JOURNAL OF COMPUTATIONAL PHYSICS **127,** 268–284 (1996)
ARTICLE NO. 0174

Blob Method for Kinetic Plasma Simulation with Variable-Size Particles

G. G. M. COPPA, G. LAPENTA,* G. DELLAPIANA, F. DONATO, AND V. RICCARDO†

Dipartimento di Energetica, Politecnico di Torino, Corso Duca Degli Abruzzi, 24, 10129 Torino, Italy

Received May 31, 1995; revised February 8, 1996

A new approach to particle in cell simulation is presented for collisionless plasmas. The new method is based on computational particles of variable size (*blobs*). Each blob represents an element of phase space and its shape and size are evolved in time to represent better the correct evolution. The blob particles can be split to increase the accuracy in selected regions of stronger gradients. Blobs can also be coalesced to keep the total number of blobs constant. The performance of the blob method is analyzed in the case of the Landau damping and in the formation of a sheath in a bounded plasma. The results show the correctness and effectiveness of the new method. When compared to standard PIC methods, the blob technique reduces the noise for a given number of computational particles.

I. INTRODUCTION

Particle in cell (PIC) methods have emerged over the years as the most effective tool for kinetic plasma simulation. However, some relevant issues still remain open. In particular, the present paper focuses on two of them.

First, PIC methods present a level of noise due to the statistical nature of particle methods for the computational model is composed of far fewer particles than the real plasma. Noise reduction techniques have been a major field of research for many years. In some applications, PIC codes are too noisy to gain conclusive physical results. An approach to reduce the noise is the δf method [1], in which only the perturbation from the equilibrium distribution is studied with computational particles. The noise level is only due to the perturbation and is considerably reduced. Unfortunately, many physical systems are hard to describe in terms of a perturbed equilibrium, and the component of the distribution described with particles is so relevant that the δf method reduces to a conventional PIC without significant noise reduction.

A second area of major concern is related to systems where multiple length scales are present. Typical examples are the collisionless shocks observed in the magnetosphere or the sheaths at the boundary of laboratory plasmas [2]. In these cases, the plasma presents local, rapid variations in small portions of the system. A possible approach in the numerical study of multiple length-scale problems is the use of nonuniform or adaptive grids, with fine grid spacing in the regions of strong gradients and larger spacing where variations are mild. Finite-element and finite-difference codes adopt nonuniform or adaptive grid packages in order to obtain the best trade-off between accuracy and cost [3]. PIC methods are difficult to rezone: the rezoning of the grid must be accompanied by a similar rezoning of the particles to gain a real increase in the accuracy. Only recently, the problem of particle rezoning has been tackled on a sound mathematical basis [2].

In the present paper, a new particle method is presented, based on variable-size particles [4]. The method shares with conventional PICs the idea of repressenting the plasma with finite-size particles, each representing a large number of physical particles. However, in the present method, also the shape and the size of the computational particles are evolved in time to account for the different evolution of the physical components of each computational particle. The new approach presents some significant improvements. The noise level is reduced. The additional information carried by each computational particle allows one to gain low levels of noise with only a few particles per cell. Moreover, the additional cost involved in handling the increased amount of information per particle is more than compensated for by the reduction of noise.

The method presented here can also address effectively multiple-length scale problems, since the concepts of particle rezoning presented in Ref. [2] can be used very naturally, leading to a self-rezoning method.

The paper is structured as follows: in Section II the physical model is presented, the equations of motion of the computational particles are derived in 1D and their validity is discussed. In Section III, the interactions between the particles and the electric field are described and the interpolation scheme is introduced. In Section IV, the problem of controlling the size of a particle is studied. In

* Currently Director's Postdoctoral Fellow Theoretical Division, Mail Stop B216, Los Alamos National Laboratory Los Alamos, MN 87545.

† Present address: JET—Joint Undertaking, Magnet and Power Supply Division, Engineering Analysis Group, Abingdon, OX14 3EA, UK.

0021-9991/96 $18.00

Section V, a selection of results is presented. The extension to multidimensional electromagnetic problems is discussed in Appendix A. The numerical solution of the equations of motion is described in detail in Appendix B.

II. PHYSICAL MODEL

In the present paper, the attention is focused on a 1D electrostatic model of an electron plasma with a uniform background of motionless ions. The electron distribution function f is governed by the Vlasov equation,

$$\frac{\partial f}{\partial t} + v\frac{\partial f}{\partial x} - \frac{eE}{m}\frac{\partial f}{\partial v} = 0, \tag{1}$$

in which the electric field can be evaluated from the charge density by means of Poisson's equation,

$$\frac{\partial^2 \Phi}{\partial x^2} = -\frac{\rho}{\varepsilon_o}, \quad E = -\frac{\partial \Phi}{\partial x}. \tag{2}$$

The net charge density is the sum of the electron charges and the uniform ion background ρ_0:

$$\rho(x,t) = -e\int f(x,v,t)\,dv + \rho_0. \tag{3}$$

The electron mass is m, the electron charge is $-e$.

Many of the following derivations hold for more general models. In the Appendix A, the extension to multidimensional electromagnetic problems is sketched.

In PIC methods, the real plasma is represented by computational particles, whose number is much smaller than the real number of electrons in the system. For this reason, each computational particle is given a finite size and represents a large number of physical particles with similar properties. A complete description of PIC methods can be found in the classical textbooks by Birdsall and Langdon [5] and by Hockney and Eastwood [6].

The method described in the present paper is based on the same particle modeling of standard PIC methods, but each particle is assigned additional descriptors. In fact, a computational particle represents a set of physical particles and should have internal degrees of freedom related to the different motion of its components. The additional descriptors can be expected to reduce the number of particles needed to represent a given system. As will be shown, an effective numerical algorithm can be obtained if the evolution of size and shape of the particles is followed. It must be noted that the idea has been presented only recently in PIC methods [4, 7], but it has been finding wide application in other areas of plasma and fluid simulation. In the multiple water-bag method for kinetic plasma simulation, the phase space is discretized in a number of zones of constant density that are evolved in time following the evolution of their boundaries [8]. Similarly, in simulations of incompressible fluids, the space is discretized in patches of constant vorticity. In the blob vortex method [9], the fluid is represented by "particles" having a given vorticity but a size changing in time. Moreover, incompressible flows have been simulated by elliptical patches of constant vorticity [10]. During the simulation, each ellipse changes its position, aspect ratio, and orientation, preserving the elliptical shape.

In the particle approach, the phase space is covered with a distribution of N_p computational particles, and the distribution function is represented as

$$f(x,v,t) = \sum_{p=1}^{N_p} n_p f_p(x,v,t), \tag{4}$$

where f_p is the particle distribution relative to the computational particle p and n_p is the number of physical particles in the computational particle p. In standard PICs, f_p is a given shape function centered around the baricenter of the particle, usually a Dirac delta $\delta(v - v_p)$ in velocity, and a b-spline $b_l(x - x_p)$ in space,

$$f_p(x,v,t) = b_l\left(\frac{x - x_p}{\alpha_p}\right)\delta(v - v_p), \tag{5}$$

where α_p is the width of the particle.

In the present paper, the assumption of fixed shape is relaxed, and a more detailed study of the evolution of the phase-space distribution function, f_p, associated to a computational particle, is performed. In fact, the formulation of the particle method allows one to include internal degrees of freedom.

In the following, computational particles of variable size and shape are called blobs, as the particle shape is not fixed but changes like a sort of blob and are labelled by b.

Each blob distribution function f_b is governed by the same Vlasov equation valid for the whole system:

$$\frac{\partial f_b}{\partial t} + v\frac{\partial f_b}{\partial x} - \frac{eE}{m}\frac{\partial f_b}{\partial v} = 0. \tag{6}$$

In addition, f_b is normalized in such a way that

$$\iint f_b(x,v,t)\,dx\,dv = 1. \tag{7}$$

The electric field in Eq. (6) is still obtained from Poisson's equation (2), where the charge ρ now is given by

$$\rho(x,t) = \sum_b q_b \int f_b(x,v,t)\,dv + \rho_0, \tag{8}$$

q_b being the charge of each blob.

The description given by Eqs. (6) and (8) is exact, if the phase-space distribution at the initial time is given by

$$f(x, v, t = 0) = \sum_{b=1}^{N_b} n_b f_b(x, v, t = 0), \tag{9}$$

where N_b is the number of blobs and n_b is the number of physical particles per blob. However, there is no numerical advantage in solving Eqs. (6) and (8) instead of Eqs. (1)–(3). To obtain a simplified closed system of equation, it is necessary to choose a simplified expression for the blob distribution function f_b. The simplest model that generalizes the PIC assumption, Eq. (5), is to require that the distribution f_b is a function of the most general linear combination of the velocity v and position x,

$$f_b(x, v, t) = J\mathcal{S}(q_1, q_2), \tag{10}$$

where the shape function $\mathcal{S}$ is a given function (to be specified later) of two variables, q_1 and q_2, obtained from x and v with a linear combination with variable coefficients,

$$\begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \hat{\Theta}(t) \cdot \begin{pmatrix} x - x_b(t) \\ v - v_b(t) \end{pmatrix}, \tag{11}$$

where $\hat{\Theta}(t)$ is the following 2×2 matrix,

$$\hat{\Theta}(t) = \begin{pmatrix} \theta_1(t) & \theta_2(t) \\ \theta_3(t) & \theta_4(t) \end{pmatrix}, \tag{12}$$

and $J = |\theta_1\theta_4 - \theta_2\theta_3|$ is the Jacobian of the linear transformation (11).

In Eq. (10), $\mathcal{S}$ must be normalized as

$$\iint \mathcal{S}(q_1, q_2)\, dq_1\, dq_2 = 1 \tag{13}$$

in order to fulfil condition (7).

Equation (11) characterizes the most general linear transformation with six parameters: the position x_b and velocity v_b of the baricenter and four shape parameters, θ_1, θ_2, θ_3, θ_4. The choice of the function $\mathcal{S}$, apart from condition (13), is free, similarly to the choice of the spatial shape function in standard PICs [6]. Figure 1 shows the phase-space support of a blob when $\mathcal{S}(q_1, q_2) = b_0(q_1)b_0(q_2)$. In this case, a blob is represented as a parallelogram of uniform density in phase-space.

