



ELSEVIER

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Computer Physics Communications 166 (2005) 81–93

Computer Physics
Communications

www.elsevier.com/locate/cpc

Eulerian Vlasov codes

E. Pohn^a, M. Shoucri^{b,*}, G. Kamelander^a

^a *Forschungszentrum Seibersdorf, Austria*

^b *Institut de Recherche d'Hydro-Québec (IREQ), Varennes, Qué., Canada J3X1S1*

Received 7 October 2004; accepted 30 October 2004

Available online 8 December 2004

Abstract

Four different Eulerian grid-based Vlasov solvers are discussed, namely a second order method and a fourth order method (symplectic integrator) using cubic splines for interpolation, the CIP (cubic interpolated propagation) method, and an Euler–Lagrange method applying two-dimensional cubic interpolants. The four methods will be presented by outlining their algorithm. The performance of the numerical methods will be compared by numerically solving the Vlasov–Poisson system for the distribution function on a fixed Eulerian grid, for the problem of a two-stream instability in a two-dimensional phase-space.

© 2004 Elsevier B.V. All rights reserved.

Introduction

Numerical simulations of the Vlasov equation are of fundamental importance for the study of many nonlinear processes in kinetic plasmas. An important method for the numerical solution of the Vlasov–Poisson system is the direct solution of the Vlasov equation for the velocity distribution function as a partial differential equation in phase-space. This differential equation is discretized on a fixed Eulerian grid. Since Cheng and Knorr [1] proposed the second-order splitting scheme as an efficient time-integration method for the Vlasov–Poisson system, several Eulerian grid-based Vlasov solvers have been developed

(see the recent works in Refs. [2,3] and references therein). It is the purpose of this work to present four of these methods, and test their results by studying a problem of a two-stream instability in a two-dimensional phase-space, i.e. namely one spatial dimension x and one velocity dimension v .

Interest in Eulerian grid-based Vlasov solvers arise from the very low noise level associated with these methods, and the recent advances of parallel computers have increased the interest in the applications of splitting schemes to higher dimensional problems, since these splitting schemes adapt very well to computer with parallel architecture. Generalizations of these methods to higher dimensions have been presented a long time ago [4,5]. However, due to the higher computational resources demanded by these higher dimensional numerical simulations, it took a

* Corresponding author.

E-mail address: shoucri.magdi@ireq.ca (M. Shoucri).

long time for the applications of these methods to Eulerian Vlasov codes to be presented (see the recent references [6–10] and references therein).

It is clear in this context that some comparisons between different Eulerian grid-based Vlasov solvers are pertinent and certainly very useful. In the present work, we restrict our attention to a spatially one-dimensional problem. The first method we present is the second order splitting scheme of Cheng and Knorr [1]. The second method is a method based on a fourth-order symplectic integrator [2]. The third method is the CIP (cubic interpolated propagation) method [11]. All of these three methods apply a splitting scheme. The differences between these schemes are in the implementation of the advection in the x and v directions. The fourth method uses also an Eulerian grid but without splitting, so the advection is done simultaneously in space and velocity direction by solving for the orbit or the characteristics in the two-dimensional phase-space. These characteristics can be calculated either exactly or by approximation [4,8], or calculated by an iterative process [12], sometimes referred to in this last case as semi-Lagrangian. It is in fact an Eulerian method advecting simultaneously in the two directions along the characteristics on a fixed Eulerian grid. A better denomination of this method would be the Euler–Lagrange method, which we will adopt in this text, since essentially it uses an Eulerian grid. The advection of the Vlasov equation in multi-dimensions along the characteristics was discussed and proposed long time ago [4], but quite recently applied.

1. The problem equations

The performance of the numerical methods is evaluated by studying a two-stream instability in the two-dimensional phase-space $x-v$, where the pertinent Vlasov equation is:

$$\partial_t f + v \cdot \partial_x f - E_x \cdot \partial_v f = 0. \quad (1)$$

$f = f(x, v, t)$ and ∂ denominates partial derivatives. The spatial dimension x is assumed to be periodic. Eq. (1) for the electron distribution function is coupled

to Poisson's equation for the potential by:

$$\frac{\partial^2 \varphi}{\partial x^2} = -(1 - n_e), \quad \text{where } n_e = \int_{-\infty}^{\infty} f \, dv, \quad (2)$$

i.e. ions are assumed to form an immobile background. The electric field is calculated from the relation:

$$E = -\frac{\partial \varphi}{\partial x}. \quad (3)$$

2. The numerical methods

2.1. One-dimensional cubic splines

This group of numerical methods consists in fractional shifts which are applied to the distribution function. Considering a problem in one-dimensional space, a shift consists in advancing the distribution function in time by:

$$f(x, t + \Delta t) = f(x - g_x \Delta t, t), \quad (4)$$

where the factor g_x depends on the considered advection. In the present work, we limit ourselves to the case where the function value at the intermediate point $x - g_x \Delta t$ is calculated by interpolation using cubic splines. A recent work [13] contains a nice comparison between different interpolating polynomials where the cubic spline performed favorably. We also note recently the use of second degree polynomials in Ref. [3].

Cubic splines are interpolating functions with a cubic polynomial where the function values of the interpolants coincide with the function values at the grid-points, and the first and second derivatives of adjacent local interpolants are continuous at the grid-points. In a higher-dimensional problem the shifts become fractional, i.e. each of the dimension of the phase-space is shifted separately. The specific order, number of shifts and choice of the size of shift-factors depends now on the numerical method. In the case of a two-dimensional phase-space problem, the splitting is effected into two parts, one part dealing with the spatial derivative:

$$f^a(x, v, t + \Delta t) = f(x - c \cdot g_x \cdot \Delta t, v, t), \quad (5)$$

and the other part dealing with the derivative in velocity space:

$$f(x, v, t + \Delta t) = f^a(x, v - d \cdot g_v \cdot \Delta t, t), \quad (6)$$

where the shift factors are $g_x = v$ and $g_v = -E$. The factors c and d are constant and are chosen according to the specific numerical method.

