Computer Physics Communications 247 (2020) 106954

Contents lists available at ScienceDirect

Computer Physics Communications

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

Multiple Boris integrators for particle-in-cell simulation☆

Seiji Zenitani [a,b,*], Tsunehiko N. Kato [c]

[a] Research Center for Urban Safety and Security, Kobe University, 1-1 Rokkodai-cho, Nada-ku, Kobe 657-8501, Japan
[b] Research Institute for Sustainable Humanosphere, Kyoto University, Gokasho, Uji, Kyoto 611-0011, Japan
[c] Center for Computational Astrophysics, National Astronomical Observatory of Japan, 2-21-1 Osawa, Mitaka, Tokyo 181-8588, Japan

ARTICLE INFO

Article history:
Received 31 January 2019
Received in revised form 21 August 2019
Accepted 9 September 2019
Available online 26 September 2019

Keywords:
Boris integrator
Particle-in-cell simulation
Lorentz force equation

ABSTRACT

We construct Boris-type schemes for integrating the motion of charged particles in particle-in-cell (PIC) simulation. The new solvers virtually combine the 2-step Boris procedure arbitrary n times in the Lorentz-force part, and therefore we call them the multiple Boris solvers. Using Chebyshev polynomials, a one-step form of the new solvers is provided. The new solvers give n^2 times smaller errors, allow larger timesteps, and have a long-term stability. We present numerical tests of the new solvers, in comparison with other particle integrators.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Particle-in-cell (PIC) simulation [1,2] has been extensively used to study complex kinetic phenomena in plasma systems. The Boris solver [3] (or the Boris pusher) is a de fact standard method to advance charged particles in PIC simulation. Although its algorithm is simple, it is robust and provides moderately good accuracy. For these reasons, it has been used about a half century.

Since mid 2010s, there is a renewed interest to develop new particle integrators. In recent PIC simulations, one often solves the system evolution much longer than before. In such a case, we want to solve the particle motion as accurately as possible, because small errors could be accumulated over many timesteps. The Boris solver is proven to be stable [4,5], however, it does produce a second-order error. For this reason, there is a growing demand for higher-accuracy extensions to the Boris solver.

Umeda [6,7] has proposed multistep Boris solvers. Repeating a Boris-type procedure multiple steps in the Lorentz-force part, the author managed to set an effective timestep 2^{N-2} times smaller. Since the Boris solver is a second-order scheme, N-step Boris solvers provide 4^{N-2} times higher accuracy for gyration. Zenitani and Umeda [8] have proposed to use a rotation formula in the Lorentz-force part. The solver virtually provides no error in the phase angle in gyration. Because of operator-splitting, their scheme offers second-order accuracy.

In addition to higher numerical accuracy, it is also important to improve computation cost. A fast solver helps us to reduce the total computation time of PIC simulation. The cost depends on many other factors, but the timestep Δt is certainly one of them. If the solver allows us to use a larger Δt, we can further reduce the computation time. Thus it is important to evaluate the constraint on Δt for particle solvers.

In this article, we propose new Boris-type integrators to solve the charged-particle motion. The new solvers use a Boris-type procedure multiple times in the Lorentz-force part. We call them "multiple Boris solvers". They provide higher accuracy and allow larger timesteps than the original Boris solver. In Section 2, we briefly review the classical 2-step Boris solver and discuss its graphical meanings. In Section 3, we introduce our multiple Boris solvers. Using Chebyshev polynomials, we derive coefficients for n-tuple Boris solvers, where n is an arbitrary integer. Then we discuss errors, limitations to the timesteps, and the stability of the phase-space volume. In Section 4, we present benchmark results. The new solvers and other particle integrators are briefly compared. Section 5 contains discussion and summary.

2. Boris solver

We describe a popular form of the Boris algorithm [1,2]. We refer to it as the classical Boris solver. The algorithm solves the particle motion in a time-staggered manner,

$$\frac{\boldsymbol{x}^{t+\Delta t}-\boldsymbol{x}^{t}}{\Delta t}=\frac{\boldsymbol{u}^{t+\frac{\Delta t}{2}}}{\gamma^{t+\frac{\Delta t}{2}}}, \tag{1}$$