With the assumptions (10)–(11) for f_b, a simple closed system of ordinary differential equations can be derived if one supposes that the electric field is a linear function of the spatial coordinate x over the blob support. In fact, by inserting Eqs. (10)–(11) into Eq. (6), the equations of motion for the six parameters characterizing a blob can be readily obtained, as

$$\begin{aligned} &\frac{dx_b}{dt} = v_b, \qquad \frac{dv_b}{dt} = a_0, \\ &\frac{d\theta_1}{dt} = -a_1\theta_2, \quad \frac{d\theta_2}{dt} = -\theta_1, \\ &\frac{d\theta_3}{dt} = -a_1\theta_4, \quad \frac{d\theta_4}{dt} = -\theta_3, \end{aligned} \tag{14}$$

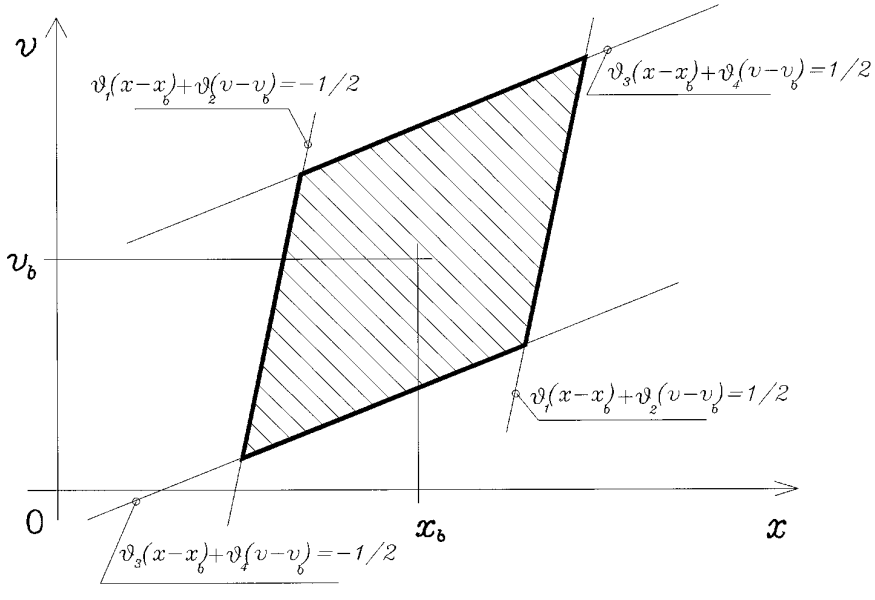

FIG. 1. Blob shape in the phase space, according to Eq. (10). A shape function $\mathcal{S}(q_1, q_2) = b_0(q_1)b_0(q_2)$ is assumed.

where the quantities

$$a_0 = -\frac{e}{m} E(x_b, t), \quad a_1 = -\frac{e}{m}\frac{\partial E}{\partial x} \tag{15}$$

have been introduced. A few comments on Eqs. (14)–(15) are in order. First, J is a constant of motion, as required by the definition (10) (it can be verified using Eqs. (14)). Second, Eqs. (14) are exact, in the assumption that E is linear over the support of the blob. The meaning of the result is that the most general motion for a system of particles obeying the Vlasov equation in a linear electric field is the linear transformation given by Eq. (11). If the electric field is nonlinear, as will be in general the case, Eqs. (14) can be regarded as an approximation valid under the hypothesis that the field $E(x)$ can be approximated satisfactorily as

$$E(x) = \mathscr{E}_{0,b}(x_b, \hat{\Theta}) + \mathscr{E}_{1,b}(x_b, \hat{\Theta})\,(x - x_b) \tag{16}$$

over the support of the considered blob.

The values of $\mathscr{E}_{0,b}$ and $\mathscr{E}_{1,b}$ are obtained by requiring a best fit of the dependence of the electric field,

$$\iint [E(x) - \mathscr{E}_{0,b} - \mathscr{E}_{1,b}(x - x_b)]^2 f_b\, dx\, dv = \min, \tag{17}$$

that yields

$$\mathscr{E}_{0,b} = \int E(x)\, \mathscr{S}_x(x - x_b, \hat{\Theta})\, dx, \tag{18}$$

$$\mathscr{E}_{1,b} = \frac{\int E(x)\,(x - x_b)\, \mathscr{S}_x(x - x_b, \hat{\Theta})\, dx}{\int (x - x_b)^2\, \mathscr{S}_x(x - x_b, \hat{\Theta})\, dx} \tag{19}$$

in which the blob spatial shape function, $\mathscr{S}_x$, has been introduced:

$$\mathscr{S}_x(x - x_b, \hat{\Theta}) = \int \mathscr{S}(\theta_1(x - x_b) + \theta_2 w, \theta_3(x - x_b) + \theta_4 w)\, dw. \tag{20}$$

A. Simplified Formulation with a Reduced Set of Shape Parameters

The Eqs. (14)–(15), together with the definitions (18)–(19) represent a closed system of ordinary differential equations that can be regarded as an alternative to standard PIC methods. This formulation has the advantage of being amenable to extension to electromagnetic multidimensional problems, as described in the Appendix A. In 1D problems, however, a simpler approach can be followed. Instead of using the six parameters, x_b, v_b, θ_1, θ_2, θ_3, θ_4, introduced above, the motion of a blob is characterized by the evolution of x_b and v_b as above and by fewer additional parameters characterizing the shape.

The alternative approach followed here makes use of a moment chain replacing Eq. (6). The chain is truncated at a given order to obtain a closed system of ordinary differential equations for the moments considered. This can be done by assigning a blob shape f_b depending on the moments considered.

To describe the shape of the blobs, the second-order moments of $f_b(x, v, t)$ are used. The evolution of the shape is characterized by the variance in x:

$$\sigma_{xx,b}(t) = \iint f_b(x, v, t)\,(x - x_b)^2\, dx\, dv \tag{21}$$

that describes the size in x, by the variance in v,

$$\sigma_{vv,b}(t) = \iint f_b(x, v, t)\,(v - v_b)^2\, dx\, dv \tag{22}$$

that describes the size in v, and by the $x - v$ cross-correlation,

$$\sigma_{xv,b}(t) = \iint f_b(x, v, t)\,(x - x_b)\,(v - v_b)\, dx\, dv \tag{23}$$

that describes the slope in phase-space.

The mathematical derivation makes use of the following definition of the mean value of a function ψ of x and v over the support $\mathscr{V}_b$ of a blob b:

$$\langle \psi \rangle_b = \iint_{\mathscr{V}_b} \psi(x, v)\, f_b(x, v)\, dx\, dv. \tag{24}$$

In the following derivations, the support $\mathscr{V}_b$ of f_b may extend to the whole phase space, but for computational reasons it is more practical to keep the support "small" in space and in velocity, as usual in PIC methods.

By multiplying Eq. (6) by ψ and integrating over x and v, the equation of motion for the average $\langle \psi \rangle_b$ follows:

$$\frac{d\langle \psi \rangle_b}{dt} = \langle v\, \partial\psi/\partial x \rangle_b - \frac{e}{m} \langle E\, \partial\psi/\partial v \rangle_b. \tag{25}$$

The moment description of the blob distribution can be obtained by specializing the definition of ψ. In the following, the attention will focus on the first five moments of f_b listed in Table I. Their equations of motion are readily

TABLE I

Definition of the First Five Moments of the Distribution Function f_b

Moment	Definition
Center of mass position	$x_b = \langle x \rangle_b$
Center of mass velocity	$v_b = \langle v \rangle_b$
Space variance	$\sigma_{xx,b} = \langle (x - x_b)^2 \rangle_b$
Velocity variance	$\sigma_{vv,b} = \langle (v - v_b)^2 \rangle_b$
Covariance	$\sigma_{vv,b} = \langle (x - x_b)(v - v_b) \rangle_b$

obtained from Eq. (25):

$$\begin{aligned}
\frac{dx_b}{dt} &= v_b \\
\frac{dv_b}{dt} &= -\frac{e}{m}\langle E\rangle_b \\
\frac{d\sigma_{xx,b}}{dt} &= 2\sigma_{xv,b} \\
\frac{d\sigma_{xv,b}}{dt} &= \sigma_{vv,b} - \frac{e}{m}\langle (E-\langle E\rangle_b)\,(x-x_b)\rangle_b \\
\frac{d\sigma_{vv,b}}{dt} &= -\frac{2e}{m}\langle (E-\langle E\rangle_b)\,(v-v_b)\rangle_b .
\end{aligned} \tag{26}$$

Equations (26) are as rigorously correct as Eq. (6). However, as usual in moment procedures, they are not closed. In particular, the terms in the second, fourth, and fifth of Eqs. (26) contain the electric field and cannot be evaluated unless the distribution f_b is fully known; i.e., all the infinite moments of f_b should be known. Thus, a closure assumption is needed. The closure of Eqs. (26) requires us to supplement them with a rule to evaluate the terms containing the electric field, once the first five moments are given. Therefore, the most general closure assumption is to require a functional dependence of f_b on x_b, v_b, $\sigma_{xx,b}$, $\sigma_{vv,b}$, and $\sigma_{xv,b}$ only:

$$f_b(x,v,t) = \mathcal{S}(x,v;\{x_b(t), v_b(t), \sigma_{xx,b}(t), \sigma_{xv,b}(t), \sigma_{xv,b}(t)\}). \tag{27}$$

The closure assumption, Eq. (27), expresses the blob distribution function in terms of the first- and second-order moments only and allows one to obtain a closed system of equations. In principle, any functional dependence on the first- and second-order moments could be assumed. This freedom can be used to choose a functional form which simplifies the final set of equations.

To simplify the charge-gathering phase, the functional dependence has been chosen by factorizing the velocity and space dependences as

$$\begin{aligned}
f_b(x,v,t) = {} & \mathcal{S}_x(x-x_b, \sigma_{xx,b}) \\
& \mathcal{S}_v\left(v - v_b - \frac{\sigma_{xv,b}}{\sigma_{xx,b}}(x-x_b), \sigma_{vv,b} - \frac{\sigma_{xv,b}^2}{\sigma_{xx,b}}\right).
\end{aligned} \tag{28}$$

In such a way, the velocity distribution at each spatial point has the same shape but shifted to lower or higher velocities by the term $\sigma_{xv,b}\,(x - x_b)/\sigma_{xx,b}$. The functions $\mathcal{S}_x$ and $\mathcal{S}_v$ are normalized to unity. The variance $\sigma_{xx,b}$ determines the size in x, while $\sigma_{vv,b}$, $\sigma_{xx,b}$, and $\sigma_{xv,b}$ are needed to characterize the size and slope in v. This particular form of the closure assumption results in simpler equations of motion.

The shape functions $\mathcal{S}_x$ and $\mathcal{S}_v$ can be expressed conveniently in terms of b-spline functions of order l:

$$\mathcal{S}_\xi(\xi,\sigma) = \frac{1}{\alpha} b_l(\xi/\alpha), \quad \alpha = \left(\frac{12\sigma}{l+1}\right)^{1/2} \tag{29}$$

(ξ indicates either x or v). Figure 2 shows the support of a blob in the phase space for the case of b-splines of order 0 in both x and v.