2.1.1. 2nd order method (Cheng and Knorr)

A method of second order in time [1] is obtained by splitting Eq. (1) as follows:

- Solve $\partial_t f + v \cdot \partial_x f = 0$ for a step $\Delta t/2$, (7)

- Solve Poisson equation for the electric field,

- Solve $\partial_t f - E \cdot \partial_v f = 0$ for a step Δt , (8)

- Solve $\partial_t f + v \cdot \partial_x f = 0$ for a step $\Delta t/2$. (9)

This splitting has the advantage that each of the x or v updates is a linear advection effected by applying successively the shifts:

- $f^a(x, v, t + \Delta t) = f(x - c_1 \cdot g_x \Delta t, v, t)$, (10)

- $f^b(x, v, t + \Delta t) = f^a(x, v - d_1 \cdot g_v \Delta t, t)$, (11)

- $f(x, v, t + \Delta t) = f^b(x - c_2 \cdot g_x \Delta t, v, t)$, (12)

with $c_1 = \frac{1}{2}$, $c_2 = \frac{1}{2}$ and $d_1 = 1$. That is, half of the spatial shift is performed first, followed by the total shift in velocity space, and finally the second half of the spatial shift is done. This shifts are effected by calculating the shifted value using a cubic spline interpolation polynomial defined by assuming that the function, and its first and second derivatives are continuous at grid points. For the interval $(i, i + 1)$, this polynomial is given by:

$$P_i(x) = A_i f_i + B_i f_{i+1} + C_i s_i + D_i s_{i+1},$$

for $x_i < x < x_{i+1}$,

$$A_i = \frac{x_{i+1} - x}{\Delta x_i}, \quad B_i = \frac{x - x_i}{\Delta x_i}, \quad (13)$$

$$C_i = \frac{\Delta x_i^2}{6} (A_i^3 - A_i), \quad D_i = \frac{\Delta x_i^2}{6} (B_i^3 - B_i)$$

and $\Delta x_i = x_{i+1} - x_i$. The second derivative s_i is related to the value of the function at grid points f_i by:

$$\begin{aligned} & \Delta x_i s_{i+1} + 2(\Delta x_i + \Delta x_{i-1}) s_i + \Delta x_{i-1} s_{i-1} \\ & = 6 \left(\frac{f_{i+1} - f_i}{\Delta x_i} - \frac{f_i - f_{i-1}}{\Delta x_{i-1}} \right). \end{aligned}$$

This cubic polynomial is defined over three grid points. One can also use a cubic B-spline defined over four grid points (see Section 2.3 below), but the results were essentially the same. A brief comparison between these two cubic polynomials will be presented.

2.1.2. 4th order method (symplectic integrator)

A method of fourth order in time is obtained by applying successively the shifts [2]:

- $f^a(x, v, t + \Delta t) = f(x - c_1 \cdot g_x \Delta t, v, t)$, (14)

- $f^b(x, v, t + \Delta t) = f^a(x, v - d_1 \cdot g_v \Delta t, t)$, (15)

- $f^c(x, v, t + \Delta t) = f^b(x - c_2 \cdot g_x \Delta t, v, t)$, (16)

- $f^d(x, v, t + \Delta t) = f^c(x, v - d_2 \cdot g_v \Delta t, t)$, (17)

- $f^e(x, v, t + \Delta t) = f^d(x - c_3 \cdot g_x \Delta t, v, t)$ (18)

- $f^f(x, v, t + \Delta t) = f^e(x, v - d_3 \cdot g_v \Delta t, t)$, (19)

- $f(x, v, t + \Delta t) = f^f(x - c_4 \cdot g_x \Delta t, v, t)$ (20)

with

$$c_1 = c_4 = \frac{1}{2(2 - 2^{1/3})}, \quad c_2 = c_3 = \frac{1 - 2^{1/3}}{2(2 - 2^{1/3})},$$

$$d_1 = d_3 = \frac{1}{2 - 2^{1/3}}, \quad d_2 = -\frac{2^{1/3}}{2 - 2^{1/3}}.$$

The sum over all c_i and over all d_i is equal to 1 resulting altogether in a whole shift. This symplectic integrator method [2] can be applied in cases where the Hamiltonian is separable, as in the present case.

2.2. The CIP method

The CIP method [11] does not only solve the equation for the distribution function f , but additionally solves the equations for its partial derivatives $\partial_v f$ and $\partial_x f$. Proper equations for the derivatives are obtained by building partial derivatives of the considered problem-equation. Similar to the cubic spline method, local cubic interpolants are used for performing the shifts, but now no additional artificial assumptions (e.g., concerning the smoothness) have to be made for uniquely determining the interpolating polynomial, because the partial derivatives of f are now known, too. That is, the interpolant F is chosen in such a way that the function values as well as its derivatives coincide at the grid-points. One obtains:

$$F_i(x) = f_i + (\partial_x f_i)x + a_i x^2 + b_i x^3 \quad (21)$$

with

$$a_i = -\frac{2\partial_x f_i + \partial_x f_{i+1}}{\Delta x} - 3\frac{f_i - f_{i+1}}{\Delta x^2}, \quad (22)$$

$$b_i = \frac{\partial_x f_i + \partial_x f_{i+1}}{\Delta x^2} + 2\frac{f_i - f_{i+1}}{\Delta x^3} \quad (23)$$

for the interpolant of the distribution function f at interval i in the discretized space. A corresponding equation for the interpolation in velocity space is obtained. The interpolants for the partial derivatives are obtain from the equation:

$$(\partial_x F_i)(x) = \partial_x f_i + 2a_i x + 3b_i x^2. \quad (24)$$

Similar to the 2nd order cubic spline method, advancing the equation for a time-step Δt is done by first applying a spatial shift for half a time-step ($\Delta t/2$), then performing the shift in velocity space for a full time-step, and finally another spatial shift for half a time-step. But now, each of these steps is slightly more complicated than for the cubic splines method, due to the effect of the derivatives. Altogether the algorithm reads as follows, with $c_1 = \frac{1}{2}$ and $d_1 = 1$:

(1) first half of spatial shift

(a) advancing $\partial_v f$ by

$$\begin{aligned} (\partial_v f)_{ij} &= (\partial_v f)_{ij} \\ &- [v_{j+1}(\partial_x f)_{ij+1} - v_{j-1}(\partial_x f)_{ij-1}] \frac{\Delta t}{4\Delta v}, \end{aligned} \quad (25)$$

(b) advancing f by

$$\begin{aligned} f^a(x, v, t + \Delta t) \\ = f(x - c_1 \cdot v \Delta t, v, t), \end{aligned} \quad (26)$$

(c) advancing $\partial_x f$ by

$$\begin{aligned} (\partial_x f)^a(x, v, t + \Delta t) \\ = (\partial_x f)(x - c_1 \cdot v \Delta t, v, t), \end{aligned} \quad (27)$$

(d) advancing $\partial_v f$ (same as (a)); (28)

(2) total shift in velocity space

(a) advancing $\partial_x f$ by

$$\begin{aligned} (\partial_x f)_{ij} &= (\partial_x f)_{ij} \\ &- [E_{i+1}(\partial_v f)_{i+1j} - E_{i-1}(\partial_v f)_{i-1j}] \frac{\Delta t}{4\Delta x}, \end{aligned} \quad (29)$$

(b) advancing f by

$$\begin{aligned} f^b(x, v, t + \Delta t) \\ = f^a(x, v + d_1 \cdot E_x \Delta t, t), \end{aligned} \quad (30)$$

(c) advancing $\partial_v f$ by

$$\begin{aligned} (\partial_v f)^b(x, v, t + \Delta t) \\ = (\partial_v f)^a(x, v + d_1 \cdot E_x \Delta t, t), \end{aligned} \quad (31)$$

(d) advancing $\partial_x f$ (same as (a)); (32)

(3) second half of spatial shift (same as above for the first half).