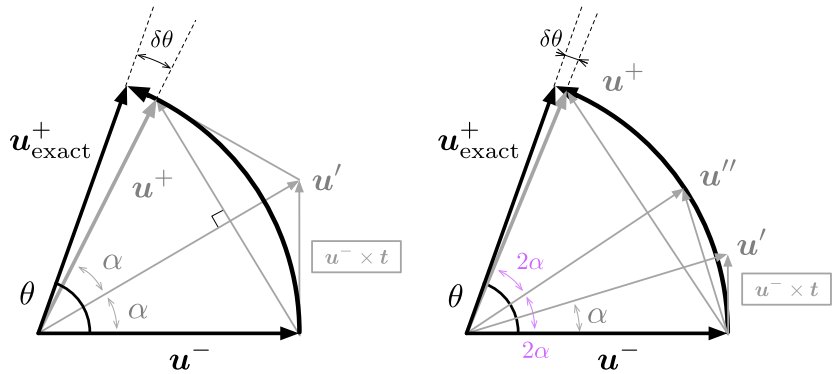
$$m\,\frac{\boldsymbol{u}^{t+\frac{\Delta t}{2}}-\boldsymbol{u}^{t-\frac{\Delta t}{2}}}{\Delta t}=q\left(\boldsymbol{E}^{t}+\frac{\boldsymbol{u}^{t+\frac{\Delta t}{2}}+\boldsymbol{u}^{t-\frac{\Delta t}{2}}}{2\gamma^{t}}\times\boldsymbol{B}^{t}\right). \tag{2}$$

The superscripts $(t, t+\Delta t, \ldots)$ indicate time, $\boldsymbol{u}=\gamma\boldsymbol{v}$ is the spatial part of a 4-vector, and $\gamma=[1-(v/c)^2]^{-1/2}=[1+(u/c)^2]^{1/2}$ is

☆ The review of this paper was arranged by David W. Walker.

* Corresponding author at: Research Center for Urban Safety and Security, Kobe University, 1-1 Rokkodai-cho, Nada-ku, Kobe 657-8501, Japan.
E-mail address: zenitani@port.kobe-u.ac.jp (S. Zenitani).

https://doi.org/10.1016/j.cpc.2019.106954
0010-4655/© 2019 Elsevier B.V. All rights reserved.



(a) Boris solver (b) Double Boris solver

Fig. 1. (a) classical Boris solver and (b) double Boris solver.

the Lorentz factor. Other symbols have their standard meanings. Hereafter we denote $\boldsymbol{B}^t$ as $\boldsymbol{B}$ for brevity.

The acceleration part (Eq. (2)) is split into the Coulomb-force part for the first half:

$$\boldsymbol{u}^- = \boldsymbol{u}^{t-\frac{\Delta t}{2}} + \boldsymbol{\varepsilon}\Delta t, \tag{3}$$

the middle Lorentz-force part:

$$\boldsymbol{t} \equiv \frac{\theta}{2}\,\hat{\boldsymbol{b}} = \frac{\omega_c \Delta t}{2}\,\hat{\boldsymbol{b}} = \frac{q\Delta t}{2m\gamma^-}\boldsymbol{B}, \tag{4a}$$

$$\boldsymbol{u}' = \boldsymbol{u}^- + \boldsymbol{u}^- \times \boldsymbol{t}, \tag{4b}$$

$$\boldsymbol{u}^+ = \boldsymbol{u}^- + \frac{2}{1+t^2}\Big(\boldsymbol{u}' \times \boldsymbol{t}\Big), \tag{4c}$$

and the Coulomb-force part for the second half:

$$\boldsymbol{u}^{t+\frac{\Delta t}{2}} = \boldsymbol{u}^+ + \boldsymbol{\varepsilon}\Delta t. \tag{5}$$

In these equations, $\boldsymbol{\varepsilon} = (q/2m)\boldsymbol{E}^t$, $\omega_c = (qB/m\gamma^-)$ is the gyro frequency, $\theta = 2t = \omega_c\Delta t$ is the gyration angle during Δt, $\hat{\boldsymbol{b}} = \boldsymbol{B}/|B|$ is a unit vector in the $\boldsymbol{B}$ direction, $\boldsymbol{u}^-$ and $\boldsymbol{u}^+$ are two intermediate states.

We focus on the Lorentz-force part in the middle (Eqs. (4a)–(4c)). This 2-step procedure is illustrated in Fig. 1(a). The coefficient $2/(1+t^2)$ is provided such that the particle energy remains constant in the Lorentz-force part, $\gamma^- = \gamma^+ = \gamma^t$. A base angle α is given by,

$$\alpha \equiv \arctan t \tag{6}$$

We will use the following relations for α in subsequent discussions.

$$\sin\alpha = \frac{t}{\sqrt{1+t^2}}, \quad \cos\alpha = \frac{1}{\sqrt{1+t^2}}, \quad \tan\alpha = t \tag{7}$$

$$\sin 2\alpha = \frac{2t}{1+t^2}, \quad \cos 2\alpha = \frac{1-t^2}{1+t^2} \tag{8}$$