With the assumption (28), the mean values in Eqs. (26) containing E can be evaluated as

$$\langle E\rangle_b = \mathscr{E}_{0,b} \tag{30}$$

$$\langle (E-\langle E\rangle_b)\,(x-x_b)\rangle_b = \sigma_{xx,b}\,\mathscr{E}_{1,b} \tag{31}$$

$$\langle (E-\langle E\rangle_b)\,(v-v_b)\rangle_b = \sigma_{xv,b}\,\mathscr{E}_{1,b}, \tag{32}$$

where $\mathscr{E}_{0,b}$ and $\mathscr{E}_{1,b}$ are still given by Eqs. (18)–(19).

Thus, the equation of motion for a blob are

$$\begin{aligned}
\frac{dx_b}{dt} &= v_b \\
\frac{dv_b}{dt} &= -\frac{e}{m}\mathscr{E}_{0,b} \\
\frac{d\sigma_{xx,b}}{dt} &= 2\sigma_{xv,b} \\
\frac{d\sigma_{xv,b}}{dt} &= \sigma_{vv,b} - \frac{e}{m}\sigma_{xx,b}\mathscr{E}_{1,b} \\
\frac{d\sigma_{vv,b}}{dt} &= -2\frac{e}{m}\sigma_{xv,b}\mathscr{E}_{1,b}.
\end{aligned} \tag{33}$$

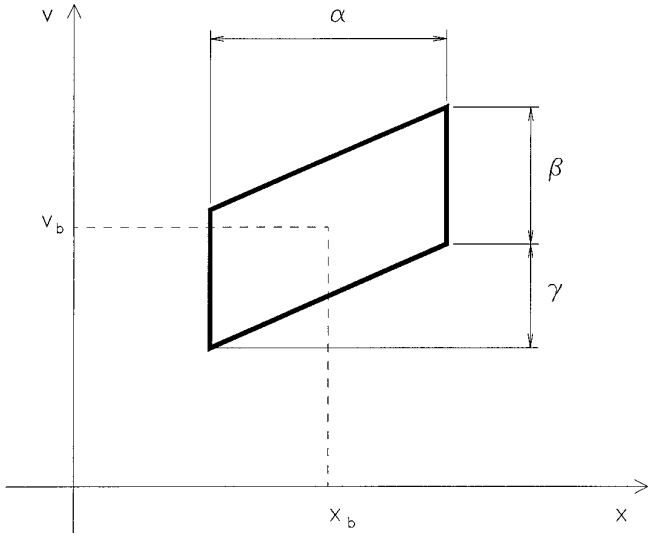

FIG. 2. Blob shape in the phase space, according to Eqs. (27), (29). b-splines of order 0 in both x and v are used.

Equations (33) represent a generalization of the equations of motion for the center of mass of a particle in standard PIC methods. Three additional equations govern shape and size of the particles. However, one of these new equations can be eliminated, observing that the relation

$$\frac{d}{dt}(\sigma_{xx,b}\,\sigma_{vv,b} - \sigma_{xv,b}^2) = 0 \tag{34}$$

is rigorously true, under the hypothesis (28). Equation (34) expresses nothing but the Liouville theorem: the area of phase space occupied by one blob is constant in time. The constant of motion $\mathscr{A}_b = \sigma_{xx,b}\,\sigma_{vv,b} - \sigma_{xv,b}^2$ allows one to reconstruct any one of the second-order moments from the remaining two. Thus, the relevant equations of motion are only four. It is worth noting that the equations of motion for the second-order moments, Eqs. (33), are compatible with the equations for the more general approach, Eqs. (14). Indeed, it is easy to show that Eqs. (14) imply Eqs. (33). From the definitions (21)–(23), the second-order moments of the distribution (10) are

$$\begin{aligned}
\sigma_{xx,b} &= \frac{\theta_2^2\varsigma_{22} + \theta_4^2\varsigma_{11} - 2\theta_2\theta_4\varsigma_{12}}{J^2} \\
\sigma_{vv,b} &= \frac{\theta_1^2\varsigma_{22} + \theta_3^2\varsigma_{11} - 2\theta_1\theta_3\varsigma_{12}}{J^2} \\
\sigma_{vv,b} &= \frac{-\theta_1\theta_2\varsigma_{22} - \theta_3\theta_4\varsigma_{11} + (\theta_1\theta_4 + \theta_2\theta_3)\varsigma_{12}}{J^2},
\end{aligned} \tag{35}$$

where $\varsigma_{ij} = \iint \mathscr{S}(q_1, q_2) q_i q_j dq_1 dq_2$ is the covariance matrix associated with $\mathscr{S}(q_1, q_2)$. Their evolution equations are obtained by taking the time derivative of Eqs. (35) and using Eqs. (14). Equations (33) are obtained again, which proves the compatibility. However, Eqs. (14) provide a more accurate description of the evolution of a blob, using six degrees of freedom. Equations (33) are a simpler, compatible, but not completely equivalent, set of equations. In the following, Eqs. (33) will be used because only four blob properties need to be stored and evolved: $x_b, v_b, \sigma_{xx,b}$, and $\sigma_{xv,b}$. The fifth one ($\sigma_{vv,b}$) is derivable from Eq. (34).

III. PARTICLE–FIELD INTERACTION

In the previous section the evolution equations for a blob in a given field E have been derived. In a plasma, the self-consistent electric field is a function of the particle distribution. Therefore, Eqs. (33) must be coupled with Poisson's equation to give a complete description of the system. The coupling between the blob motion and the electric field is mediated by a computational grid, as in conventional PIC codes. In this section, the blob-in-cell approach is described.

The electric field is studied on a spatial grid. In the 1D model described here, the system is discretized in cells of centers x_g and vertices $x_{g\pm1/2}$. Blob charges are gathered to the grid to obtain the charge q_g in each cell g and the electric field is calculated from the discretized Poisson's equation. The same algorithms used in PIC codes can be applied here [5, 6]. The electric field is then interpolated to the blobs to get $\mathscr{E}_{0,b}$ and $\mathscr{E}_{1,b}$. The novelties of the blob method over conventional PICs are in the gather step, where the blobs are assigned to the cells, and in the interpolation step, where the electric field on the grid centers E_g is used to obtain $\mathscr{E}_{0,b}$ and $\mathscr{E}_{1,b}$.

The assignment of the blob charge q_b to the cells is based on the definition of the charge in a cell g as

$$q_g = \sum_b q_b \int_{x_{g-1/2}}^{x_{g+1/2}} dx \int_{-\infty}^{+\infty} f_b(x, v, t)\, dv. \tag{36}$$

According to this equation, the charge in cell g is calculated as the fraction of the charge of blob b that overlaps cell g. As usual, the charge gathering phase is best represented in terms of an assignment function $\mathscr{W}$, defined as [6]

$$\mathscr{W}(x_g - x_b, \sigma_{xx,b}) = \int_{x_{g-1/2}}^{x_{g+1/2}} \mathscr{S}_x(x - x_b, \sigma_{xx,b})\, dx. \tag{37}$$

In the case of shape functions chosen as b-splines of order 0 (see Eq. (29)), the assignment function $\mathscr{W}$ is drawn in Fig. 3. When the grid is uniform with spacing h, the assignment function $\mathscr{W}$ depends only on the relative position of the blob x_b and the cell center x_g, on the spatial size of the blob $\alpha_b = (12\sigma_{xx,b})^{1/2}$ and on the grid spacing h.

The mathematical expression of this function differs according to the relative dimensions of the blob and the cell. In fact, when $\alpha_b \geq h$, $\mathscr{W}$ can be written as

$$\mathscr{W}(x; \alpha_b, h) = \begin{cases} \dfrac{h}{\alpha_b}, & |x| \leq (\alpha_b - h)/2, \\ \dfrac{h}{\alpha_b}\left[1 - \dfrac{|x| - (\alpha_b - h)/2}{h}\right], & (\alpha_b - h)/2 < |x| \leq (\alpha_b + h)/2, \\ 0, & |x| > (\alpha_b + h)/2, \end{cases} \tag{38}$$

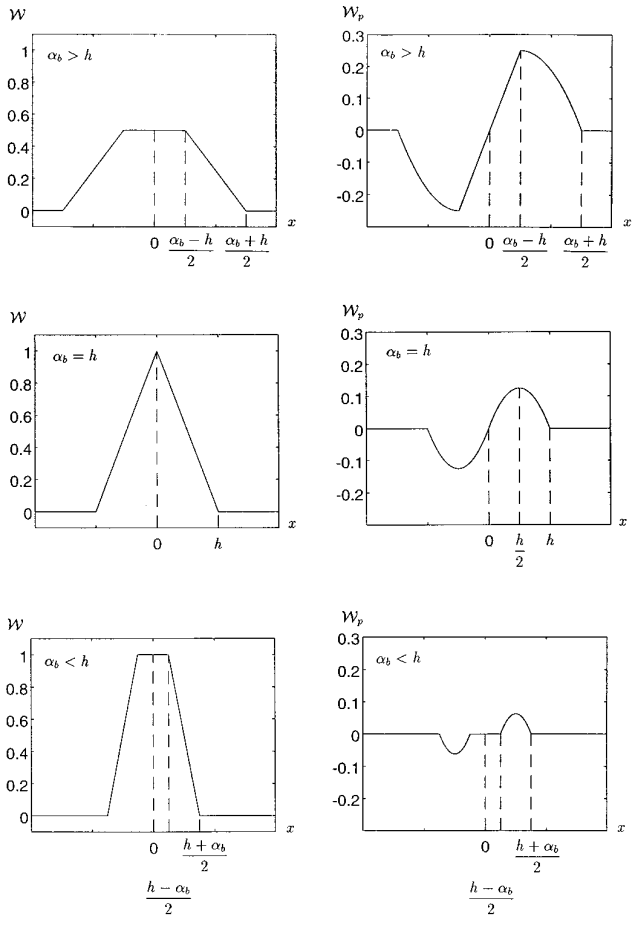

FIG. 3. Assignment functions $\mathscr{W}$ and $\mathscr{W}_p$, for different blob widths α_b relative to the cell size h.

while when $\alpha_b < h$, the expression of $\mathscr{W}$ is

$$\mathscr{W}(x;\alpha_b,h)=\begin{cases}1, & |x|\le (h-\alpha_b)/2,\\ 1-\dfrac{|x|-(h-\alpha_b)/2}{\alpha_b}, & (h-\alpha_b)/2<|x|\le (h+\alpha_b)/2\\ 0, & |x|>(h+\alpha_b)/2.\end{cases} \tag{39}$$

At any rate, the charge gathering phase for the blob method is quite similar to conventional PICs. Variations in size of the computational particles affects the assignment function. A comparison between the case in Fig. 3 and the corresponding CIC assignment function for PIC [6] shows that the use of variable-size particles introduces very little additional intricacy and computational cost.

The introduction of the assignment function $\mathscr{W}$ reduces the charge gathering step to the formula

$$q_g=\sum_b q_b\,\mathscr{W}(x_g-x_b;\alpha_b,h) \tag{40}$$

which can be easily implemented in existing PIC codes.