2.3. Euler–Lagrange method with two-dimensional cubic B-splines

This method is also called semi-Lagrangian method [12]. We have explained in the introduction our preference to call this method Euler–Lagrange method, since it is essentially using an Eulerian grid and integrates the Vlasov equation along the characteristics. It differs from the other methods by the use of two-dimensional interpolation [4]. Thus, no splitting is necessary when a problem in a two-dimensional phase space is solved. Considering the characteristics of the problem equation presented, a integration scheme can be derived, which works as follows. Writing Eq. (1) as

$$\partial_t f + \vec{G} \cdot \begin{pmatrix} \partial_x f \\ \partial_v f \end{pmatrix} = 0 \quad \text{with } \vec{G} = \begin{pmatrix} v \\ -E \end{pmatrix}, \quad (33)$$

the Euler–Lagrange method calculates the shift factor \vec{d} from the equation:

$$\vec{d}_i = \Delta t \vec{G}(\vec{x}_i - \vec{d}_i, t_n) \quad (34)$$

with $\vec{x} = \begin{pmatrix} x \\ v \end{pmatrix}$ for each grid point i at the current time t_n .

Each time-step consists now in first calculating the shift factor iteratively using the iteration:

$$\vec{d}_i^{k+1} = \Delta t \vec{G}(\vec{x}_i - \vec{d}_i^k, t_n) \quad (35)$$

and then applying this factor to the single shift:

$$f(\vec{x}_i, t + \Delta t) = f(\vec{x}_i - 2\vec{d}_i, t - \Delta t). \quad (36)$$

The Euler–Lagrange method is a two-step method, which means that the distribution function at time $t + \Delta t$ is calculated using information from time t

(calculating the shift factor in Eq. (35)) as well as from time $t - \Delta t$ (performing the shift in Eq. (36)). The shift in Eq. (36) is effected using a 2D cubic B-spline [7,8,12], defined as a tensor product of one-dimensional B-splines:

$$s(x, v) = \sum_{i=0}^{N_x} \sum_{j=0}^{N_v} \eta_{ij} B_i(x) B_j(v), \quad (37)$$

where the B-splines are defined as:

$$B_i(x) = \frac{1}{6} \begin{cases} (x - x_i)^3 & x_i \leq x < x_{i+1}, \\ 1 + 3(x - x_{i+1}) + 3(x - x_{i+1})^2 - 3(x - x_i)^3 & x_{i+1} \leq x < x_{i+2}, \\ 1 + 3(x_{i+3} - x) + 3(x_{i+3} - x)^2 - 3(x_{i+3} - x)^3 & x_{i+2} \leq x < x_{i+3}, \\ (x_{i+4} - x)^3 & x_{i+3} \leq x < x_{i+4} \end{cases}$$

and $B_i(x) = 0$ otherwise. We have a similar definition for $B_j(v)$. For the calculation of the coefficients of the B-spline interpolation function $s(x, v)$, see details in Refs. [7,8].

Few iterations are necessary for the calculation of the shift-factor in the intermediate step in Eq. (35) to converge. A linear interpolation as well as a cubic interpolation have been used for interpolating \tilde{G} . The results we present use the cubic B-spline interpolation for the solution of Eq. (35), which, as we shall see, gives a more accurate result than the linear interpolation. It has also been shown in [8] for instance, that to an order $O(\Delta t^2)$, the displacement calculated in the iteration in Eq. (35) is equivalent to the integration of the orbit along the characteristics. An accurate discussion for the calculation of the characteristics for the 2D and multi-dimensional interpolation can be found in [4].

3. The advection term

All previously presented methods are based on advection solvers based on interpolation techniques. We consider first the simple one-dimensional equation:

$$\partial_t f + v \partial_x f = 0 \quad (38)$$

whose solution at a time t_n is given as a function of the initial condition by the relation:

$$f(x, v, t_n) = f(x - vt_n, v, 0). \quad (39)$$

If we consider at $t = 0$ an initial Maxwellian distribution perturbed by a small perturbation:

$$f(x, v, t = 0) = \frac{e^{-v^2/2}}{\sqrt{2\pi}} (1 + \varepsilon \cos(kx)), \quad (40)$$

then the charge density ρ at time t will be given by:

$$\begin{aligned} \rho(x, t) &= 1 - \int_{-\infty}^{\infty} f(x - vt, v, 0) dv \\ &= e^{-k^2 t^2 / 2} \varepsilon \cos(kx). \end{aligned} \quad (41)$$

The analytical solution is decaying exponentially in time. For the numerical solution, the density of the electrons is calculated at every spatial grid point by summation over all grid points in velocity space:

$$\begin{aligned} n_e(x, t) &= \sum_{j=0}^{N_v} f(x_i, v_j, t) = \sum_{j=0}^{N_v} f(x_i - v_j t, v_j, 0) \\ &= \sum_{j=0}^{N_v} f(x_i, v_j, 0) \varepsilon \cos(k(x_i - j \Delta v t)). \end{aligned} \quad (42)$$

So the exponential decay in Eq. (41) is replaced by a quasi-periodic function. When at a time $t = T_R$ we have $k(x_i - j \Delta v t) = kx_i + 2\pi n$, then the density n_e recovers exactly its value at $t = 0$. This recurrence time T_R is given by:

$$T_R = \frac{2\pi}{k \Delta v}. \quad (43)$$

We show in the following the results obtained from the second order splitting scheme using cubic spline interpolation and the second order CIP method. The Euler–Lagrange method, using two-dimensional interpolation, cannot be applied in the present one-dimensional problem. We use $\varepsilon = 0.1$. Two examples will be presented, with grid points $N_x \times N_v = 8 \times 32$ and $N_x \times N_v = 16 \times 32$. The length in space is $L_x = 4\pi$ and in velocity space we use $-5 \leq v \leq 5$. The recurrence time for this case is $T_R = 38.95$.

