ISSN 1063-7842, Technical Physics, 2014, Vol. 59, No. 3, pp. 318–324. © Pleiades Publishing, Ltd., 2014.
Original Russian Text © A.A. Kuraev, A.O. Rak, S.V. Kolosov, A.A. Koronovskii, A.E. Hramov, 2014, published in Zhurnal Tekhnicheskoi Fiziki, 2014, Vol. 84, No. 3, pp. 8–13.

THEORETICAL AND MATHEMATICAL PHYSICS

Fast Algorithm for Numerically Integrating Equations of Motion for Large Particles in Microwave Devices

A. A. Kuraev[a, b], A. O. Rak[a], S. V. Kolosov[a], A. A. Koronovskii[b, c], and A. E. Hramov[b, c]*

[a] *Belarusian State University of Informatics and Radioelectronics, ul. P. Brovki 6, Minsk, 220013 Belarus*
[b] *Chernyshevsky State University, Bol'shaya Kazach'ya ul. 112, Saratov, 410012 Russia*
[c] *Gagarin State Technical University, Politekhnicheskaya ul. 77, Saratov, 410054 Russia*
**e-mail: hramovae@gmail.com*

Received April 23, 2013

Abstract—An algorithm for numerically integrating the equations of motion for large particles is formulated. The algorithm is represented in three forms—an explicit form, a predictor—corrector form, and a modified form that is intended for simulating the motion of relativistic particles in an electromagnetic field. The order of precision of the proposed algorithm is analyzed. The results of a comparison with standard algorithms are presented. The efficiency of this algorithm is demonstrated in considering the test example of a harmonic oscillator and in numerically simulating the Pierce diode in the mode of virtual-cathode formation.

DOI: 10.1134/S1063784214030165

INTRODUCTION

State-of-the-art methods for numerically simulating and analyzing processes accompanying the interaction of electron and ion flows with electromagnetic fields play an ever increasing role in modern electronics and plasma physics [1–7]. In simulating beam–plasma systems, various particle-method modifications, which make it possible to solve efficiently relevant problems [1, 7], are some of the most promising and widely used approaches. We emphasize that, in simulating electron flows interacting with electromagnetic fields by using any modification of the particle-method, it is necessary to solve numerically the equations of motion for large particles.

The rate and accuracy of these calculations play an especially important role in employing particle-in-cell (PIC) methods in which case it is necessary to calculate many times trajectories and velocities for millions of particles. In the present article, we describe an efficient method intended for numerically integrating equations of motion and based on the ideas put forth in [8].

In the nonrelativistic case, the motion of a large particle is described by Newton's second law:

$$a = \frac{F}{m}.$$

Basic parameters that describe the state of a particle are its coordinate $x(t)$ and its velocity $v(t)$ and acceleration $a(t)$.

These quantities are related by the equations

$$\dot{x} = v,$$

$$\ddot{x} = \dot{v} = a.$$

1. FORMULATION OF THE ALGORITHM

We define the values of a function and its even and odd derivatives within different time layers [1], as one can see in the figure. Accordingly, we have

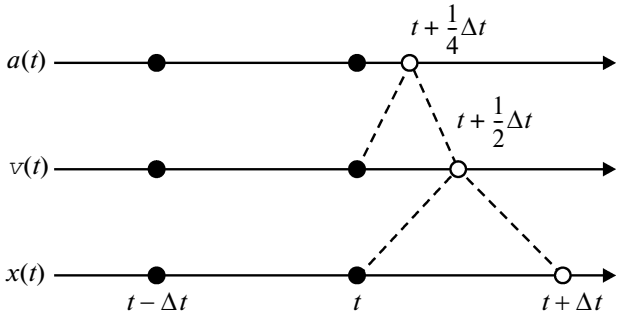
$$\frac{x_{t+\Delta t} - x_t}{\Delta t} = v_{t+\frac{1}{2}\Delta t}, \quad (1)$$

$$x_{t+\Delta t} = x_t + v_{t+\frac{1}{2}\Delta t}\Delta t, \quad (2)$$

$$\frac{v_{t+\frac{1}{2}\Delta t} - v_t}{\frac{1}{2}\Delta t} = a_{t+\frac{1}{4}\Delta t}, \quad (3)$$

$$v_{t+\frac{1}{2}\Delta t} = v_t + \frac{1}{2}a_{t+\frac{1}{4}\Delta t}\Delta t.$$