Once q_g are known, the electric field at the grid centers x_g can be evaluated. This task can be accomplished in different ways [6]. For the present derivation, it suffices to assume that the electric field can be derived from the grid charges q_g as

$$E_g=\sum_{g'} D_{gg'}q_{g'}, \tag{41}$$

where $D_{gg'}$ is the Green's function of the discretized Poisson's equation. The value of the electric field at the grid points, E_g, can be used to evaluate $\mathscr{E}_{0,b}$ and $\mathscr{E}_{1,b}$, defined in Eqs. (18), (19). Consistency considerations suggest [6] that the electric field $E(x)$ must be interpolated from Eqs. (18), (19) by assuming a stepwise dependence of the electric field

$$E(x)=\sum_g b_0\left(\frac{x-x_g}{h}\right)E_g; \tag{42}$$

i.e., the electric field is constant in each cell (equal to the value in the cell center x_g).

With this assumption, Eqs. (18), (19) become

$$\mathscr{E}_{0,b}=\sum_g E_g\int_{x_g-h/2}^{x_g+h/2}\mathscr{S}_x(x-x_b;\sigma_{xx,b})\,dx \tag{43}$$

$$\mathscr{E}_{1,b}=\sum_g E_g\frac{1}{\sigma_{xx,b}}\int_{x_g-h/2}^{x_g+h/2}(x-x_b)\,\mathscr{S}_x(x-x_b;\sigma_{xx,b})\,dx. \tag{44}$$

Equations (43)–(44) can be conveniently expressed in terms of the already-defined assignment function $\mathscr{W}$ and a second assignment function $\mathscr{W}_p$, defined as

$$\mathscr{W}_p=\frac{1}{\sigma_{xx,b}}\int_{x_g-h/2}^{x_g+h/2}(x-x_b)\,\mathscr{S}_x(x-x_b;\sigma_{xx,b})\,dx. \tag{45}$$

Assuming a uniform grid and using b-splines of order 0 for $\mathcal{S}_x$, it follows for $\alpha_b \geq h$

$$\mathcal{W}_p(x;\alpha_b,h) = \frac{12}{\alpha_b^3} \times \begin{cases} hx, & |x| \leq (\alpha_b - h)/2, \\ \dfrac{\mathrm{sgn}(x)}{2}\left[\left(\dfrac{\alpha_b}{2}\right)^2 - \left(|x| - \dfrac{h}{2}\right)^2\right], & (\alpha_b - h)/2 < |x| \leq (\alpha_b + h)/2, \\ 0, & |x| > (\alpha_b + h)/2, \end{cases} \tag{46}$$

and for $\alpha_b < h$

$$\mathcal{W}_p(x;\alpha_b,h) = \frac{12}{\alpha_b^3} \times \begin{cases} 0, & |x| \leq (h - \alpha_b)/2, \\ \dfrac{\mathrm{sgn}(x)}{\alpha_b}\left[\left(\dfrac{\alpha_b}{2}\right)^2 - \left(|x| - \dfrac{h}{2}\right)^2\right], & (h - \alpha_b)/2 < |x| \leq (h + \alpha_b)/2, \\ 0, & |x| > (h + \alpha_b)/2, \end{cases} \tag{47}$$

where $\mathrm{sgn}(x)$ is the sign function. The behavior of the function $\mathcal{W}_p$ is shown in Fig. 3.

The assignment functions $\mathcal{W}$ and $\mathcal{W}_p$ allow one to formulate the electric field interpolation phase as

$$\mathcal{E}_{0,b} = \sum_{g(b)} E_g \, \mathcal{W}(x_g - x_b;\alpha_b,h), \tag{48}$$

$$\mathcal{E}_{1,b} = \sum_{g(b)} E_g \, \mathcal{W}_p(x_g - x_b;\alpha_b,h). \tag{49}$$

The summations in Eqs. (48)–(49) extend only to the grid cells overlapped by the support of a blob. Moreover, the choice of a stepwise electric field, (Eq. (42)), results in the same assignment function for the charge gathering phase, (Eq. (36)), and for the average field interpolation, (Eq. (48)). This circumstance provides a momentum-conserving scheme [5, 6].

Once more, the differences with respect to standard PICs are minimal; the only real change is in the assignment functions. In fact, the additional parameter $\mathcal{E}_{1,b}$ is obtained in a way very similar to that for $\mathcal{E}_{0,b}$. These similarities allow one to upgrade preexisting PIC codes to the blob method with only marginal changes.

The above reported equations complete the picture of the basic elements of the blob method. In Fig. 4 the basic steps of a time cycle of a blob-in-cell code are summarized

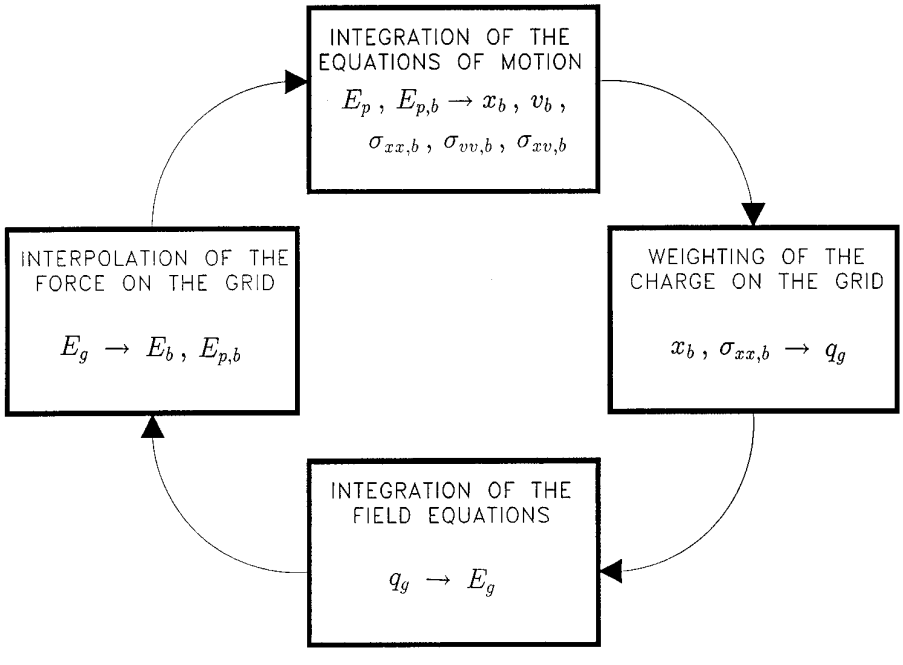

FIG. 4. Basic steps of the time cycle in a blob-in-cell code.

From the configuration of the blobs at a given time, the charge on the grid can be obtained, (Eq. (40)), and the electric fields can be found on the grid. With Eqs. (48)–(49), the coefficients $\mathscr{E}_{0,b}$ and $\mathscr{E}_{1,b}$ can be evaluated. Then, the equations of motion (33) are used to advance particles for a time step Δt, and a new configuration of the blob is obtained from which the computational cycle is followed again.

The numerical solution of the equations of motion for a time step Δt can be performed through standard methods. However, some peculiarities of the equations of motion of the blob method require additional attention. Most notably, the numerical schemes must preserve the positivity of the variances $\sigma_{xx,b}$ and $\sigma_{vv,b}$. In Appendix B, an effective time discretization technique is presented.

IV. CONTROL OF THE SIZE OF A BLOB

In the blob method, the size of the computational particles changes dynamically. Therefore, it is important to address the issue of the maximum allowed size. For computational reasons, the size of a blob should not grow too large. In fact, the cost of the interpolation and gathering phases is proportional to the number of grid points affected by each computational particle. However, a more stringent physical requirement sets the maximum size of a computational particle. In fact, in the derivation of the equations of motion, a closure assumption has been applied, limiting the validity of the results obtained. As discussed in Section II, the closure assumption, Eq. (27), is equivalent to assuming the linearity of the electric field over the support of a computational particle. When the nonlinearity is strong, higher-order moments become necessary for an accurate description of the time evolution and the closure assumption fails.

Two causes lead to the failure of the hypothesis of a linear field during the evolution of a blob. First, in multiple length scale problems, blobs may move from regions of small field gradients, where the linearity requirement is satisfied by fairly large particles, to regions of strong gradients, where the size must be much smaller. Second, the natural evolution of variable-size particles is towards larger sizes. As intuition suggests, the portion of a blob with higher velocity will move faster, leading to a spreading of the computational particle.

To prove this effect rigorously, the equations governing the evolution of a single blob in a field-free region can be solved. From Eq. (33), one obtains

$$\sigma_{xx,b}(t) = \sigma_{xx,b}(0) + 2\sigma_{xv,b}(0)\, t + \sigma_{vv,b}(0)\, t^2. \qquad (50)$$

Equation (50) asserts the natural tendency of a finite size particle to widen. A few considerations on Eq. (50) are in order. First, in the presence of electric fields, $\mathscr{E}_{1,b}$ can accelerate or decelerate the increase of the width σ_{xx}, depending on its sign. Second, even though one considered a zero width in velocity $\sigma_{vv}(0) = 0$, to avoid any widening (as done in standard PIC methods) the electric field would distort the computational particle from a straight horizontal line in the phase space and the widening effect would be reintroduced.

The maximum size of each blob is determined by the requirement of linearity of the electric field over the support of a blob. This suggests the criterion

$$\frac{\int_{\gamma_b} [E(x) - \mathscr{E}_{0,b} - (x - x_b)\, \mathscr{E}_{1,b}]^2\, \mathscr{S}_x(x - x_b, \sigma_{xx,b})\, dx}{\int_{\gamma_b} E^2(x)\, \mathscr{S}_x(x - x_b, \sigma_{xx,b})\, dx} \le \varepsilon, \qquad (51)$$

where ε is a suitable constant. Even if initially the computational particles satisfy the requirement in Eq. (51), during the evolution their size may grow too large.

The effect of multiple length-scales and the natural tendency of the computational particles to widen require a method to control the size. To address this issue, useful hints can be found in recent studies on the control of the number of particles in PIC methods [2]. Following the approach presented in Ref. [2], the size is controlled by splitting a computational particle when the constraint in Eq. (51) is no longer satisfied. The parallelogram in phase-space is split in two equal parts, giving rise to two new blobs whose properties are given by

$$x_{1,2} = x_0 \pm \left(\frac{3}{4}\sigma_{xx,0}\right)^{1/2},$$

$$v_{1,2} = v_0 \pm \left(\frac{3}{4}\frac{\sigma_{xv,0}^2}{\sigma_{xx,0}}\right)^{1/2}, \qquad (52)$$

$$\sigma_{vv_{1,2}} = \sigma_{vv,0} - \frac{3}{4}\frac{\sigma_{xv,0}^2}{\sigma_{xx,0}},$$

$$\sigma_{xx_{1,2}} = \frac{1}{4}\sigma_{xx,0}, \quad \sigma_{xv_{1,2}} = \frac{1}{4}\sigma_{xv,0}, \qquad (53)$$

where the parameters are labeled by 0 for the parent particle and by 1 and 2 for the daughters. Such a scheme satisfies the two basic requirements discussed in Ref. [2]. In fact, the charge assignment to the cells is not changed by the splitting process and neither does the velocity distribution. The fulfillment of these two requirements guarantees the exact preservation of the correct evolution of the system [2]. Consequently, the blob method can be rezoned very accurately and in a natural way. The method is very suitable to multiple length-scale problems where the rezoning is essential.