From Eqs. (4b), (4c), and (8), we obtain

$$\boldsymbol{u}^+ = \frac{1}{1+t^2}\Big((1-t^2)\boldsymbol{u}^- + 2(\boldsymbol{u}^- \times \boldsymbol{t}) + 2(\boldsymbol{u}^- \cdot \boldsymbol{t})\boldsymbol{t}\Big) \tag{9}$$

$$= \boldsymbol{u}^- \cos 2\alpha + (\boldsymbol{u}^- \times \hat{\boldsymbol{b}})\sin 2\alpha + (1-\cos 2\alpha)\boldsymbol{u}^-_{\parallel} \tag{10}$$

$$= (\boldsymbol{u}^- - \boldsymbol{u}^-_{\parallel})\cos 2\alpha + (\boldsymbol{u}^- \times \hat{\boldsymbol{b}})\sin 2\alpha + \boldsymbol{u}^-_{\parallel}, \tag{11}$$

where the subscript $\parallel$ indicates the component along $\boldsymbol{B}$. These forms are known as Euler–Rodrigues rotation formula. The equations correspond to a rotation about the magnetic field by an approximate angle of 2α, instead of the true angle θ. As indicated in Fig. 1(a), the half acceleration in the tangent direction ($\boldsymbol{u}^- \times \boldsymbol{t}$; Eq. (4b)) corresponds to the half approximate angle α by the Boris solver. The true rotation angle θ contains a second-order error of $\delta\theta/\theta \approx -\frac{1}{12}\theta^2$.

$$\theta \approx 2\alpha = 2\arctan\frac{\theta}{2} = \theta\left(1 - \frac{1}{3}\left(\frac{\theta}{2}\right)^2 + \frac{1}{5}\left(\frac{\theta}{2}\right)^4 - \cdots\right). \tag{12}$$

3. Multiple Boris solver

Umeda [6] has proposed a novel way to reduce the error ($\delta\theta/\theta$). He employed a half timestep $\Delta t \to \frac{1}{2}\Delta t$ in the Lorentz-force part (Eqs. (4a)–(4c)) and then repeated the Boris procedure twice. Importantly, the timestep for the position (Eq. (1)) and the Coulomb-force parts (Eqs. (3) and (5)) are unchanged. We call this method the double Boris solver, and it is illustrated in Fig. 1(b). Instead of Eq. (4a), we use

$$\boldsymbol{t} \equiv \frac{\theta}{4}\,\hat{\boldsymbol{b}} = \frac{q\Delta t}{4m\gamma^-}\boldsymbol{B}, \quad \alpha \equiv \arctan t \tag{13}$$

The base angle α is redefined, as indicated in Fig. 1(b). Then, we repeat the 2-step Boris procedure (Eqs. (4b) and (4c)) twice. We obtain a half state $\boldsymbol{u}''$ by the first Boris procedure, which is given by Eq. (11) with Eq. (13). Applying $\cdot\hat{\boldsymbol{b}}$ to Eq. (11), one can confirm that the parallel component is preserved, $\boldsymbol{u}''_{\parallel} = (\boldsymbol{u}'' \cdot \hat{\boldsymbol{b}})\hat{\boldsymbol{b}} = \boldsymbol{u}^-_{\parallel}$. Substituting $\boldsymbol{u}''$ to Eq. (11) again, we obtain the next state $\boldsymbol{u}^+$,

$$\boldsymbol{u}^+ = (\boldsymbol{u}'' - \boldsymbol{u}^-_{\parallel})\cos 2\alpha + (\boldsymbol{u}'' \times \hat{\boldsymbol{b}})\sin 2\alpha + \boldsymbol{u}''_{\parallel} \tag{14}$$

$$= (\boldsymbol{u}^- - \boldsymbol{u}^-_{\parallel})\cos 4\alpha + (\boldsymbol{u}^- \times \hat{\boldsymbol{b}})\sin 4\alpha + \boldsymbol{u}^-_{\parallel} \tag{15}$$

The second Boris operation advances the phase angle by another 2α, and the entire procedure gives gyration by 4α. In Ref. [6], the author proposed a 3-step procedure to repeat the Boris operation twice. Now we obtain a smaller second-order error in gyration, $\delta\theta/\theta \approx -\frac{1}{48}\theta^2$.

$$\theta \approx 4\alpha = 4\arctan\frac{\theta}{4} = \theta\left(1 - \frac{1}{3}\left(\frac{\theta}{4}\right)^2 + \frac{1}{5}\left(\frac{\theta}{4}\right)^4 - \cdots\right). \tag{16}$$

Let us extend this idea. In order to better approximate the gyro motion, we propose to repeat the Boris procedure n times, where n is an arbitrary positive integer. We call the resulting schemes "multiple Boris solvers". Here below, we describe its procedure in the Lorentz-force part.