The value of $a_{t+\frac{1}{4}\Delta t}$ is extrapolated by using the acceleration values at the preceding two time steps ($a_{t-\Delta t}$ and a_t); that is,



Disposition of time steps in the finite-difference scheme being considered.

$$a_{t+\frac{1}{4}\Delta t} = \frac{1}{4}(5a_t - a_{t-\Delta t}). \quad (4)$$

Substituting Eqs. (4) and (3) into Eq. (2), we obtain an expression for $x_{t+\Delta t}$ in the form

$$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{8}(5a_t - a_{t-\Delta t})\Delta t^2.$$

In order to refine the value of $x_{t+\Delta t}$, we interpolate $a_{t+\frac{1}{4}\Delta t}$ by using the values of a_t and $a_{t+\Delta t}(x_{t+\Delta t})$; that is,

$$a_{t+\frac{1}{4}\Delta t} = \frac{1}{4}(a_{t+\Delta t} + 3a_t). \quad (5)$$

Substituting Eqs. (5) and (3) into Eq. (2), we obtain the following expression for the corrector of the coordinate:

$$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{8}(a_{t+\Delta t} + 3a_t)\Delta t^2. \quad (6)$$

We now find an expression for $v_{t+\Delta t}$ by using $a_{t+\Delta t}(x_{t+\Delta t})$. As a result, we arrive at

$$\frac{v_{t+\Delta t} - v_{t+\frac{1}{2}\Delta t}}{\frac{1}{2}\Delta t} a_{t+\frac{3}{4}\Delta t}, \quad (7)$$

$$a_{t+\frac{3}{4}\Delta t} = \frac{1}{4}(3a_{t+\Delta t} + a_t). \quad (8)$$

By using Eqs. (1) and (8), we recast expression (7) into the form

$$v_{t+\Delta t}\Delta t = x_{t+\Delta t} - x_t + \frac{1}{8}(3a_{t+\Delta t} + a_t)\Delta t^2. \quad (9)$$

Ultimately, the algorithm in the predictor–corrector form is specified as follows.

The predictor is

$$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{8}(5a_t - a_{t-\Delta t})\Delta t^2. \quad (10)$$

The corrector is given by

$$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{8}(a_{t+\Delta t} + 3a_t)\Delta t^2, \quad (11)$$

$$v_{t+\Delta t}\Delta t = x_{t+\Delta t} - x_t + \frac{1}{8}(3a_{t+\Delta t} + a_t)\bar{\Delta} t^2. \quad (12)$$

The number of iterations in the corrector can be controlled by the difference of the values of $x_{t+\Delta t}$ at different correction steps.

Employing Eq. (10) to determine the coordinate and substituting Eq. (10) into Eq. (12), we recast the algorithm into an explicit form; that is,

$$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{8}(5a_t - a_{t-\Delta t})\Delta t^2, \quad (13)$$

$$v_{t+\Delta t} = v_t + \frac{1}{8}(3a_{t+\Delta t} + 6a_t - a_{t-\Delta t})\Delta t. \quad (14)$$

The algorithms specified by Eqs. (10)–(14) can be used only in the case where a_t is independent of v_t. If electrons move in a magnetic field or at relativistic velocities, this condition does not hold. In such cases, it is also necessary to predict the value of $v_{t+\Delta t}$; that is,

$$\frac{v_{t+\Delta t} - v_t}{\Delta t} = a_{t+\frac{1}{2}\Delta t} = \frac{1}{2}(3a_t - a_{t-\Delta t}),$$
$$v_{t+\Delta t} = v_t + \frac{1}{2}(3a_t - a_{t-\Delta t})\Delta t. \quad (15)$$

In order to correct the value in (15), we can use expression (14).