An unwelcome side effect of the splitting procedure is the increase in the number of blobs during simulations. Two approaches can be followed to overcome this effect.

Apparently, the simplest way is to invert the splitting process by coalescing two small blobs into a bigger one that still satisfies condition (51). Unfortunately, the coalescences are difficult to perform in practice. In general, any couple of blobs will never be exactly in the configuration required by Eqs. (52)–(53). Therefore, the coalescence method has to be applied to blobs that are approximately in that configuration. However, it is difficult to assess the error introduced. The same problem arises when splitting and coalescence are used in conventional PICs [2]. In the case of the blob method, the problem is complicated further by the different sizes and shapes of the blobs.

A second approach can be followed to avoid this difficulty. If the widening of the blob were avoided altogether, there would be no need for splitting or coalescing the blobs. If during the computation a blob grows so large that requirement (51) fails, the blob can be reshaped to satisfy the condition. The difficulty is that the physics must not be altered. A possible solution is to add new particles when a blob is reshaped, in order to alter as little as possible the phase-space distribution. The idea can be made formal as follows: the term of the Vlasov equation responsible for the widening is the stream in space, $v\partial f_b/\partial x$. The widening would be removed if v were replaced by the velocity of the baricenter of the blob v_b. The intermediate situation in which v is replaced by $v_b + \theta_b(v - v_b)$, $0 \le \theta_b \le 1$, limits the widening without removing it, but introduces an error. The error can be eliminated by adding a new equation for the difference δf_b between the exact f_b and the one associated with the reshaped blob, f_b^*. By considering the set of equations,

$$\frac{\partial f_b^*}{\partial t} + [v_b + \vartheta_b(v - v_b)]\frac{\partial f_b^*}{\partial x} - \frac{eE}{m}\frac{\partial f_b^*}{\partial v} = 0 \tag{54}$$

$$\frac{\partial(\delta f_b)}{\partial t} + v\frac{\partial(\delta f_b)}{\partial x} - \frac{eE}{m}\frac{\partial(\delta_b)}{\partial v} = -(1 - \vartheta_b)(v - v_b)\frac{\partial f_b^*}{\partial x}, \tag{55}$$

one can see that $f_b^* + \delta f_b$ satisfies exactly the Vlasov equation (6). The distributions f_b^* and δf_b can be described with different levels of accuracy, as δf_b generally represents a small correction to f_b^*: Eq. (54) is still solved with the blob method described in Section II, but Eq. (55) can be solved more coarsely with the PIC method. In fact, by adding Eqs. (55) over the blob index b, one obtains

$$\frac{\partial(\delta f)}{\partial t} + v\frac{\partial(\delta f)}{\partial x} - \frac{eE}{m}\frac{\partial(\delta f)}{\partial v} = \mathscr{S}, \tag{56}$$

where $\delta f = \sum_b \delta f_b$ and the source term

$$\mathscr{S}(x, v, t) = -\sum_b (1 - \vartheta_b)(v - v_b)\frac{\partial f_b^*}{\partial x} \tag{57}$$

have been introduced. Equation (56) is similar to the characteristic equation of the δf method [1] and can be solved with a similar technique. It must be pointed out that here δf is due to the reshaping process (in fact, the source term is proportional to the "reshaping rate," $(1 - \vartheta_b)$) and not to a perturbation of the equilibrium, as in the δf method.

If $\vartheta_b = 1$, the blob method, as described by Eqs. (33), is obtained again; if $\vartheta_b < 1$ is used, the equations of motion for the second-order moments are slightly altered as

$$\begin{aligned}
\frac{d\sigma_{xx,b}}{dt} &= 2\vartheta_b\sigma_{xv,b} \\
\frac{d\sigma_{xv,b}}{dt} &= \vartheta_b\sigma_{vv,b} - \frac{e}{m}\mathscr{E}_{1,b}\sigma_{xx,b} \\
\frac{d\sigma_{vv,b}}{dt} &= -2\frac{e}{m}\mathscr{E}_{1,b}\sigma_{xv,b},
\end{aligned} \tag{58}$$

while the equations for the baricenter still hold. As anticipated, the effect of ϑ_b is to reduce the widening and to cut it completely when $\vartheta_b = 0$.

Equation (55) has a source term depending on f_b^*:

$$\mathscr{S}_g = -(1 - \vartheta_b)(v - v_b)\frac{\partial f_b^*}{\partial x}. \tag{59}$$

The source $\mathscr{S}_g$ has four distinct regions of different signs, depending on the sign of $(v - v_b)$ and of $\partial f_b^*/\partial x$. Each region can be associated with the creation of a new particle, called a seed. Thus, in the reshaping procedure acting in the computational time interval $(t_n, t_n + \Delta t)$, four particles are created, two positive and two negative, as holes in the phase space. The total mass of the four particles, as well as their charge, is zero. So, the new particles introduced in every computational time interval are like small seeds in phase space that produce a correction to the main term due to the blobs. During the evolution of a blob, $\vartheta_b = 1$ can be used until condition (51) is about to fail (the blob is "ripe"); then, ϑ_b can be lowered to reduce the widening and the seeds are generated.

This method of controlling the size of the blobs keeps their number constant and generates seed PIC particles instead. The seed particles are moved and included in the field solution as in the usual PIC method. The approach has two advantages. First, classic PIC particles are easier to control [2] and their evolution is simpler. Second, seed particles are positive and negative and, when pairs are close enough, they can be annihilated, as shown in Fig. 5.

However, the seed method complicates the implementation of the blob method. For this reason, in the following, the application of the blob method without seeds is considered, where only splitting is used to satisfy condition (51). The number of blobs is not controlled.

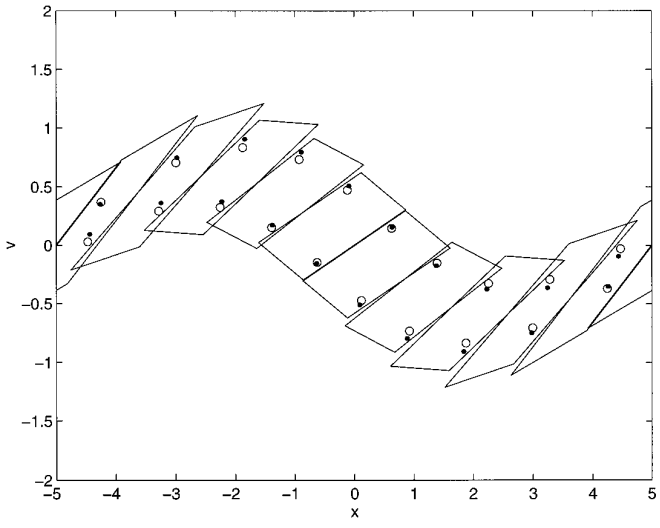

FIG. 5. Creation of positive seeds (dots) and negative seeds (circles) in a warm beam simulated by blobs (full line).

V. RESULTS

The blob method has been implemented in a new code for electrostatic plasma simulations. As mentioned above, many components of the computer code are similar if not identical to existing PIC codes. Nevertheless, the blob method is based on a new approach and needs to be tested both in validity and in effectiveness.

The new blob code has been tested in several problems whose solution was already known. To start with, the blob approximation is analyzed in an idealized case of particles moving in an assigned field. Then, self-consistent solutions are calculated for two-stream instability, Landau damping, and the formation of a sheath. The results are compared with reference results obtained by means of the well-known PIC codes ES1 and PDP1 [5]. In all cases, the blob method shows its capability to represent correctly and effectively the physical evolution.

A. Motion in Assigned Fields

In a first series of calculations, the validity of the approximations behind the blob method is studied. As discussed above, the blob method introduces new parameters to describe a computational particle. A single blob has more degrees of freedom than a conventional particle and it provides a more accurate description of a region of phase space.

To show the improvement induced by these extra parameters, the evolution of a group of physical particles under the action of a harmonic field is considered first. More specifically, particles having the following initial phase space distribution,

$$f(x, v, t = 0) = b_0\left(\frac{x}{L}\right)\delta(v), \tag{60}$$

are considered. These particles can be modelled either with a single blob or by using N fixed-shaped superparticles. If a harmonic field $E(x) = -E_0 \cdot x$ is taken, both the blob and the PIC representation (no matter what the value of N is) provide the exact evolution of the center of mass (i.e., $\bar{x} = 0$). Things are different when the spatial RMS $\Delta x(t) = \sqrt{\sigma_{xx}(t)}$ is considered. In fact, one blob gives the correct answer

$$\Delta x(t) = \frac{L}{\sqrt{12}}|\cos(\omega t)| \tag{61}$$

(ω being the frequency of the oscillator), while according to the PIC modelling, one has

$$\delta x(t) = \frac{L}{\sqrt{12}}\frac{1}{N}[1 + (N^2 - 1(\cos^2(\omega t)]^{1/2}. \tag{62}$$

As shown in Fig. 6, some classic superparticles are needed where a single blob is sufficient.

As a second example, the nonlinear Duffing oscillator is considered. Using dimensionless units, the Hamiltonian for such an oscillator can be written as

$$H(x, p) = \frac{p^2}{2} + \frac{x^4}{4} - \frac{x^2}{2}, \tag{63}$$

where p is the particle momentum. In this case, shown in Figs. 7–9, representing a phase-space region with a single blob gives only an approximate picture of the evolution of the system, and yet, the values of $\bar{x}$ and σ_{xx} obtained are very well approximated for a long time. At length, due to the energy dependence of the oscillator frequency, any initial phase-space region spreads in space so much that a

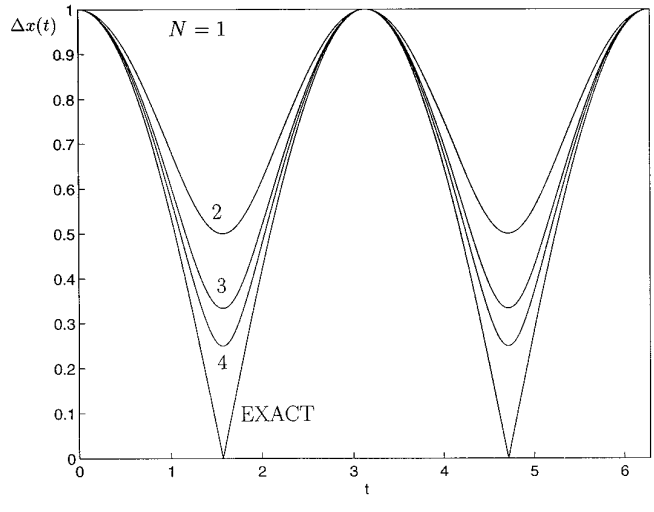

FIG. 6. Behavior of the spatial RMS for a harmonic oscillator, representing the physical system by means of 1, 2, or 4 classic particles and by one blob.