We define the tangent vector $\boldsymbol{t}$ and the base angle α as follows.

$$\boldsymbol{t} \equiv \frac{\theta}{2n}\hat{\boldsymbol{b}} = \frac{q\Delta t}{2\,nm\gamma^-}\boldsymbol{B}, \quad \alpha \equiv \arctan t. \tag{17}$$

Note that $\boldsymbol{t}$ is divided by n. From the discussion on the double Boris solver, repeating the Boris operation n times, one can obtain

a rotation angle of $2n\alpha$. The next state $\boldsymbol{u}_n^+$ should be given by

$$\boldsymbol{u}_n^+ = (\boldsymbol{u}^- - \boldsymbol{u}_\parallel^-)\cos(2n\alpha) + (\boldsymbol{u}^- \times \hat{\boldsymbol{b}})\sin(2n\alpha) + \boldsymbol{u}_\parallel^- \tag{18}$$

The subscript n indicates quantities for the n-tuple solver. For brevity, we hereafter omit the minus sign from $\boldsymbol{u}^-$. Since we desire a simulation-friendly expression with $\boldsymbol{t}$, we modify Eq. (18) to

$$\boldsymbol{u}_n^+ = \cos(2n\alpha)\boldsymbol{u} + \frac{\sin(2n\alpha)}{t}(\boldsymbol{u}\times\boldsymbol{t}) + \frac{2\sin^2(n\alpha)}{t^2}(\boldsymbol{u}\cdot\boldsymbol{t})\boldsymbol{t} \tag{19}$$

$$= c_{n1}\boldsymbol{u} + c_{n2}\,(\boldsymbol{u}\times\boldsymbol{t}) + c_{n3}(\boldsymbol{u}\cdot\boldsymbol{t})\boldsymbol{t} \tag{20}$$

where c_{n1}, c_{n2}, and c_{n3} are coefficients.

Next, we construct generic expressions for these coefficients. We use Chebyshev polynomials of the first kind $T_n(x)$: $T_0(x) = 1,\ T_1(x) = x,\ T_2(x) = 2x^2 - 1, \ldots$ and the second kind $U_n(x)$: $U_0(x) = 1,\ U_1(x) = 2x,\ U_2(x) = 4x^2 - 1, \ldots$. They satisfy useful relations:

$$\cos nx = T_n(\cos x), \quad \sin nx = \sin x\, U_{n-1}(\cos x) \tag{21}$$

We also employ

$$p \equiv \cos 2\alpha = \frac{1-t^2}{1+t^2}, \quad 1+p = \frac{\sin 2\alpha}{t} = \frac{2}{1+t^2}. \tag{22}$$

With help from Eqs. (7), (8), (21), and (22), we express the coefficients in Eq. (20) in the following way,

$$c_{n1} = T_n(\cos 2\alpha) = T_n(p) \tag{23}$$

$$c_{n2} = \frac{1}{t}\sin 2\alpha\ U_{n-1}(\cos 2\alpha) = (1+p)U_{n-1}(p) \tag{24}$$

$$c_{n3} = \begin{cases} 1+p & (\text{for } n=1) \\ (1+p)\left(U_k(p) + U_{k-1}(p)\right)^2 & (\text{for } n = 2k+1) \\ 2\left((1+p)U_{k-1}(p)\right)^2 & (\text{for } n = 2k) \end{cases} \tag{25}$$

Here, k is a positive integer. We derive Eq. (25) for $n = 2k+1$ ($n = 3, 5, \ldots$) in Appendix. Practically, we compute p first, and then calculate Eqs. (23)–(25) by using p. Some higher-order polynomials can be calculated via nesting relations [9]:

$$T_{mn}(x) = T_m(T_n(x)),\ U_{mn-1}(x) = U_{m-1}(T_n(x))U_{n-1}(x) \tag{26}$$

We can also express the coefficients with t. From the polynomials,

$$c_{n1} = T_n\left(\frac{1-t^2}{1+t^2}\right) \tag{27}$$

$$c_{n2} = \frac{2}{1+t^2}U_{n-1}\left(\frac{1-t^2}{1+t^2}\right) \tag{28}$$

$$c_{n3} = \begin{cases} \frac{2}{1+t^2} & (\text{for } n=1) \\ \frac{2}{1+t^2}\left(U_k\left(\frac{1-t^2}{1+t^2}\right) + U_{k-1}\left(\frac{1-t^2}{1+t^2}\right)\right)^2 & (\text{for } n = 2k+1) \\ \frac{8}{(1+t^2)^2}\left(U_{k-1}\left(\frac{1-t^2}{1+t^2}\right)\right)^2 & (\text{for } n = 2k) \end{cases} \tag{29}$$

we obtain explicit forms, for example,

$$c_{11} = \frac{1-t^2}{1+t^2}, \quad c_{12} = \frac{2}{1+t^2}, \quad c_{13} = \frac{2}{1+t^2}, \tag{30}$$

$$c_{21} = \frac{1-6t^2+t^4}{(1+t^2)^2}, \quad c_{22} = \frac{4(1-t^2)}{(1+t^2)^2}, \quad c_{23} = \frac{8}{(1+t^2)^2}, \tag{31}$$

$$c_{31} = \frac{1-15t^2+15t^4-t^6}{(1+t^2)^3},$$

$$c_{32} = \frac{2(t^2-3)(3t^2-1)}{(1+t^2)^3}, \quad c_{33} = \frac{2(t^2-3)^2}{(1+t^2)^3}, \tag{32}$$