The algorithm that we constructed here is rather close to Beeman's (Beeman–Schofield's) algorithm. For a comparison to be convenient, both algorithms are presented in Table 1.

2. DETERMINATION OF THE ORDER OF PRECISION OF THE ALGORITHM

We now determine the order of precision of the proposed algorithm. For this, we employ a Taylor series in basic expressions. For expression (2), we represent $x_{t+\Delta t}$ and x_t in the form

$$x_{t+\Delta t} = x_{t+\frac{1}{2}\Delta t+\frac{1}{2}\Delta t} = x_{t+\frac{1}{2}\Delta t} + \frac{1}{2}v_{t+\frac{1}{2}\Delta t}\Delta t + \frac{1}{8}a_{t+\frac{1}{2}\Delta t}\Delta t^2 + \frac{1}{48}b_{t+\frac{1}{2}\Delta t}\Delta t + O(\Delta t^4), \quad (16)$$

$$x_t = x_{t+\frac{1}{2}\Delta t-\frac{1}{2}\Delta t} = x_{t+\frac{1}{2}\Delta t} - \frac{1}{2}v_{t+\frac{1}{2}\Delta t}\Delta t + \frac{1}{8}a_{t+\frac{1}{2}\Delta t}\Delta t^2 - \frac{1}{48}b_{t+\frac{1}{2}\Delta t}\Delta t + O(\Delta t^4). \quad (17)$$

Subtracting Eq. (17) from Eq. (16), we arrive at

$$x_{t+\Delta t} = x_t + v_{t+\frac{1}{2}\Delta t}\Delta t + \frac{1}{24}b_{t+\frac{1}{2}\Delta t}\Delta t^3 + O(\Delta t^4). \quad (18)$$

We now perform similar transformations for expression (3). We have

$$v_{t+\frac{1}{2}\Delta t} = v_{t+\frac{1}{4}\Delta t+\frac{1}{4}\Delta t} = v_{t+\frac{1}{4}\Delta t} + \frac{1}{4}a_{t+\frac{1}{4}\Delta t}\Delta t + \frac{1}{32}b_{t+\frac{1}{4}\Delta t}\Delta t^2 + O(\Delta t^3),$$

$$v_t = v_{t+\frac{1}{4}\Delta t-\frac{1}{4}\Delta t} = v_{t+\frac{1}{4}\Delta t} - \frac{1}{4}a_{t+\frac{1}{4}\Delta t}\Delta t + \frac{1}{32}b_{t+\frac{1}{4}\Delta t}\Delta t^2 + O(\Delta t^3).$$

After the subtraction, we obtain

$$v_{t+\frac{1}{2}\Delta t} = v_t + \frac{1}{2}a_{t+\frac{1}{4}\Delta t}\Delta t + O(\Delta t^3). \quad (19)$$

For expression (4), we have

$$a_{t+\frac{1}{4}\Delta t} = a_t + \frac{1}{4}b_t\Delta t + O(\Delta t^2), \quad (20)$$