Fig. 1 shows for 8 grid points and 16 grid points in space respectively the time evolution of the peak value of $\rho(x, t)$ calculated analytically (dotted curve,

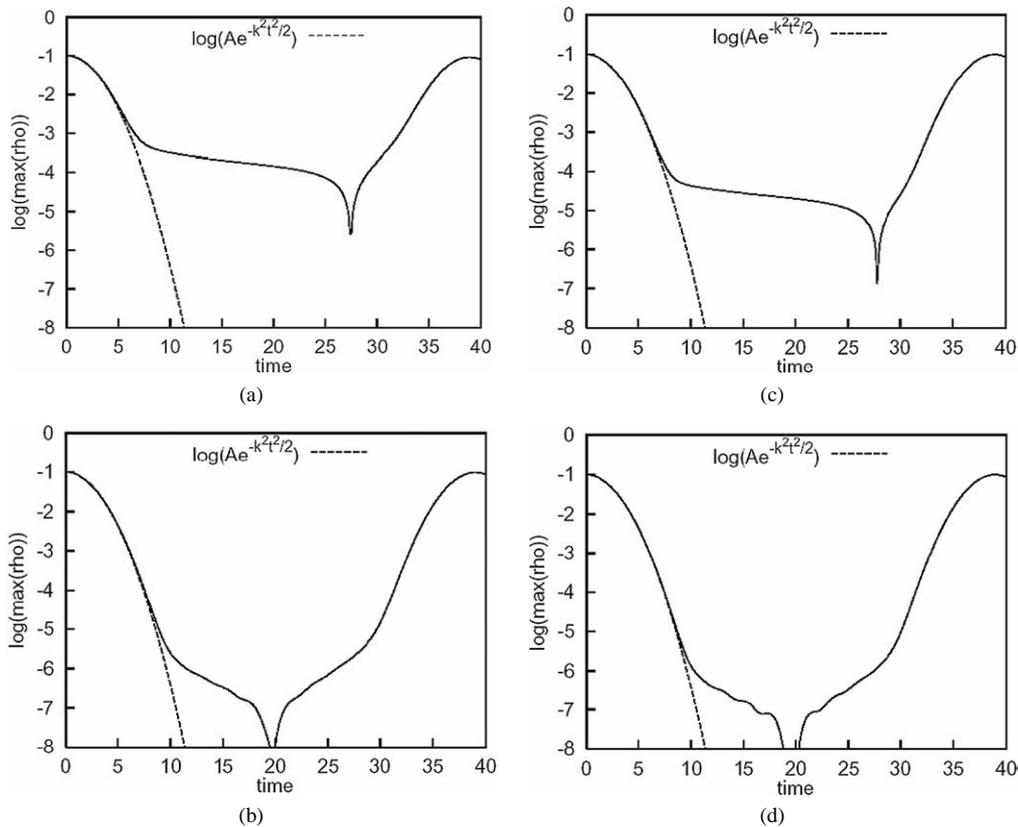


Fig. 1. Time evolution of the peak of ρ . (a) CIP method (8 grid points). (b) 2nd order cubic splines (8 grid points). (c) CIP method (16 grid points). (d) 2nd order cubic splines (16 grid points).

Eq. (41)), and numerically (full curve). The results for the CIP method are shown in Fig. 1(a), and the results for the second order splitting scheme with the cubic spline interpolation are shown in Fig. 1(b). One recognizes that the results in Fig. 1(b) appear more accurate. In the CIP-method the numerical curve detaches itself earlier in time from the analytical curve. The relative error at $t = 4.75$ in Fig. 1(a) is 0.1, while for the results in Fig. 1(b) the relative error of 0.1 is reached only at $t = 6.5$. In Fig. 1(b) the recurrence effect appears at $t = 39$, and reaches a peak of 0.0974, only 2.6% less than the initial peak of 0.1. In the results of the CIP method in Fig. 1(a), the recurrence time appears a time-step Δt earlier, at $t_n = 38.875$, and the amplitude of the peak at recurrence is 0.0917, about 8.3% less than the initial peak of 0.1.

Figs. 1(c)–(d) present the equivalent results for the case $N_x = 16$ grid points. In both cases the numeri-

cal results remain closer to the analytical results for a longer time. The recurrence effect is $t_n = 39$ in both cases. The error in the peak at the recurrence time is 0.95% in the CIP method and 0.48% in the second order splitting scheme with cubic spline interpolation.

Figs. 2(a)–(d) present the evolution of the numerical density $\rho(x, t)$ on a linear scale for 15 recurrence times for $N_x = 8$ grid points and for $N_x = 16$ grid points, for the two methods. In Fig. 2(a) for $N_x = 8$, the amplitude of ρ is only 38% of the original value for the CIP method after 15 recurrence times, while in Fig. 2(b) for the second order splitting scheme with cubic spline interpolation, the peak of ρ is still 73% of the original peak. Figs. 2(c)–(d) show the results for the two methods for $N_x = 16$. The second order cubic spline splitting scheme is still giving a better conservation of the peak after 15 recurrence times.

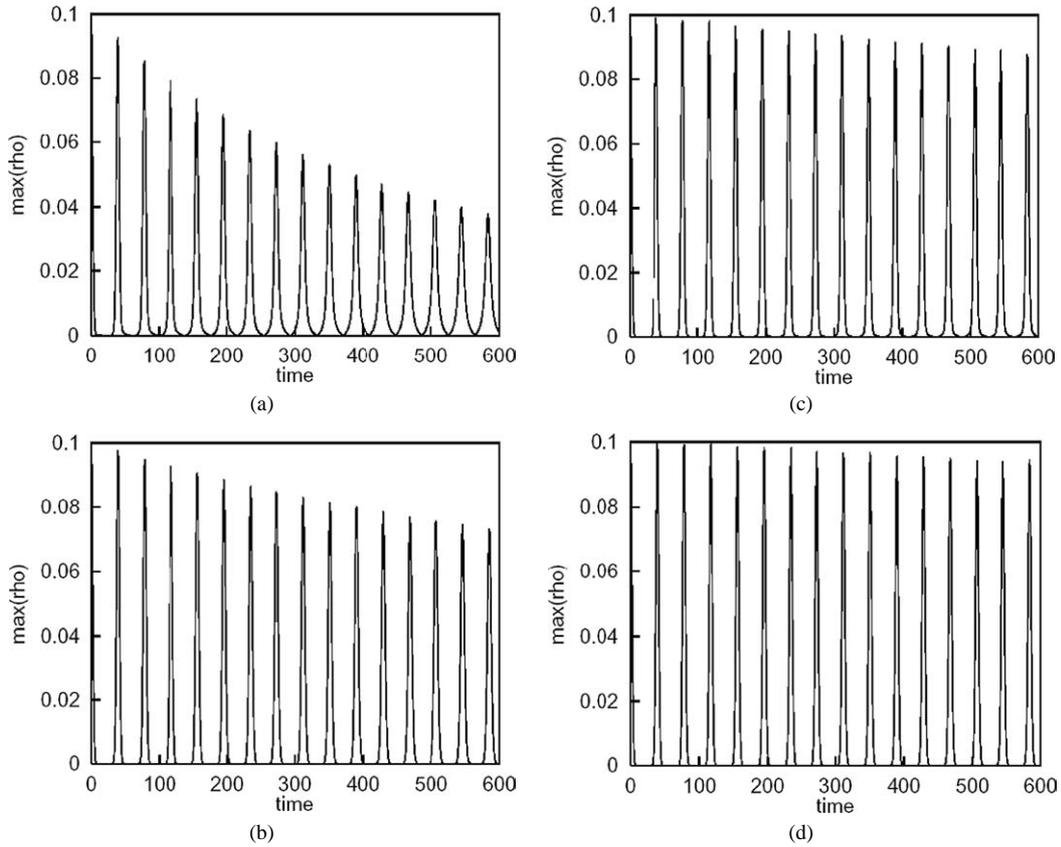


Fig. 2. Time evolution of the peak of ρ . (a) CIP method (8 grid points). (b) 2nd order cubic splines (8 grid points). (c) CIP method (16 grid points). (d) 2nd order cubic splines (16 grid points).