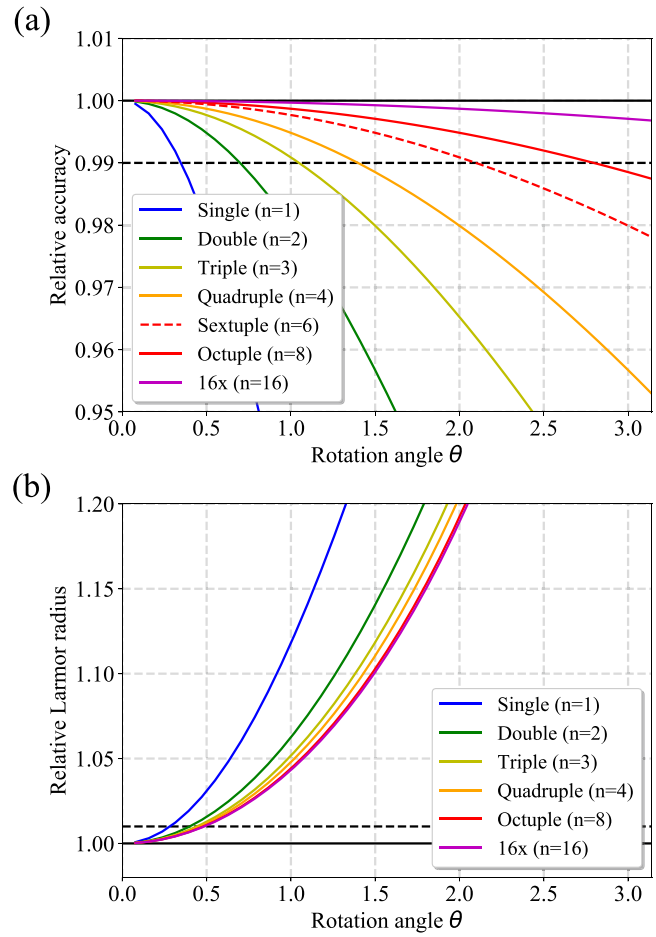


Fig. 2. (a) Relative accuracy in phase angle and (b) relative length of Larmor radius for the multiple Boris solvers.

$$c_{41} = \frac{\left((1-t^2)^2 - 24t^2\right)(1-t^2)^2 + 16t^4}{(1+t^2)^4}$$

$$= \frac{1-28t^2+70t^4-28t^6+t^8}{(1+t^2)^4},$$

$$c_{42} = \frac{8(1-t^2)(1-6t^2+t^4)}{(1+t^2)^4}, \quad c_{43} = \frac{32(1-t^2)^2}{(1+t^2)^4} \tag{33}$$

It is interesting to see that the c_{n1}, c_{n2}, and c_{n3} share the same denominator, $(1+t^2)^n$. One can confirm this by inspecting Eqs. (27)–(29).

As a result, the multiple Boris solver advances the velocity by using Eqs. (3), (17), (22)–(25), (20), and (5). One can also use the coefficients in Eqs. (30)–(33) instead of Eqs. (22)–(25).

The Lorentz-force part of the multiple Boris solvers provides a second-order error in phase angle. From Eq. (17), we obtain

$$\theta \approx 2n\alpha = 2n\arctan\frac{\theta}{2n} = \theta\left(1 - \frac{\theta^2}{12n^2} + \frac{\theta^4}{80n^4} - \cdots\right). \tag{34}$$

Thus, the multiple Boris solvers provide n^2 times smaller errors than the single Boris solver. Eq. (34) approaches the exact angle for $n \to \infty$.

In Fig. 2(a), we plot relative accuracy in phase angle, $\theta^{-1} \cdot 2n\arctan(\theta/2n)$, for several n. In the classical Boris solver, which is equivalent to the $n = 1$ case, we typically set $\theta = \omega_c \Delta t < 0.2$–$0.3$ to keep an error within 1%. Considering $\theta^2/(12n^2) \lesssim 0.01$, we obtain

$$\theta \lesssim 0.35n. \tag{35}$$

The multiple Boris solvers allow n-times larger timestep than the single Boris solver.