$$a_{t-\Delta t} = a_t - b_t\Delta t + O(\Delta t^2). \quad (21)$$

Table 1. Basic relations for the proposed algorithm and for Beeman's algorithm

Our algorithm	Beeman's algorithm
Predictor–corrector form	
Predictor	Predictor
$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{8}(5a_t - a_{t-\Delta t})\Delta t^2 + O(\Delta t^3)$	$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{6}(4a_t - a_{t-\Delta t})\Delta t^2 + O(\Delta t^4)$
Corrector	Corrector
$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{8}(a_t + 3a_t)\Delta t^2 + O(\Delta t^3)$	$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{6}(a_t + 2a_t)\Delta t^2 + O(\Delta t^4)$
$v_{t+\Delta t} = \frac{x_{t+\Delta t} - x_t}{\Delta t} + \frac{1}{8}(3a_{t+\Delta t} + a_t)\Delta t + O(\Delta t^2)$	$v_{t+\Delta t} = \frac{x_{t+\Delta t} - x_t}{\Delta t} + \frac{1}{6}(2a_{t+\Delta t} + a_t)\Delta t + O(\Delta t^3)$
Explicit form	
$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{8}(5a_t - a_{t-\Delta t})\Delta t^2 + O(\Delta t^3)$	$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{6}(4a_t - a_{t-\Delta t})\Delta t^2 + O(\Delta t^4)$
$v_{t+\Delta t} = v_t + \frac{1}{8}(3a_{t+\Delta t} + 6a_t - a_{t-\Delta t})\Delta t + O(\Delta t^3)$	$v_{t+\Delta t} = v_t + \frac{1}{6}(2a_{t+\Delta t} + 5a_t - a_{t-\Delta t})\Delta t + O(\Delta t^3)$
Modified predictor–corrector form	
Predictor	Predictor
$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{8}(5a_t - a_{t-\Delta t})\Delta t^2 + O(\Delta t^3)$	$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{6}(4a_t - a_{t-\Delta t})\Delta t^2 + O(\Delta t^4)$
$v_{t+\Delta t} = v_t + \frac{1}{2}(3a_t - a_{t-\Delta t})\Delta t + O(\Delta t^3)$	$v_{t+\Delta t} = v_t + \frac{1}{2}(3a_t - a_{t-\Delta t})\Delta t + O(\Delta t^3)$
Corrector	Corrector
$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{8}(a_{t+\Delta t} + 3a_t)\Delta t^2 + O(\Delta t^3)$	$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{6}(a_{t+\Delta t} + 2a_t)\Delta t^2 + O(\Delta t^4)$
$v_{t+\Delta t} = v_t + \frac{1}{8}(3a_{t+\Delta t} + 6a_t - a_{t-\Delta t})\Delta t + O(\Delta t^3)$	$v_{t+\Delta t} = v_t + \frac{1}{6}(2a_{t+\Delta t} + 5a_t - a_{t-\Delta t})\Delta t + O(\Delta t^3)$

By using the representations in (20) and (21), we express $a_{t+\Delta t}$ in two ways: first, we multiply Eq. (20) by the factor of 4 and add Eq. (21) to the result; second, we multiply Eq. (20) by the factor of 3 and add Eq. (21) to the result. We then obtain the expressions

$$a_{t=\frac{1}{4}\Delta t} = \frac{1}{4}(5a_t - a_{t-\Delta t}) + O(\Delta t^2), \quad (22)$$

$$a_{t+\frac{1}{4}\Delta t} = \frac{1}{3}(4a_t - a_{t-\Delta t}) - \frac{1}{21}b_t\Delta t + O(\Delta t^2). \quad (23)$$

Substituting in turn Eqs. (22) and (23) into Eq. (19) and thereupon the result into Eq. (18), we obtain expressions for the predicted particle coordinate in the form

$$x_{t+\Delta t} = x_t v_1\Delta t + \frac{1}{8}(5a_t - a_{t-\Delta t})\Delta t^2 + \frac{1}{24}b_t\Delta t^3 + O(\Delta t^4) = x_t + v_1\Delta t + \frac{1}{8}(5a_t - a_{t-\Delta t})\Delta t^2 + O(\Delta t^3), \quad (24)$$

$$x_{t+\Delta t} = x_t + v_1\Delta t + \frac{1}{6}(4a_t - a_{t-\Delta t})\Delta t^2 + O(\Delta t^4). \quad (25)$$

Expression (25) corresponds to Beeman's algorithm [8]. From Eq. (24), one can see that the proposed algorithm determines the position to the third order of precision, but, at small values of Δt and b_t (this depends on the character of motion), the term $\frac{1}{24}b_t\Delta t^3$ may tend to zero (for example, in the case of motion at a constant acceleration). It should also be noted that the coefficient $1/8 = 0.125$, which is a decimal fraction, is used in Eq. (24) in contrast to the coefficient 1/6 in Beeman's algorithm. The latter leads to the appearance of a rounding error because of the replacement of an infinite decimal fraction by a finite fraction.