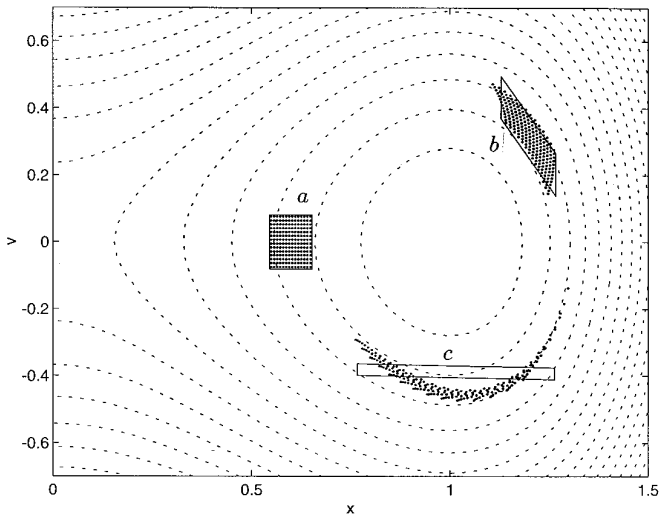

FIG. 7. Phase-space representation of a group of particles for the Duffing oscillator at three different times (a. $t = 0$; b. $t = 2$; c. $t = 23$) and their approximation by means of one blob.

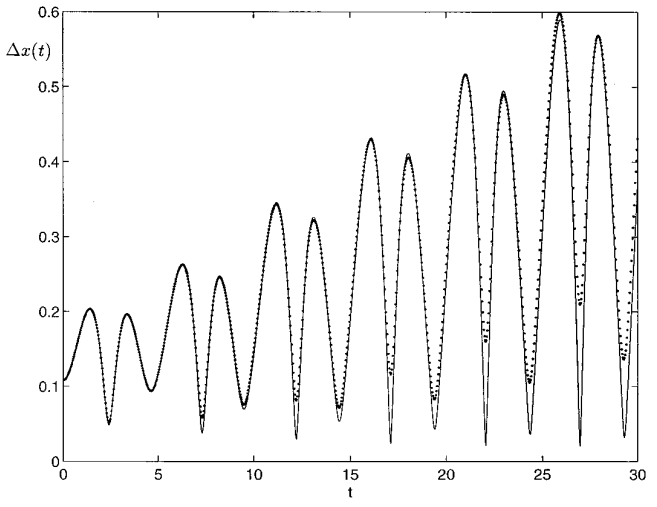

FIG. 9. Behavior of Δx during the evolution of the particle system represented in Fig. 7. The exact solution (dotted line) is compared with the one obtained using one blob (full line).

single blob is no longer sufficient; in this case, the blob must be split in two, and the system is correctly described for another time interval.

B. Two-Stream Instability

The test above proves the accuracy of the blob approach in rather idealized cases. To investigate the viability of the blob method, a simulation code has been applied to some basic Vlasov plasma problems. Units, where the dielectric constant ε_0 is 1 and the charge to mass ratio for electrons is -1, are used.

The two-stream instability is studied first. Two electron beams are counterstreaming with a speed $v_0 = \pm 1$ in a cold background of motionless ions. The electron beams have a thermal velocity $v_t = 0.005$. The electron plasma frequency ω_{pe} is 1. The system has a width $L = 2\pi$, and periodic boundary conditions are used. For the blob simulation, 64 computational particles per beam were loaded initially. The simulation with the blob code is compared with the reference provided by ES1 (4,096 particles for each beam). Figure 10 shows the time evolution of the field energy for the reference case (dotted line) and the blob case (solid line). The qualitative and quantitative agreement is evident.

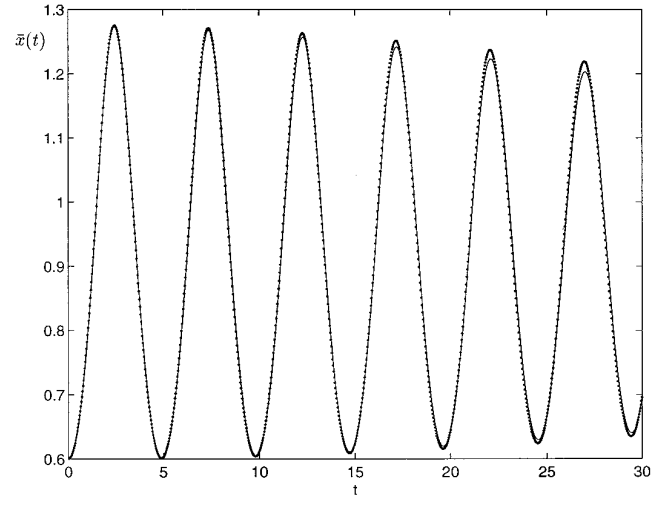

FIG. 8. Behavior of $\bar{x}$ during the evolution of the particle system represented in Fig. 7. The exact solution (dotted line) is compared with the one obtained using one blob (full line).

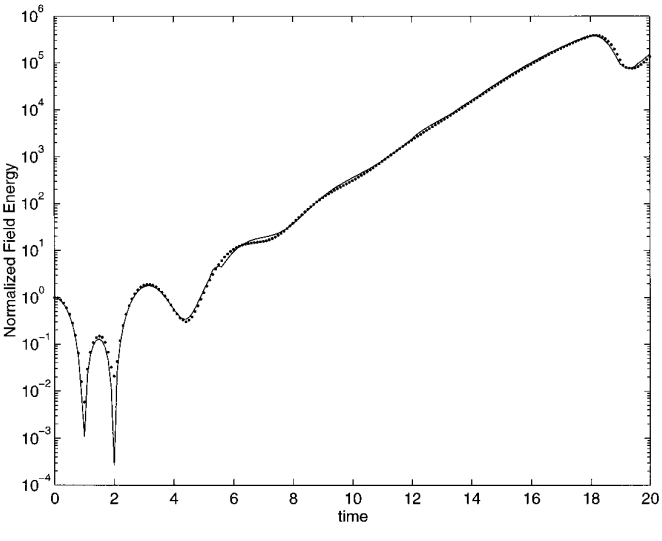

FIG. 10. Two-stream instability: time evolution of the field energy. Solution with the blob method (full line), and reference solution (dotted line). The field energy is normalized to its initial value.

C. *Landau Damping*

A significant test to measure the level of noise of a computational method can be obtained by studying the Landau damping of an initial perturbation in a Maxwellian plasma. The decay process can be followed only for a limited time; eventually the amplitude of the perturbation becomes comparable with the noise level, and the simulation is no longer accurate. A Maxwellian electron plasma (thermal velocity $v_t = 0.4$, $\omega_{pe} = 1$) with a cold ion background is studied. The system has a width $L = 2\pi$, and periodic boundary conditions are used.

The PIC simulation uses ES1 with a quiet start. The blob method uses blobs having initially equal spatial width, with the velocity and the width in velocity chosen to approximate a Maxwellian distribution [11]. The perturbation is imposed on the particle positions, as in ES1,

$$x_b = x_{b,0} + X_1 \sin\left(2\pi \frac{x_{b,0}}{L}\right),$$

where $x_{b,0}$ is the unperturbed position of the blob b and X_1 is the amplitude of the perturbation.

A perturbation $X_1 = 0.1$ is considered first. Figure 11 shows the evolution of field energy obtained with the blob method using 512 particles. The results are compared with a reference calculation obtained with ES1 using a large number of particles (dotted line). Results of calculations performed with ES1 using 1024 particles are shown in Fig. 12. The best performance (solid line) is obtained by introducing a suitable smoothing. Without smoothing (case a) the results are noisy, and with too much smoothing (case b) the frequency and damping rate are inaccurate. On the other hand, the blob method provides automatically a sort of smoothing that reduces the noise. Note that the smoothing in PIC codes can be regarded as a reshaping that increases the size of the particles. A similar effect is present in blob method. However, in the blob case, the additional degrees of freedom of the computational particles reduce the error in the dispersion relation due to the introduction of finite-size particles [11].

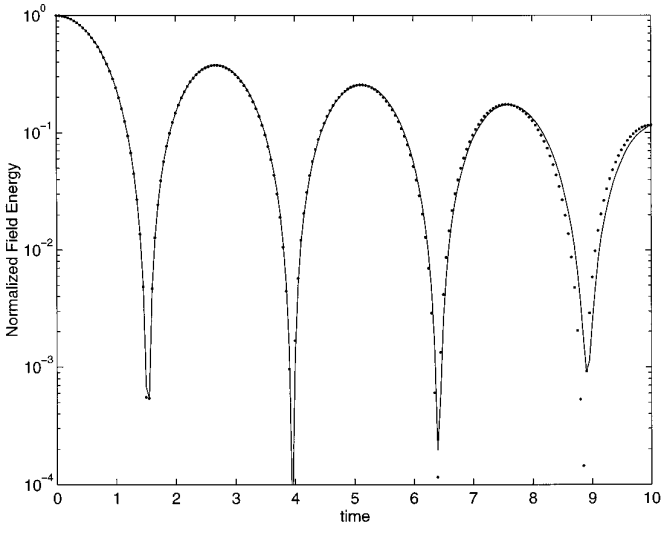

FIG. 11. Landau damping: time evolution of the field energy (normalized to its initial value) with initial perturbation $X_1 = 0.1$. Solution with the blob method with 512 particles (full line) and reference solution (dotted line).