Next, we evaluate an effective Larmor radius. We consider a simple gyration about $\boldsymbol{B}$ in the case of $\boldsymbol{E} = 0$. From Eq. (1), we obtain

$$\left|\boldsymbol{x}^{t+\Delta t} - \boldsymbol{x}^t\right| = \frac{\left|\boldsymbol{u}^{t+\frac{\Delta t}{2}}\right|}{\gamma^{t+\frac{\Delta t}{2}}}\Delta t = |v_R|\Delta t, \tag{36}$$

where v_R is the velocity. The two points $\boldsymbol{x}^{t+\Delta t}$ and $\boldsymbol{x}^t$ are located on the same circle with a Larmor radius of r_L'. The two points should be two endpoints of an arc with a central angle $2n\alpha$. Then we expect

$$|\boldsymbol{x}^{t+\Delta t} - \boldsymbol{x}^t| = 2r_L' \sin\left(\frac{2n\alpha}{2}\right). \tag{37}$$

From Eqs. (36) and (37), we obtain

$$r_L' = \frac{|v_R|\Delta t}{2\sin(n\alpha)} = r_L \frac{\theta}{2\sin(n\alpha)} \tag{38}$$

where $r_L = |v_R|/\omega_c$ is a physical Larmor radius. For $n = 1$, we obtain a familiar result of $r_L' = r_L[1 + (\theta/2)^2]^{1/2}$ from Eq. (7). In general cases, Eq. (38) can be expanded to

$$r_L' = r_L\left(1 + \left[\frac{1}{24} + \frac{1}{12n^2}\right]\theta^2 + \frac{7n^4 - 20n^2 - 32}{5760n^4}\theta^4 + \mathcal{O}(\theta^6)\right) \tag{39}$$

Fig. 2(b) shows the relative length of Larmor radius, $r_L'/r_L = \theta/[2\sin(n\alpha)]$. In the square brackets in Eq. (39), the first term is insensitive to n, while the second depends on n. Thus, higher n does not allow n-times larger timestep. To keep an error within $\approx$1%, we need to limit the timestep to

$$\theta = \omega_c \Delta t \lesssim 0.5\sqrt{\frac{n^2}{n^2+2}}. \tag{40}$$

For $n \to \infty$, we find $\omega_c \Delta t < 0.5$. If we set the threshold to ϵ%, both Eqs. (35) and (40) can be relaxed by a factor of $\sqrt{\epsilon}$. For example, Eq. (40) gives $\omega_c \Delta t < 1$ for 4%.

The multiple Boris solvers have two nice properties for stable solvers. The first one is the time symmetry. As evident in Eqs. (3), (18), and (5), the multiple Boris solvers is symmetric in time. In such a case, it is unlikely that the solvers numerically damp or amplify a motion in one time direction. In Eq. (20), procedures in the reverse-time direction are obtained by reversing the sign of the second coefficient, i.e., $c_{-n1} = c_{n1}, c_{-n2} = -c_{n2}$, and $c_{-n3} = c_{n3}$. It is obvious that $c_{01} = 1, c_{02} = 0$, and $c_{03} = 0$ for $n = 0$. As a result, we have the coefficients not only for positive integers, but also for any integer n.

The second is the volume preservation [4,5], which is related to the long-term stability. We consider the evolution of a volume in the phase-space from $t - \frac{\Delta t}{2}$ to $t + \frac{\Delta t}{2}$. Since the Lorentz-force part of the multiple Boris solver is equivalent to n-times repetition of the Boris-type element, we express the equivalent numerical procedures into the following steps [5] of $S_1, S_2, \ldots, S_5$, where S_k indicates the kth step. The third step, the Lorentz-force part of the solver, is further divided into n substeps. $\texttt{Boris}(\boldsymbol{u}, \Delta t)$ indicates the Boris-type procedure (Eqs. (4a)–(4c)) on the vector $\boldsymbol{u}$ for a timestep Δt.

$$S_1: \quad \boldsymbol{x}^t = \boldsymbol{x}^{t-\frac{\Delta t}{2}} + \frac{\Delta t}{2}\frac{\boldsymbol{u}^{t-\frac{\Delta t}{2}}}{\gamma^{t-\frac{\Delta t}{2}}}, \quad \boldsymbol{u}^{t-\frac{\Delta t}{2}} = \boldsymbol{u}^{t-\frac{\Delta t}{2}} \tag{41}$$

$$S_2: \quad \boldsymbol{x}^t = \boldsymbol{x}^t, \quad \boldsymbol{u}^- = \boldsymbol{u}^{t-\frac{\Delta t}{2}} + \boldsymbol{\varepsilon}\Delta t \tag{42}$$

$$S_{31}: \quad \boldsymbol{x}^t = \boldsymbol{x}^t, \quad \boldsymbol{u}_1^+ = \texttt{Boris}(\boldsymbol{u}^-, \frac{\Delta t}{n}) \tag{43}$$