4. The two-stream instability

We consider the interval $0 \leq x < 4\pi$ in the spatial direction and $-5 \leq v \leq 5$ in velocity space.

The initial condition is:

$$f(x, v, t = 0) = v^2 \frac{e^{-v^2/2}}{\sqrt{2\pi}} (1 + \varepsilon \cos(kx)). \quad (44)$$

This function has two maxima in velocity space at $v = \pm\sqrt{2}$, and a periodic spatial perturbation. The parameters of the perturbation are $\varepsilon = 0.05$ and $k = 0.5$. A grid of $N_x \times N_v = 32 \times 256$ points is used. The time-step is $\Delta t = 1/8$. The cubic spline in Eq. (13) is used with the second order and fourth order splitting schemes. The cubic polynomials in Eqs. (21)–(24) are used with the CIP method. The tensor product of cubic B-splines in Eq. (37) is used with the Euler–Lagrange method.

In Fig. 3 we present on a logarithmic scale the results obtained for the time evolution of the four harmonics E_1, E_2, E_3 and E_4 of the electric field for the four different methods. There is a close agreement for the first two harmonics. Figs. 3(e)–(f) show the time evolution of the modes E_3 and E_4 in a magnified area. The oscillations of the second order splitting scheme appear more important for the higher harmonics E_3 and E_4 , but the CIP method deviates completely. Note the good agreement for the higher harmonics between the fourth order splitting scheme and the Euler–Lagrange method.

The time evolution of the electric energy:

$$\text{E.E.} = \frac{1}{2} \frac{1}{L_x} \int_0^{L_x} E^2 dx, \quad (45)$$

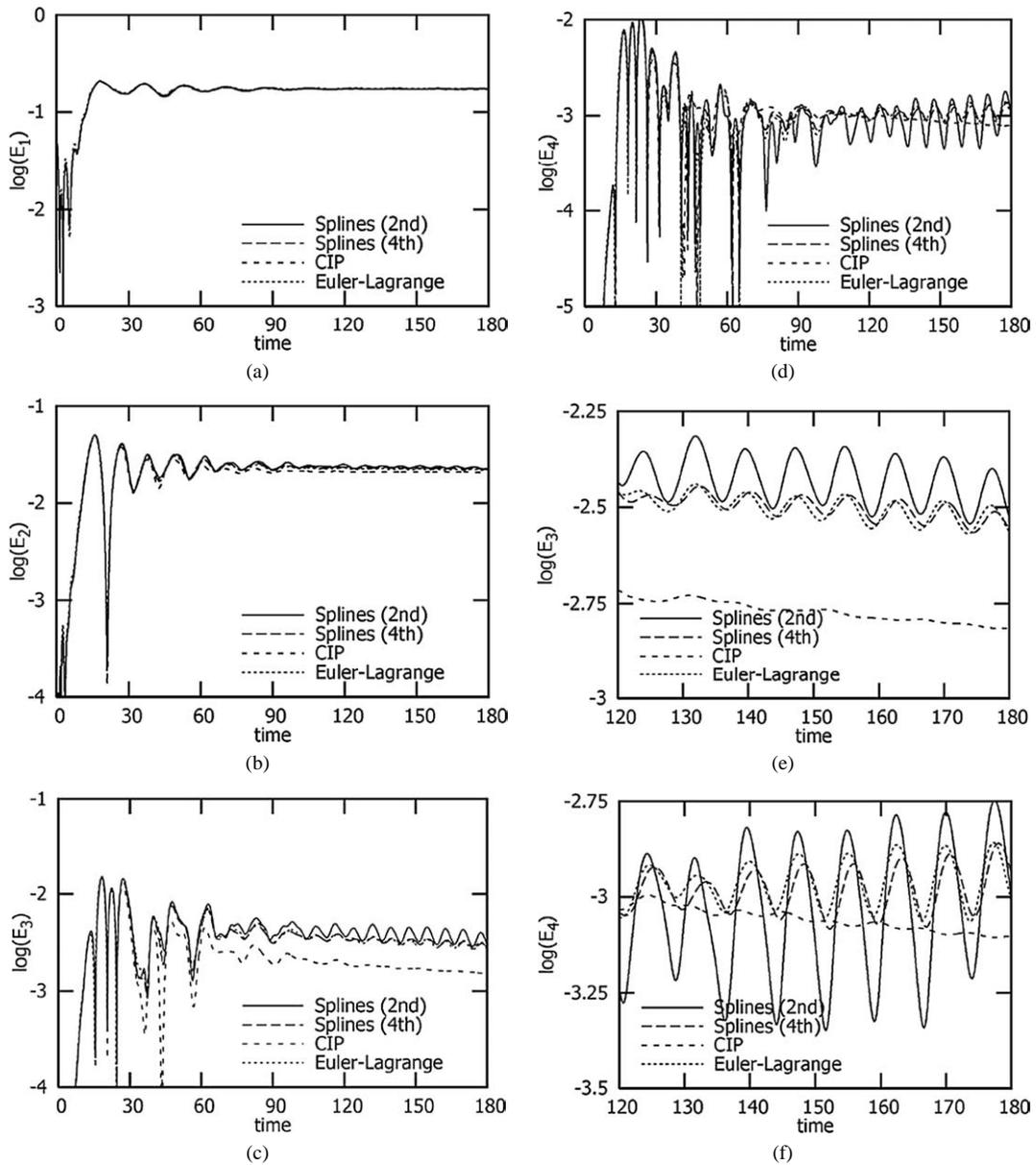


Fig. 3. Time evolution of the Fourier mode E_1 (a), E_2 (b), E_3 (c), E_4 (d). Time evolution (magnified) of Fourier mode E_3 (e), E_4 (f).

is presented in Fig. 4. Fig. 5 presents the time evolution of the kinetic energy,

$$\text{K.E.} = \frac{1}{2} \frac{1}{L_x} \int_0^{L_x} \int_{-\infty}^{\infty} v^2 f(x, v, t) dx dv \quad (46)$$

and the total energy is presented in Fig. 6.