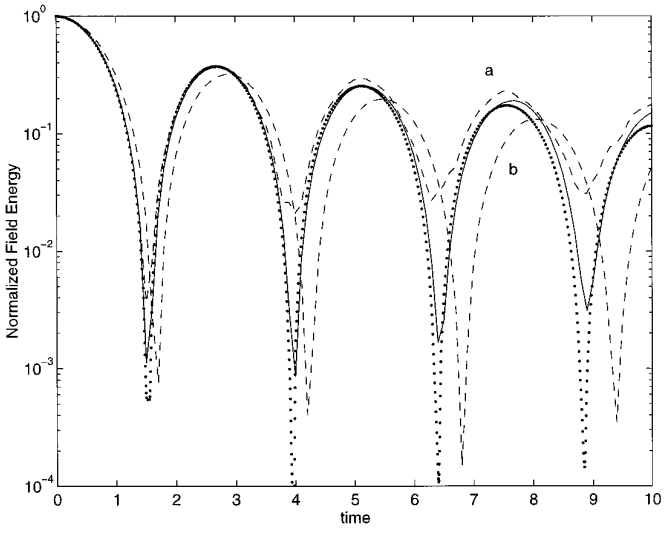

FIG. 12. Landau damping: time evolution of the field energy (normalized to its initial value) with initial perturbation $X_1 = 0.1$. Solution with PIC method using ES1 with 1024 particles and with different smoothing (as determined by the input parameter a_2 of ES1): optimal smoothing ($a_2 = 100$, full line), no smoothing (line a) and excessive smoothing ($a_2 = 1000$, line b). The reference solution (dotted line) is reported also.

The low-noise properties of the blob method are confirmed when a very small initial perturbation is considered, $X_1 = 0.01$. Figure 13 shows the results obtained with the PIC method using 16,000 particles. Again, using smoothing as prescribed above (solid line) the results are very accurate when compared with a reference solution (dotted line). The effect of excessive smoothing or the absence of smoothing is similar as before. When fewer particles are used (4096) the noise of the calculation increases (Fig. 14). Figure 15 shows a blob simulation with 1024 blobs (half the cost of the case in Fig. 14 and one-eighth the cost of the case in Fig. 13). The blob method has low levels of noise and represents correctly the phase and frequency of the wave. In the present cases, blob splitting is not required.

D. *Electrostatic Sheath*

As remarked above, the splitting technique allows an accurate description of multiple length scale problems with the blob method. In such cases, the size of the blobs is automatically adapted to the local length scale; i.e., the width of a blob is fitted to the local variations of the field.

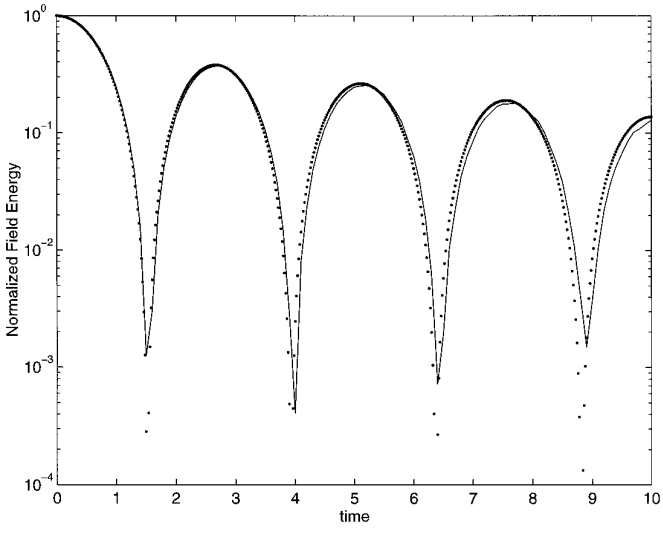

FIG. 13. Landau damping: time evolution of the field energy (normalized to its initial value) with initial perturbation $X_1 = 0.01$. Solution with the PIC method using ES1 with 16,000 particles and optimal smoothing (full line); reference solution (dotted line).

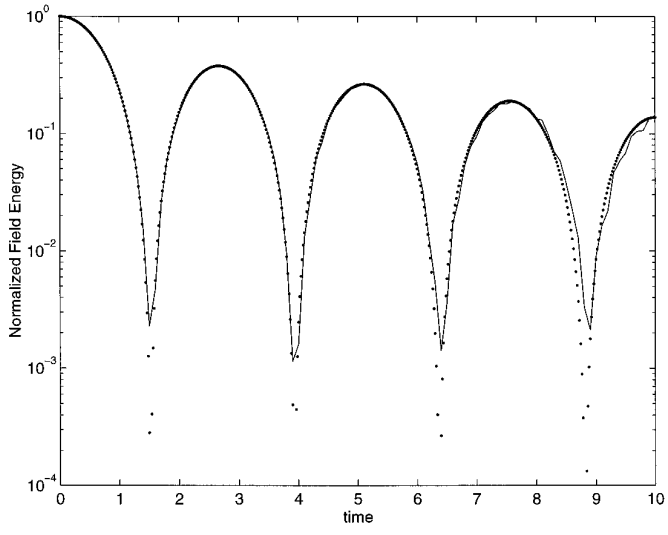

FIG. 15. Landau damping: time evolution of the field energy (normalized to its initial value) with initial perturbation $X_1 = 0.01$. Solution with the blob method with 1024 particles and reference solution (dotted line).

A typical example is the study of sheath formation. An electron plasma with a uniform and Maxwellian distribution ($\omega_{pe} = 1$, $v_t = 0.1$, $\varepsilon_0 = 1$, charge to mass ratio equal to -1) is initially loaded in a system of size $L = 1$. Open boundary conditions are used: in usual PIC codes, a particle is removed when it hits the wall; in the blob approach, only the part of a blob beyond the boundary is removed, while the remaining part of the blob is left in the system.

Figure 16 shows a comparison of the electrostatic potential at $\omega_{pe}t = 6$, from a blob simulation (solid line) and a reference PIC simulation (circles) obtained with PDP1 (using 4000 particles). The potential drop at the wall is very similar. The blob simulation starts with 80 blobs and the number rises to 113 during the simulation in order to satisfy condition (51). The phase-space plot from the blob simulation at $\omega_{pe}t = 6$ (Fig. 17) shows the adaptation of the blob width to the local length scale. Small blobs are used near the walls while large blobs can be used in the interior. This effect is further shown in Fig. 18, where the

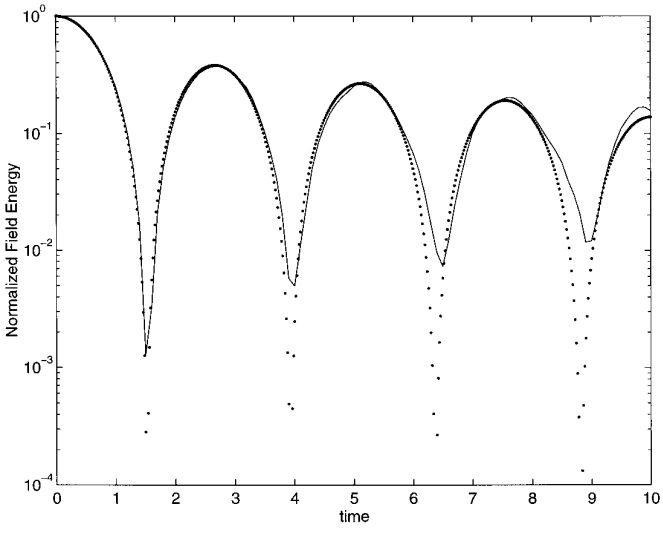

FIG. 14. Landau damping: time evolution of the field energy (normalized to its initial value) with initial perturbation $X_1 = 0.01$. Solution with the PIC method using ES1 with 4096 particles and optimal smoothing (full line); reference solution (dotted line).

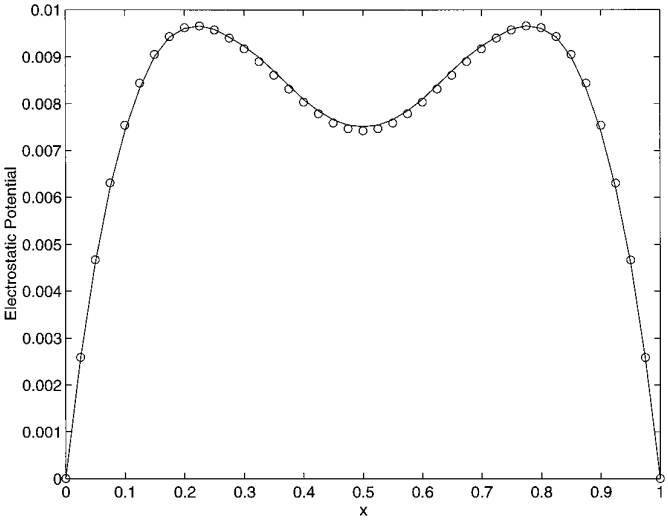

FIG. 16. Electrostatic sheath: space behavior of the electrostatic potential obtained with the PIC method using PDP1 (circles) and with the blob method (solid line) at $\omega_{pe}t = 6$.

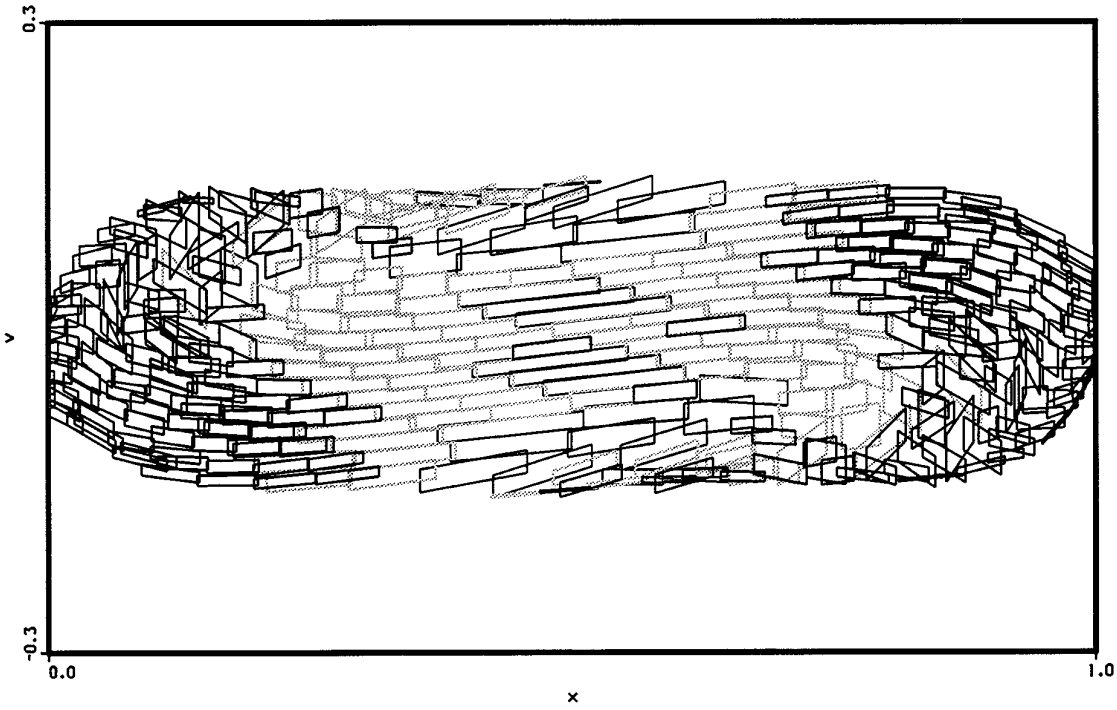

FIG. 17. Electrostatic sheath: phase-space plot obtained with the blob method at $\omega_{pe}t = 6$.

size $\sigma_{xx,b}$ of the blobs is plotted versus the blob position x_b. The average size of the blobs decreases from the bulk of the plasma towards the walls. Figure 17 shows also that blobs that are not lost bounce off the wall and are reflected towards the bulk.