In a similar way, one can determine the order of precision for the remaining expressions.

For relations (6) and (9), we have

$$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{8}(a_{t+\Delta t} + 3a_t)\Delta t^2 + \frac{1}{24}b_t\Delta t^3 + O(\Delta t^4), \quad (26)$$

$$v_{t+\Delta t}\Delta t = x_{t+\Delta t} - x_t + \frac{1}{8}(3a_{t+\Delta t} + a_t)\Delta t^2 - \frac{1}{24}b_t\Delta t^3 + O(\Delta t^4). \quad (27)$$

After the substitution of Eq. (26) into expression (27), which is of the second order of precision, we obtain an expression corresponding to Eq. (14) and ensuring the third order of precision; that is,

$$v_{t+\Delta t} = v_t + \frac{1}{8}(3a_{t+\Delta t} + 6a_t - a_{t-\Delta t})\Delta t + O(\Delta t^3).$$

The results obtained by estimating the order of precision for all forms of our algorithm are given in Table 1.

Table 1 shows that Beeman's algorithm determines the coordinate to the fourth order of precision and the velocity to the third order of precision; thus, this algorithm has a total error of the third order. The method that we proposed has the second order of precision in the predictor–corrector form and the third order of precision in the explicit form and in the modified predictor–corrector form. In addition, our algorithm is more stable, as was shown above, with respect to rounding errors owing to the use of the coefficient 1/8 instead of 1/6. We also note that a high degree of energy conservation and a low sensitivity to rounding errors owing to the fact that the difference of markedly different numbers ($4a_t - a_{t-\Delta t}$) is calculated in Beeman's algorithm, are advantages of this algorithm. In our algorithm, one calculates the difference ($5a_t - a_{t-\Delta t}$) of numbers that differ more strongly (than in Beeman's algorithm).

3. NUMERICAL COMPARISON WITH OTHER ALGORITHMS

In order to evaluate the properties of our algorithm in practical applications and to perform a comparison with other methods, we will simulate numerically the motion of a harmonic oscillator. In one-dimensional space, its equation of motion has the form

$$a = -\omega^2 x. \quad (28)$$

The analytic solution of this equation is

$$x = C_1\cos(\omega t) + C_2\sin(\omega t).$$

The constants C_1 and C_2 depend on initial conditions

We performed calculations on the basis of the following methods: (i) the velocity Verlet algorithm, (ii) Beeman's algorithm in the explicit form, (iii) fourth-order Runge–Kutta method [the second-order equation in (28) was reduced to a set of first-order equations], and (iv) our algorithm in the explicit form.

Table 2. Implementation time, ms

Algorithm	Number of points				
	10^3	10^4	10^5	10^6	10^7
Verlet	0.67	6.82	57	552	5585
Beeman's	0.63	6.07	59	610	6102
RK4	34.4	351	3544	35203	351758
Our	0.63	5.95	58	601	6001

The velocity Verlet algorithm is a numerical method that is intended for solving equations of motion. Beeman's algorithm is its modification that preserves energy better.

The velocity Verlet algorithm can written as

$$x_{t+\Delta t} = x_t + v_t\Delta t + \frac{1}{2}a_t\Delta t^2,$$

$$v_{t+\Delta t} = v_t + \frac{1}{2}(a_{t+\Delta t} + a_t)\Delta t.$$

The Verlet algorithm, Beeman's algorithm, and the algorithm proposed in the present study are special algorithms intended for integrating second-order differential equations. A higher rate of respective calculations is their basic advantage over algorithms for first-order differential equations. The calculation of the force (or acceleration) at each time step is performed four times in the Runge–Kutta method but only once in the algorithms quoted above. It follows that, in the case where the calculation of the force consumes a dominant portion of computer time, the above special algorithms are four times faster. But if the determination of the force requires a smaller volume of calculations than the algorithm itself, then the special algorithms prove to be faster by a factor of several tens. For the example being considered, in particular, the special algorithms are more than 60 times as fast as the Runge–Kutta method (see Table 2).