Figs. 4(b), 5(b) and 6(b) show the results of the second order cubic spline and fourth order cubic spline methods following each other closely. The CIP method and the Euler–Lagrange method are not far, but the oscillations of the CIP method have its peaks more smoothed due to a numerical diffusion associated with the method [14], and showing even a difference where

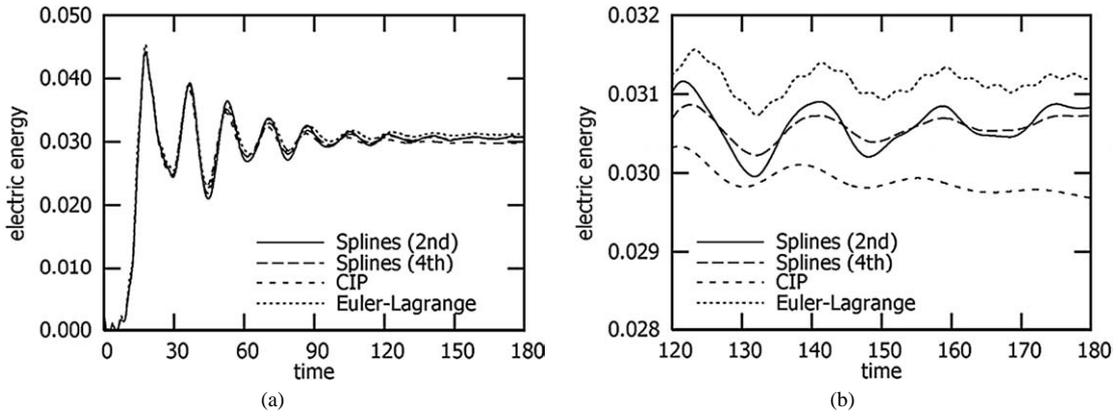


Fig. 4. (a) Time evolution of the electric energy, (b) time evolution of the electric energy (magnified scale).

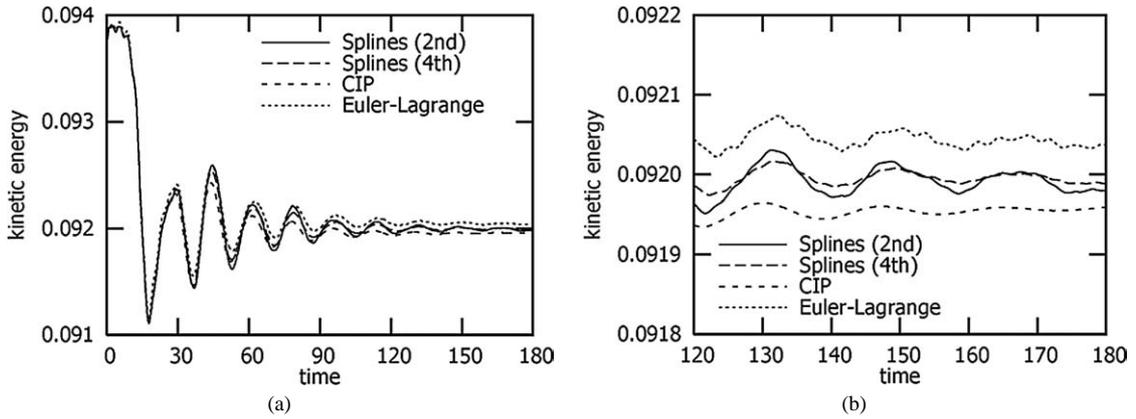


Fig. 5. (a) Time evolution of the kinetic energy, (b) time evolution of the total energy (magnified scale).

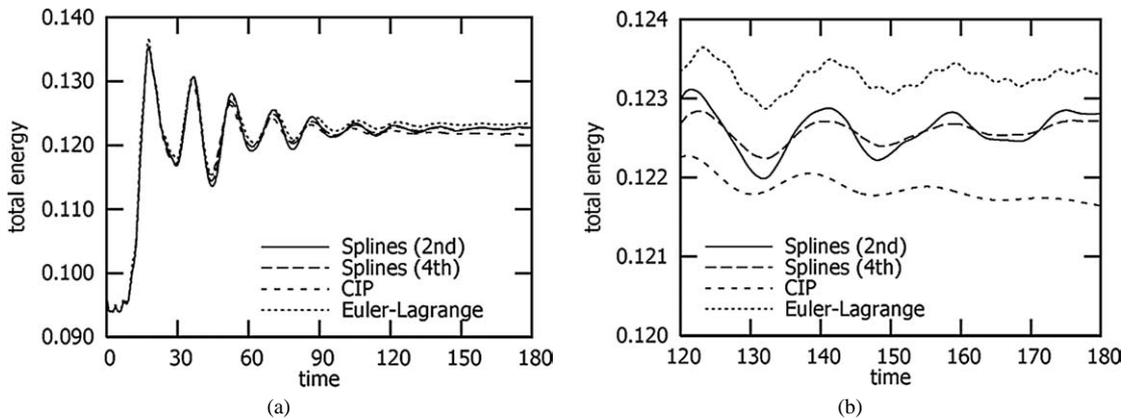


Fig. 6. (a) Time evolution of the total energy, (b) time evolution of the total energy (magnified scale).

