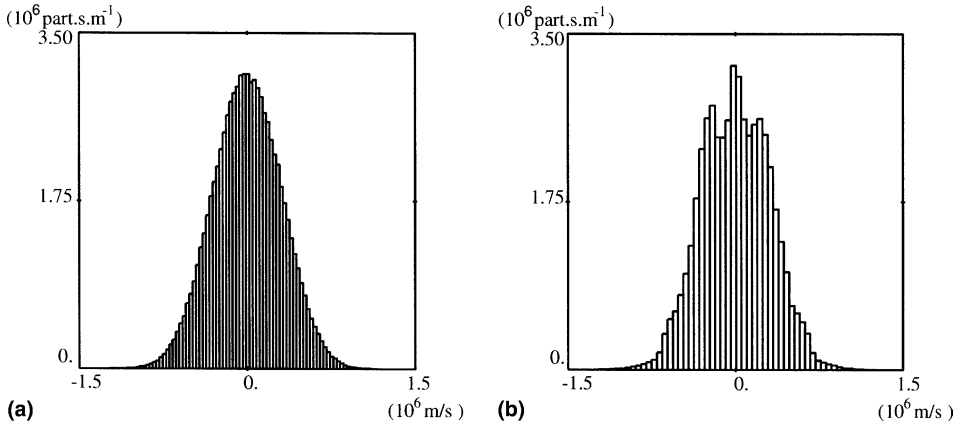

Fig. 12. Velocity distribution function without coalescence (a) and after coalescing with the second energy conservation scheme (b).

9. Conclusion

We have developed a process to construct schemes for coalescing or splitting particles in PIC codes. Particular attention was paid on the conservation of the mesh and particle charge and current densities, and also on the particle energy. Moreover, the number of particles after coalescence can be controlled by choosing an appropriate quadrature formula. As an illustration, we presented numerical results obtained with a 7-particle and a 25-particle schemes.

However, to take fully advantage of this method, there is still work ahead. Let us particularly mention the extension of this coalescence process to quadrilaterals in the two-dimensional case, or to tetrahedra or hexahedra in the three-dimensional one.

Moreover, as it has been shown by numerical results, the use of any coalescence scheme, because of the reduction of the number of particles, can increase the numerical noise on some variables, even if for instance the electromagnetic field distributions are very well preserved.

To overcome this difficulty, it can be worthwhile to define a new strategy of coalescence. As an example, we have proposed in the last section a global approach based on a local procedure, that proved to be efficient. More generally, even if the definition of a well-adapted strategy is case-dependant, it is very useful to be able to a priori control the number of particles as in our method.

Finally it would also be interesting to try other types of interpolation functions, or other quadrature formulae, such as Monte-Carlo methods, in order to continue the investigations about the conservation of the distribution function.

References

[1] P. Degond, F. Hermeline, P.A. Raviart, J. Segré, Numerical modeling of axisymmetric electron beam devices using a coupled particle-finite element method, The 4th Biennial IEEE Conference on Electromagnetic Field Computation, Toronto, Canada, 1990.
[2] A.T. Forrester, Large Ion Beam, Wiley-Interscience, New York, 1988.
[3] R. Illner, S. Rjasanow, Numerical solution of the Boltzmann equation by random discrete velocity models, Eur. J. Mech. B/Fluids 13-2 (1994) 197–210.
[4] G. Lapenta, J.U. Brackbill, Dynamic and selective control of the number of particles in kinetic plasma simulation, J. Comput. Phys. 115 (1994) 213–227.

[5] G. Lapenta, J.U. Brackbill, Control of the number of particles in fluid and MHD particle in cell methods, Comput. Phys. Commun. 87 (1995) 139–154.
[6] T. Pougeard Dulimbert, Extraction de Faisceaux d'Ions à partir de Plasmas Neutres: Modélisation et Simulation Numérique, Thèse de Doctorat de l'Université Paris VI, 2001.
[7] P.A. Raviart, An Analysis of Particle Methods, Springer, Berlin, 1985.
[8] O.C. Zienkiewicz, The Finite Element Method in Engineering Science, Ediscience, McGraw-Hill, 1971.