$$\vdots$$

$$S_{3m}: \quad \boldsymbol{x}^t = \boldsymbol{x}^t, \quad \boldsymbol{u}_m^+ = \texttt{Boris}(\boldsymbol{u}_{m-1}^+, \frac{\Delta t}{n}) \tag{44}$$

$$\vdots$$

$$S_{3n}: \quad \boldsymbol{x}^t = \boldsymbol{x}^t, \quad \boldsymbol{u}_n^+ = \texttt{Boris}(\boldsymbol{u}_{n-1}^+, \frac{\Delta t}{n}) \tag{45}$$

$$S_4: \quad \boldsymbol{x}^t = \boldsymbol{x}^t, \quad \boldsymbol{u}^{t+\frac{\Delta t}{2}} = \boldsymbol{u}_n^+ + \boldsymbol{\varepsilon}\Delta t \tag{46}$$

$$S_5: \quad \boldsymbol{x}^{t+\frac{\Delta t}{2}} = \boldsymbol{x}^t + \frac{\Delta t}{2}\frac{\boldsymbol{u}^{t+\frac{\Delta t}{2}}}{\gamma^{t+\frac{\Delta t}{2}}}, \quad \boldsymbol{u}^{t+\frac{\Delta t}{2}} = \boldsymbol{u}^{t+\frac{\Delta t}{2}} \tag{47}$$

We check the Jacobian J_k for the kth step, S_k.

$$J_k = \left|\frac{\partial(\boldsymbol{x}^{new}, \boldsymbol{u}^{new})}{\partial(\boldsymbol{x}^{old}, \boldsymbol{u}^{old})}\right| = \det\begin{pmatrix} \frac{\partial \boldsymbol{x}^{new}}{\partial \boldsymbol{x}^{old}} & \frac{\partial \boldsymbol{x}^{new}}{\partial \boldsymbol{u}^{old}} \\ \frac{\partial \boldsymbol{u}^{new}}{\partial \boldsymbol{x}^{old}} & \frac{\partial \boldsymbol{u}^{new}}{\partial \boldsymbol{u}^{old}} \end{pmatrix}. \tag{48}$$

For the first two and the last two steps (S_1, S_2, S_4, and S_5), we obtain

$$J_1 = J_5 = \begin{vmatrix} I & \frac{\partial \boldsymbol{x}^{new}}{\partial \boldsymbol{u}^{old}} \\ 0 & I \end{vmatrix} = 1, \quad J_2 = J_4 = \begin{vmatrix} I & 0 \\ \frac{\partial \boldsymbol{u}^{new}}{\partial \boldsymbol{x}^{old}} & I \end{vmatrix} = 1 \tag{49}$$

In addition, Zhang et al. [5] proved that the Lorentz-force part of the relativistic Boris solver ($\texttt{Boris}(\boldsymbol{u}, \Delta t)$) preserves a phase-space volume,[1] and the proof holds true for an arbitrary timestep Δt. Then, for the substeps in the third step ($S_{31}, \ldots, S_{3m}, \ldots, S_{3n}$), each Boris-type element ($\texttt{Boris}(\boldsymbol{u}, \frac{\Delta t}{n})$) preserves the phase-space volume, i.e., $J_{31} = \cdots = J_{3m} = \cdots = J_{3n} = 1$. As a result, the Jacobian throughout the entire procedure satisfies $J_{\rm all} = J_1 J_2 (J_{31} \cdots J_{3m} \cdots J_{3n}) J_4 J_5 = 1$. This indicates that the entire procedure preserves the phase-space volume. The multiple Boris solvers are volume-preserving.

4. Numerical tests

To evaluate the new solvers, we have conducted four numerical tests. For comparison, the following ten particle integrators are benchmarked:

1. Classical Boris solver (Eqs. (4a)–(4c)),
2. Single Boris solver ($n = 1$; Eqs. (20), (30))
3. Double Boris solver ($n = 2$; Eqs. (20), (31))
4. Quadruple Boris solver ($n = 4$; Eqs. (20), (33)).
5. Octuple Boris solver ($n = 8$; Eqs. (20), (22)–(25)).
6. 16x Boris solver ($n = 16$; Eqs. (20), (22)–(25)).
7. 32x Boris solver ($n = 32$; Eqs. (20), (22)–(25)).
8. Umeda solver ($N = 3$) [6]
9. Umeda solver ($N = 4$) [7]
10. Exact-gyration solver [8]

The first one is the classical Boris solver [1,2]. The next six are the multiple Boris solvers with $n = 1$, $n = 2$, $n = 4$, $n = 8$, $n = 16$, and $n = 32$. The coefficients are given by Eqs. (30)–(33) or Eqs. (22)–(25). We refer to them as the single, double, quadruple, octuple, 16x and 32x Boris solvers, respectively. The next two are the Umeda solvers with $N = 3$ steps [6] and $N = 4$ steps [7]. We follow procedures in Eqs. (8)–(10) in Ref. [7]. As they contain 2^{N-2} Boris elements, they are equivalent to the double and quadruple Boris solvers. The last one is Zenitani and Umeda [8]'s solver, which directly calculates Euler's rotation formula in

[1] The authors claimed that they developed a new volume-preserving algorithm, but the algorithm appears to be the standard relativistic Boris solver.