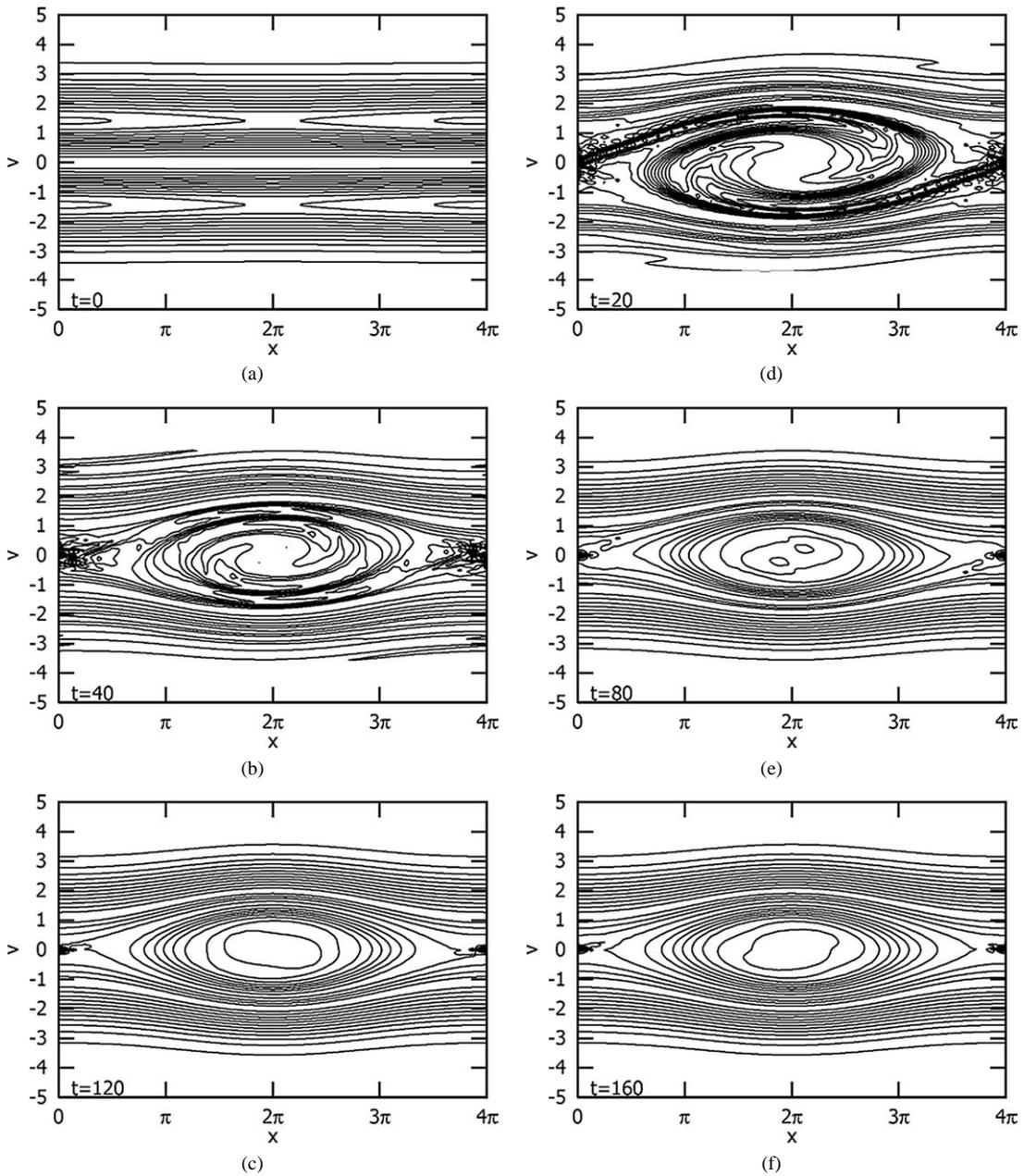


Fig. 7. Phase space contour plots for the evolution of the two-stream instability, calculated from 2nd order splines.

the maxima and minima appear with respect to the other methods. Note also a small modulation in the evolution of the curve associated with the Euler–Lagrange method, probably due to the iterative step in Eq. (35). Fig. 7 presents some sequences of the

phase-space evolution with a final BGK structure appearing, obtained using the second order cubic splines method.

A further insight in the difference between these four methods is presented in Fig. 8 which shows the

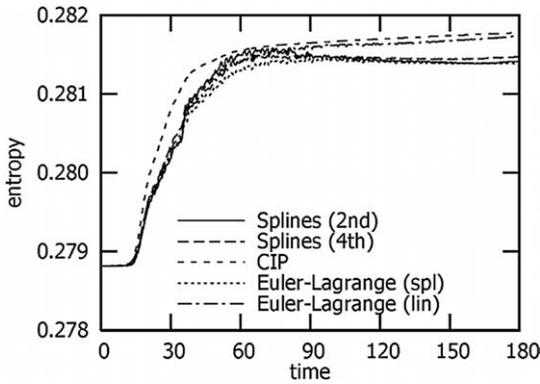


Fig. 8. Time evolution of the entropy.

time evolution of the entropy:

$$S = -\frac{1}{L_x} \int_0^{L_x} \int_{-\infty}^{\infty} f \ln f \, dx \, dv \quad (47)$$

associated with the four methods. When the microstructure was initially evolving, the entropy was increasing. The entropy curve obtained with the CIP method grows more rapidly than the entropy curves of the other methods, underlining a more rapid cleaning of the initial microstructure, and then shows a continuous small growth without reaching saturation, underlining the numerical dissipation associated with the CIP method. The entropy curves of the Euler–Lagrange method and the second order and fourth order cubic spline methods show similar evolution. After the initial rapid growth phase, the curves clearly show a saturation indicating that an equilibrium has been reached. When the microstructure starts to be cleaned, the final entropy remains constant, indicating that these three methods give a good conservation of the equilibrium. We have added however, for comparison, the curve obtained for the entropy of the Euler–Lagrange method when a linear interpolation is used for the solution of Eq. (35) (denoted by (lin) in Fig. 8). The curve in this case follows the entropy curve of the CIP method, underlining the numerical diffusion associated with the Euler–Lagrange method in this case. Hence the importance of using cubic spline for the iterative step in Eq. (35).

Figs. 9(a)–(c) show the variation in electron-density:

$$\Delta n = 100 \frac{\int_0^{L_x} n(x, t) \, dx - \int_0^{L_x} n(x, 0) \, dx}{\int_0^{L_x} n(x, 0) \, dx}, \quad (48)$$

during the simulation, comparing the density conservation of the different numerical methods. The second and fourth order splitting methods using cubic spline and the Euler–Lagrange method conserve the density best (Figs. 9(a)–(b)). The integral over density for the second order spline method increases from 1 ($t = 0$) to less than 1.000018 ($t = 180$), i.e. the percentage increase is about 0.0018%. The percentage density increase for the fourth order cubic spline method is less than 0.0025%, and the Euler–Lagrange method in Fig. 9(b) gives a percentage density increase of 0.0015%. For the CIP method, the density increases much more, about 0.037% in Fig. 9(c), especially during the initial phase when the microstructures evolve. We also show in Fig. 9(b) the percentage density increase when the second order splitting method is solved using cubic B-splines, instead of the cubic spline in Eq. (13). The percentage density increase (shown in Fig. 9(b), with the notation B-spline (2nd)) is now 0.0014% instead of 0.0018% in Fig. 9(a). The other curves presented in Figs. 3–6 for the second order splitting method remained essentially the same when the cubic B-spline is used. Fig. 9(c) also demonstrates the important role of the iteration in Eq. (35) for calculating the shift-factors \bar{d}_i . Linear interpolation as well as cubic splines were compared for calculating \bar{G} in Eq. (35) at the intermediate points. The effect on density conservation is drastic. When using linear interpolation, the Euler–Lagrange method shows a linear increase of density without stabilization even when reaching the equilibrium (Fig. 9(c)), whereas the cubic splines interpolation results in very good conservation (Fig. 9(a)). This is in agreement with previously reported results [15] showing poor performance of the Euler–Lagrange method when linear interpolation is used for the solution of Eq. (35).

For the computational time required to do this simulation, the fourth order cubic spline method and the CIP method were a factor of 2.3 slower than the second order cubic spline method, and the Euler–Lagrangian method was a factor of 6 slower than the second order cubic spline method, because of the iterative steps involved in Eq. (35).