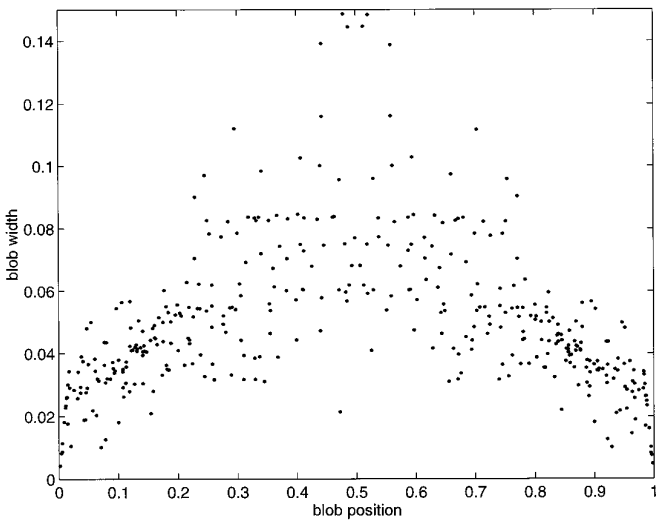

FIG. 18. Electrostatic sheath: blob width α_b versus blob position x_b at $\omega_{pe}t = 6$.

VI. CONCLUSIONS

A new approach to PIC plasma simulation has been presented. It is based on computational particles having a variable shape and size. A 1D electrostatic computer code has been developed and tested in several problems. The blob method gives a low noise and accurate description of the plasma dynamics. The results show that the method can reduce the level of noise at a given cost. The blob method can be easily rezoned and is particularly suited for multiple length-scale problems.

To exploit more fully the possibility of the method, some generalizations are under way. To start with, in the present paper the most simple shape function has been assumed. As with traditional PIC methods, it can be expected that an optimal choice for the shape can give a better accuracy. Second, the technique for blob splitting performs satisfactorily, but further analysis is needed to control their number. Some directions have been outlined above. The method can be extended easily to handle 1D electromagnetic Vlasov plasmas. The inclusion of Fokker–Planck collision terms is straightforward; blobs widen in response to the diffusion and friction terms [12].

Finally, the extension to multidimensional problems can also be undertaken (see Appendix A), even though the straightforward application of the methods presented here would lead to a large number of variables for each blob.

Other approaches can be followed, by introducing additional simplifications on the shape of the blobs in the six-dimensional phase space.

APPENDIX A: FORMULATION OF THE BLOB METHOD IN MULTIDIMENSIONAL SYSTEMS

The formal extension to 2D and 3D problems of the methods described in Section II is straightforward. In the general case, the assumption (10) is generalized as

$$f_b(\mathbf{x}, \mathbf{v}, t) = \hat{\Theta}(t) \cdot \begin{pmatrix} \mathbf{x} - \mathbf{x}_b(t) \\ \mathbf{v} - \mathbf{v}_b(t) \end{pmatrix}, \tag{A1}$$

where

$$\hat{\Theta}(t) = \begin{pmatrix} \hat{\theta}_1 & \hat{\theta}_2 \\ \hat{\theta}_3 & \hat{\theta}_4 \end{pmatrix} \tag{A2}$$

is a 2×2 matrix composed by elements that are matrices themselves (2×2 in 2D and 3×3 in 3D).

The derivation of the equations of motion for the parameters that determine the distribution f_b follow the same steps described above. The acceleration is assumed to be linear over the support of f_b,

$$\mathbf{a} = \mathbf{a}_{0,b} + \hat{a}_{1,b} \cdot (\mathbf{x} - \mathbf{x}_b) + \hat{a}_{2,b} \cdot (\mathbf{v} - \mathbf{v}_b), \tag{A3}$$

where $\hat{a}_{1,b}$ and $\hat{a}_{2,b}$ are matrices (2×2 in 2D and 3×3 in 3D).

Equation (A3) implies a linear electric field and a constant magnetic field over the support of f_b. The following equations of motion for the blobs are obtained by substituting Eq. (A1) into the Vlasov equation:

$$\begin{aligned} \frac{d\mathbf{x}_b}{dt} &= \mathbf{v}_b, & \frac{d\mathbf{v}_b}{dt} &= \mathbf{a}_0, \\ \frac{d\hat{\theta}_1}{dt} &= -\hat{\theta}_2 \hat{a}_1, & \frac{d\hat{\theta}_2}{dt} &= -\hat{\theta}_1 - \hat{\theta}_2 \hat{a}_2, \\ \frac{d\hat{\theta}_3}{dt} &= -\hat{\theta}_4 \hat{a}_1, & \frac{d\hat{\theta}_4}{dt} &= -\hat{\theta}_3 - \hat{\theta} \hat{a}_2, \end{aligned} \tag{A4}$$

which generalize system (14).

The coefficients of the linear expansion (A3) can be calculated by extending the method used in Section II. Now the quantities $\mathbf{a}_{0,b}$, $\hat{a}_{1,b}$, and $\hat{a}_{2,b}$ are obtained by minimizing the expression

$$\int\int \left[\frac{e}{m} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) - \mathbf{a}_{0,b} - \hat{a}_{1,b} \cdot (\mathbf{x} - \mathbf{x}_b) - \hat{a}_{2,b} \cdot (\mathbf{v} - \mathbf{v}_b) \right]^2 f_b \, d\mathbf{x} \, d\mathbf{v} = \min. \tag{A.5}$$

The number of parameters and equations is 20 in 2D and 42 in 3D for each blob. These figures are to be compared with 4 and 8 for the PIC method in 2D and 3D, respectively. As discussed above, the accuracy of the blob method may still offset this seemingly overwhelming increase in complexity. Moreover, one can assume simpler expressions for the shape to replace Eq. (A1). For example, preferred directions can be assumed, reducing the number of parameters per blob.

APPENDIX B: NUMERICAL SOLUTION OF THE EQUATIONS OF MOTION

The equations of motion of a blob, Eqs. (33), are to be solved numerically. This task can be accomplished by using one of the many well-known techniques for the solution of ordinary differential equations. However, σ_{xx} and σ_{vv} are positive definite quantities and the simplest numerical schemes, most notably the leap-frog algorithm, may fail to preserve this properly, leading to severe numerical instabilities. These considerations suggest rewriting the equations of motion in a more convenient form.

An effective scheme can be derived if the second-order moments σ_{xx}, σ_{vv}, and σ_{xv} are expressed in terms of the spatial size α, the width in the velocity space β, and the velocity shear γ (see Fig. 2):

$$\sigma_{xx,b} = \alpha_b^2/12 \tag{B1}$$

$$\sigma_{vv,b} = (\gamma_b^2 + \beta_b^2)/12 \tag{B2}$$

$$\sigma_{xv,b} = \alpha_b \gamma_b/12. \tag{B3}$$

Using Eqs. (B1)–(B3), equations of motion for α, β, and γ can be derived:

$$\begin{aligned} \frac{d\alpha_b}{dt} &= \gamma_b \\ \frac{d\beta_b}{dt} &= -\frac{\beta_b \gamma_b}{\alpha_b} \\ \frac{d\gamma_b}{dt} &= \frac{\beta_b^2}{\alpha_b} - \frac{e}{m} E_{p,b} \alpha_b. \end{aligned} \tag{B4}$$

As noted above, the Liouville theorem requires the area of each blob $\mathscr{A}_b = \alpha_b \beta_b$ to be a constant of motion; one

of the equations (for instance, the one for β_b) becomes redundant and can be eliminated.

In summary, the equations of motion of a blob are

$$\begin{aligned} \frac{dx_b}{dt} &= v_b \\ \frac{dv_b}{dt} &= -\frac{e}{m} E_b \\ \frac{d\alpha_b}{dt} &= \gamma_b \\ \frac{d\gamma_b}{dt} &= \frac{\mathscr{A}_b^2}{\alpha_b^3} - \frac{e}{m} E_{p,b} \alpha_b . \end{aligned} \tag{B5}$$

The width in the velocity space β_b is obtained simply as $\mathscr{A}_b/\alpha_b$.

The formulation above allows a very simple numerical discretization. The first two equations in Eq. (B5) have the usual structure of the Newton equations for the baricenter. The last two equations in Eq. (B5) also have a structure similar to the Newton equations, where α and γ play the role of conjugate variables. Equations (B5) can now be solved effectively with the leap-frog algorithm.

ACKNOWLEDGMENTS

The authors thank Ned Birdsall and Jerry Brackbill for useful advice and discussions, and Norman J. Zabusky for pointing out some similarities between the present work and the elliptical method for vortex dynamics. One of the authors (G.C.) thanks Professor Silvio Corno for suggesting the use of transformations involving the "most general act of motion." The work has been performed with the financial support of Italian Ministry of University and Scientific Research.

REFERENCES

1. G. Dipeso, E. C. Morse, and R. W. Ziolkowski, *J. Comput. Phys.* **96,** 325 (1991).
2. G. Lapenta and J. U. Brackbill, *J. Comput. Phys.* **115,** 213 (1994).
3. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation* (North Holland, New York, 1985).
4. G. G. M. Coppa, G. Lapenta, G. Dellapiana, and F. Donato, "Kinetic Plasma Simulation with Variable-Size Particles," in *15th Int. Conf. on the Numerical Simulation of Plasmas, Valley Forge, 1994.*
5. C. K. Birdsall and A. B. Langdon, *Plasma Physics Via Computer Simulation* (McGraw–Hill, New York, 1985).
6. R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (Hilger, Bristol, 1988).
7. W. B. Bateson, D. W. Hewett, and J. DeGroot, "GaPH—A New Smart Particle Algorithm for Economical Kinetic Modeling," in *ICOPS 95, Madison, 1995.*
8. H. L. Berk and K. V. Roberts, *Methods of Computational Physics,* Vol. 9 (Academic Press, San Diego, 1970), p. 87.
9. A. J. Chorin and P. Bernard, *J. Comput. Phys.* **13,** 423 (1973).
10. B. Legras and D. G. Dritschel, *Phys. Fluids A* **3,** 845 (1991).
11. G. G. M. Coppa, G. Lapenta, and V. Riccardo, *Phys. Plasmas,* June, 1996.
12. G. G. M. Coppa, G. Lapenta, and V. Riccardo, "Vlasov–Fokker–Planck Plasma Simulation Using the Blob Technique," in *ICOPS 95, Madison, 1995.*