In the methods being considered, the error in evaluating the coordinate manifests itself differently. In the Runge–Kutta method, there occurs an accumulation of the error, and the solution becomes degenerate at large values of the step and for a wide interval of the calculations; within the above special methods, the error in calculating the coordinate leads to a change in the oscillation frequency, but the amplitude and energy remain virtually unchanged, which makes it possible to perform calculations with a large step and over arbitrarily wide intervals.

Energy conservation is an important factor in simulating the motion of a particle. Within the special methods in question, the error in determining energy does not go beyond specific boundaries over the whole interval of the calculations. The application of the Runge–Kutta method entails an accumulation of the

Table 3. Maximum error in determining energy at $t = 1000$, %

Method	dt, s					
	0.001	0.01	0.02	0.08	0.1	0.3
Verlet	2.5×10^{-5}	2.5×10^{-3}	0.01	0.15	0.25	2.3
Beeman's	8×10^{-6}	8.5×10^{-4}	3.3×10^{-3}	0.05	0.08	0.65
RK4	6×10^{-12}	1.5×10^{-7}	4.5×10^{-6}	4.5×10^{-3}	0.015	3.3
Our	6×10^{-9}	6.5×10^{-6}	4.8×10^{-5}	3.8×10^{-3}	5×10^{-3}	0.3

error, with the result that that this method preserves energy poorer over wide intervals of the calculations.

Tables 3 and 4 give the errors in determining the energy versus, respectively, the step and interval of integration. These data show that the proposed algorithm preserves energy substantially better than the other special algorithms. It is noteworthy that the Runge–Kutta method becomes inferior to the proposed algorithm as the step or interval of the calculations grows substantially.

4. NUMERICAL SIMULATION OF NONSTATIONARY DYNAMICS IN THE CASE OF THE PIERCE DIODE IN THE MODE OF VIRTUAL-CATHODE FORMATION

Let us now consider the Pierce diode [7, 9], which is a classic example of a UHF electronics system featuring intricate nonstationary chaotic dynamics. The Pierce diode is formed by two infinite plane parallel grids traversed by an indefinitely wide flow of monoenergetic electrons. The charge density ρ_0 and the flow velocity v_0 at the inlet of the diode gap are maintained at a constant level. The space between the grids is filled with a neutralizing background of immobile ions, whose density is denoted by ρ_i. The neutralizing-charge density is equal to the unperturbed charge density in the electron flow ($\rho_0 = -\rho_i$).

The development of a nonradiative instability, called the Pierce instability [9, 10], is possible in this system under some specific conditions. This instability is associated with the existence of an external feedback through the circuit connecting the grids, since they are grounded [10]. The condition $\alpha = \omega_p L/v_0 > \pi$, which leads to the formation of a virtual cathode—a region lying in the interaction space, reflecting some electrons toward the injection plane, and executing oscillations in space and time [7]—is the condition of development of the Pierce instability. In [11–13], it was shown that, in such a system, various regimes of oscillations, including chaotic dynamics, are possible in the mode of virtual-cathode formation.

Table 4. Maximum error in determining energy at $dt = 0.04$, %

Method	t, s			
	10^3	10^3	10^4	10^5
Verlet	4×10^{-2}	4×10^{-2}	4×10^{-2}	4×10^{-2}
Beeman's	1.4×10^{-2}	1.4×10^{-2}	1.4×10^{-2}	1.4×10^{-2}
RK4	1.4×10^{-5}	1.4×10^{-4}	1.4×10^{-3}	1.4×10^{-2}
Our	4.5×10^{-4}	4.5×10^{-4}	4.5×10^{-4}	4.5×10^{-4}

In order to analyze the nonstationary nonlinear dynamics of a virtual cathode, it is necessary to use a numerical simulation by the PIC method [1]. In flat geometry, the electron flow is represented in the form of large particles (charged sheets) injected at regular time intervals with a constant velocity into the interaction space. The problem was solved in terms of the dimensionless variables $\rho = \rho'/|\rho_0|$, $v = v'/v_0$, $t = t'v_0/L$, $x = x'/L$, and $E = E'/\omega_p v_0$.