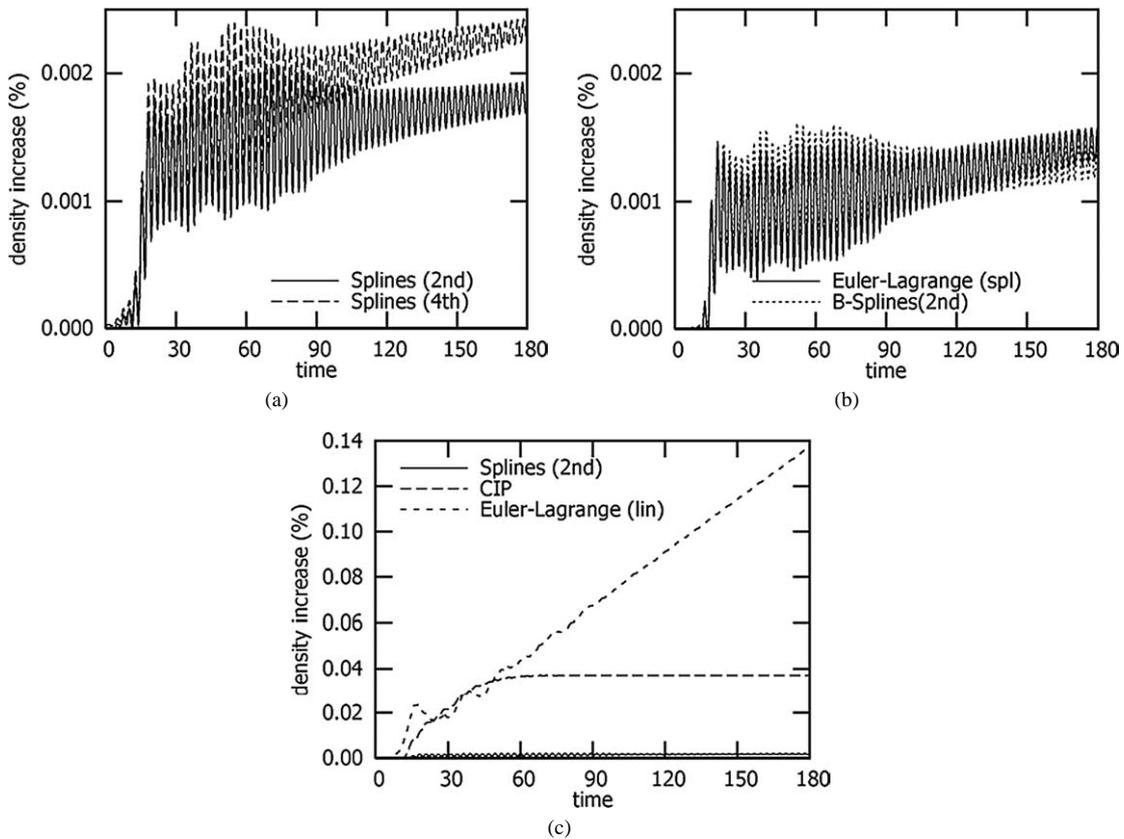


Fig. 9. Density increase in percent.

5. Conclusions

We have presented results obtained using four Eulerian grid-based Vlasov solvers, when studying a problem of a two-stream instability. Differences observed are sufficient to draw some general conclusions, which help complement some previously obtained results [14,15]. The entropy of the fourth order cubic spline symplectic integrator method shows a flat saturation of the entropy curve after the initial rapid cleaning of the microstructure, indicating nicely the conservation of the equilibrium. The second order cubic spline method and Euler–Lagrange method follow closely the entropy curve of the fourth order symplectic integrator method. All three methods show a nice conservation of the density. For the Euler–Lagrange method however, this requires the use of cubic B-splines interpolation for the iterative step in Eq. (35), in addition to the use of cubic B-splines

for the interpolation in Eq. (36). Note also the small modulation superimposed on the energy curve of the Euler–Lagrange method in Fig. 6, probably the result of the iterative step in Eq. (35). A linear interpolation in Eq. (35) leads to a poor performance of the Euler–Lagrange method, especially concerning the conservation of density, where the error shows a continuous linear growth as in Fig. 9(c), and the continuous increase in the entropy curve associated with a numerical diffusion (curve Euler–Lagrange (lin) in Fig. 8), a result previously pointed out in Ref. [15]. The continuous growth of the entropy curve observed with the CIP method in Fig. 8 clearly underlines a numerical dissipation associated with this method (a result previously pointed out in Ref. [14]). Note in Figs. 3(e) and (f) the deviation of the CIP curves from the other curves. Also, the density conservation of the CIP method in Fig. 9(c) was a factor of about 15 lower than the other methods. Finally we note that the methods we

presented and discussed are quite general. Recent applications to the fluid equations of the shallow water, of the methods of fractional steps associated with integration along the characteristics, of the associated Riemann invariants using cubic splines interpolation [16] have produced very good results.

Acknowledgements

M. Shoucri is grateful to Professor C.-Z. Cheng, Princeton University, for his interest and many fruitful discussions.

References

- [1] C.Z. Cheng, G. Knorr, *J. Comput. Phys.* 22 (1976) 330; M. Shoucri, et al., *J. Comput. Phys.* 24 (1977) 445.
- [2] T. Watanabe, H. Sugama, in: *Proc. of the Vlasovia Conf.* Nancy, November 2003.
- [3] T.D. Arber, R.G.L. Vann, *J. Comput. Phys.* 180 (2002) 339.
- [4] C.Z. Cheng, *J. Comput. Phys.* 24 (1977) 348.
- [5] M. Shoucri, *IEEE Trans. Plasma Sci.* PS-7 (1979) 69; See also: Vlasov equation, a new efficient and structured approach, in: *Proc. 10th Annual Pittsburg Conf., April 25–27, Modeling and Simulation*, vol. 10, part 3, School of Engineering, University of Pittsburg (Ed. Instrument Society of America), 1979, p. 1887.
- [6] M. Manfredi, M. Shoucri, et al., *Phys. Scripta* 58 (1998) 159.
- [7] E. Pohn, *Numerische Simulation des anomalen Transportes in der Plasmarandschicht*, Ph.D. Dissertation, Technische Universität Wien, Austria, 2001.
- [8] M. Shoucri, H. Gerhauser, K.-H. Finken, *Comput. Phys. Comm.* 164 (2004) 138.
- [9] M.L. Begué, A. Ghizzo, P. Bertrand, *J. Comp. Phys.* 151 (1999) 458.
- [10] T. Watanabe, H. Sugama, T. Sato, *J. Phys. Soc. Japan* 70 (2001) 3565.
- [11] T. Nakamura, T. Yabe, *Comput. Phys. Comm.* 120 (1999) 122.
- [12] E. Sonnendrücker, J. Roche, P. Bertrand, A. Ghizzo, *J. Comput. Phys.* 149 (1999) 201.
- [13] A. Mangeney, F. Califano, C. Cavazzoni, P. Travnicek, *J. Comput. Phys.* 179 (2002) 457.
- [14] E. Pohn, M. Shoucri, G. Kamelander, *Comput. Phys. Comm.* 137 (2001) 396.
- [15] M. Shoucri, *Czech. J. Phys.* 51 (2001) 1139.
- [16] M. Shoucri, *Comput. Phys. Comm.* 164 (2004) 396.