For each sheet, one solves the dimensionless nonrelativistic equations of motion

$$\frac{d^2x_i}{dt^2} = -E(x_i), \tag{29}$$

where x_i is the coordinate of the ith charged sheet and $E(x_i)$ is the strength of the spatial-charge field at the point whose coordinate is x_i.

In order to calculate the strength and potential of the spatial-charge field, as well as the charge density, we introduce an equidistant spatial mesh with a step Δx_i. In the electrostatic approximation, the potential of the spatial-charge field is determined from Poisson's equation, which, in the one-dimensional approximation, is given by

$$\frac{\partial^2\varphi}{dx^2} = -\alpha^2(\rho(x) - 1) \tag{30}$$

and which is solved for the following boundary conditions: $\varphi(0) = \varphi(L) = 0$. The strength of the spatial-charge field is determined by numerically differentiating the potential values obtained in this way.

In order to calculate the spatial-charge density, we applied the procedure of linearly weighing particles (sheets) on the spatial mesh. Within this method, the spatial-charge density at the ith node of the spatial mesh—that is, at the point whose coordinate is x_j—can be written as

$$\rho(x_j) = \frac{1}{n_0}\sum_{i=1}^{N}\Theta(x_i - x_j), \quad (31)$$

where x_i is the coordinate of the ith particle, N is the total number of large particles, n_0 is a computational-scheme parameter that is equal to the number of particles per cell in an unperturbed state, and

$$\Theta(\eta) = \begin{cases} 1 - |\eta/\Delta x|, & |\eta| < \Delta x \\ 0, & |\eta| > \Delta x \end{cases}$$

is a shape function that determines the procedure of weighing a large particle on the spatial mesh with a step Δx.

The problem was solved at the following values of the numerical-scheme parameters: $n_0 = 8$ and $\Delta x = 8.3 \times 10^{-4}$. This corresponds to the number $N_x = 1200$ of spatial-mesh nodes and to the number $N = 19\,200$ of particles in an unperturbed state.

We have considered the following schemes for solving the equation of motion in (29): the leapfrog algorithm [1], which is the simplest and most frequently used in dealing with such problems; the Verlet algorithm; and the new algorithm proposed in the present study. In our test calculations, we have considered two values of the Pierce parameter, $\alpha_1 = 1.3\pi$ and $\alpha_2 = 1.2\pi$. These values correspond to the cases where the regime of oscillations in the spatially distributed beam–plasma system is close to a simple periodic regime for the former and is intricate chaotic for the latter [11–14]. Table 5 gives the results of an analysis of oscillation regimes for the dimensionless-time step fixed at $\Delta t = 1.04 \times 10^{-4}$. In Table 5, we present the results obtained by estimating the relative inaccuracy $\delta = \Delta\varepsilon/W_E$ in fulfillment of the energy-conservation law—following [1], we define it as the ratio of the macroscopic change in the total energy, $\Delta\varepsilon$, to the field energy, W_E.

One can clearly see that, in dealing with the Pierce diode problem, the method that we proposed in the present study provides an accuracy that is substantially higher than the accuracy of the traditional leapfrog algorithm and which is approximately twice as high as the accuracy of the Verlet method. At the same time, the leapfrog method is less time-consuming, but it is much less accurate. Yet another positive facet of the method that we proposed is worthy of special note. For periodic and chaotic-oscillation regimes, this method leads to approximately identical degrees of precision of fulfillment of the energy-conservation law; moreover, solutions are virtually independent of the regime of oscillations in the system. This distinguishes our method from the other methods under comparison within which the accuracy of fulfillment of the energy-conservation law becomes lower upon going over to the chaotic regime of space–time oscillations. The reason is that, within the proposed method, the coordinate and velocity and, hence, the energy of particles executing several oscillations in the interaction space (metastable particles appearing in the chaotic regime in the virtual-cathode region) are determined with a rather small error over extended intervals of calculations.

Table 5. Relative precision of fulfillment of the energy-conservation law and code-implementation time T at the dimensionless time step of $\Delta t = 1.04 \times 10^{-4}$ (the calculation time corresponds to 200 characteristic oscillation periods)

Method	Pierce parameter, α			
	1.3π (nearly periodic regime)		1.2π (chaotic regime)	
	δ, %	T, s	δ, %	T, s
Leapfrog	1.16	3.62	2.61	3.59
Verlet	0.21	4.25	0.39	4.19
Our	0.12	5.32	0.13	5.28

CONCLUSIONS

The results presented in this article indicate that it is reasonable to employ the proposed algorithm as a substitute to Beeman's algorithm and the Verlet algorithm and, in the case where one has to perform a large volume of calculations within a limited time, as an alternative to other popular methods. It is also advisable to use the new algorithm in solving problems involving an analysis of an intricate chaotic behavior of particle systems, since, in that case, the accuracy of the solution is virtually independent of the regime of oscillations in the system, in contrast to what we have within other methods.

ACKNOWLEDGMENTS

This work was supported by the Ministry of Education and Science of Russian Federation (contract no. 14.V37.21.0903), the Russian Foundation for Basic Research (project nos. 12-02-90022 and 12-02-33071), and the Council in Charge of Distributing Grants from the President of the Russian Federation for support of young doctors of sciences (project no. MD-345.2013.2).

REFERENCES

1. C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation* (McGraw-Hill, New York, 1985).
2. T. M. Antonsen, A. A. Mondelli, B. Levush, J. P. Verboncoeur, and C. K. Birdsall, Proc. IEEE **87**, 804 (1999).
3. A. S. Roshal', *Modeling of Charged Beams* (Atomizdat, Moscow, 1979) [in Russian].
4. M. P. Batura, A. A. Kuraev, and T. D. Popkova, *Irregular Electrodynamic Structures: Theory and Calculation Methods* (Bestprint, Minsk, 2011) [in Russian].

5. A. A. Kuraev and D. I. Trubetskov, *Methods of Nonlinear Dynamics and Chaos Theory in Problems of Microwave Electronics,* Vol. 1: *Stationary Processes* (Fizmatlit, Moscow, 2009) [in Russian].
6. A. A. Koronovskii, D. I. Trubetskov, and A. E. Khramov, *Methods of Nonlinear Dynamics and Chaos in Problems of Microwave Electronics,* Vol. 2: *Nonstationary and Chaotic Processes* (Fizmatlit, Moscow, 2009) [in Russian].
7. D. I. Trubetskov and A. E. Khramov, *Lectures on Microwave Electronics for Physicists* (Fizmatlit, Moscow, 2003) [in Russian].
8. H. Gould and J. Tobochnik, *An Introduction to Computer Simulation Methods: Applications to Physical Systems,* Vol. 1 (Addison-Wesley, Reading, 1988).
9. J. R. Pierce, J. Appl. Phys. **15**, 721 (1944).
10. M. V. Kuzelev, A. A. Rukhadze, and P. S. Strelkov, *Plasma Relativistic Microwave Electronics* (MGTU im. N. E. Baumana, Moscow, 2002).
11. V. G. Anfinogentov and A. E. Khramov, Izv. Ross. Akad. Nauk, Ser. Fiz. **61**, 2391 (1997).
12. H. Matsumoto, H. Yokoyama, and D. Summers, Phys. Plasmas **3**, 177 (1996).
13. V. G. Anfinogentov, Izv. Vyssh. Uchebn. Zaved., Prikl. Nelineinaya Din. **2**, 69 (1994).
14. A. E. Hramov, A. A. Koronovskii, and I. S. Rempen, Chaos **16**, 013123 (2006).

Translated by A. Isaakyan