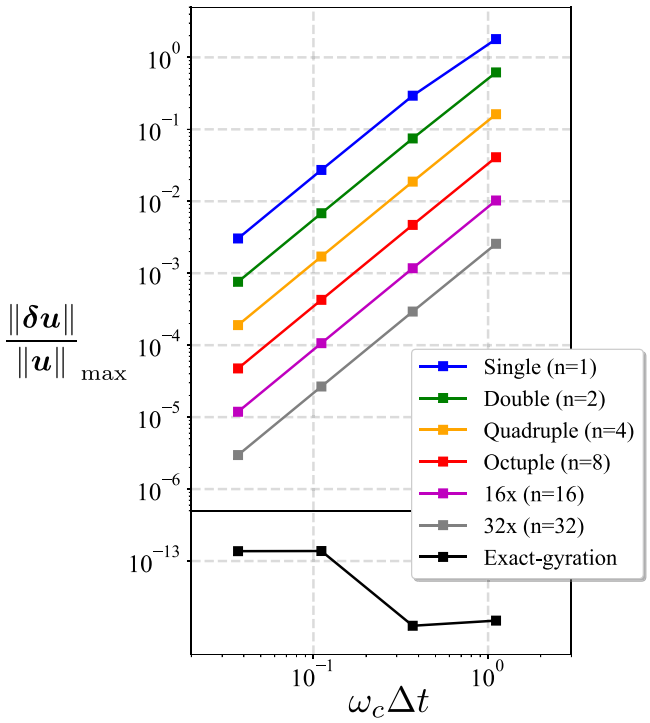


Fig. 3. Normalized errors in $\boldsymbol{u}$ by the test-particle simulations as a function of $\theta = \omega_c \Delta t$.

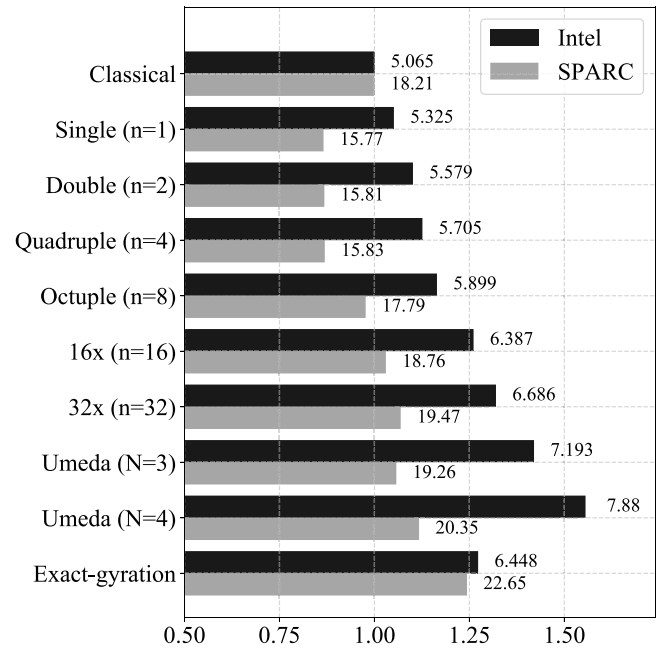


Fig. 4. Benchmark results of the test-particle simulations. The average elapse times (in seconds) are normalized to those of the classical Boris solver.

the Lorentz-force part:

$$\theta \equiv \frac{qB}{m\gamma}\Delta t, \tag{50}$$

$$\boldsymbol{u}_{\parallel} = (\boldsymbol{u}\cdot\hat{\boldsymbol{b}})\,\hat{\boldsymbol{b}} \tag{51}$$

$$\boldsymbol{u}^{+} = \boldsymbol{u}_{\parallel} + (\boldsymbol{u}-\boldsymbol{u}_{\parallel})\cos\theta + (\boldsymbol{u}\times\hat{\boldsymbol{b}})\sin\theta \tag{52}$$

By definition, this solver does not produce an error during the gyration phase. We refer to it as the exact-gyration solver. Since it uses trigonometric functions, its computation cost is usually higher than those of polynomial-based solvers. Note that all these solvers share the same Coulomb-force procedures (Eqs: (3) and (5)).

In the first test, we have run test-particle simulations in a static magnetic field $\boldsymbol{B} = (0, 0, 1)$ and $\boldsymbol{E} = (0, 0, 0)$. Parameters are set to $m = 1$, $q = 1$, and $c = 1$. The initial condition is $\boldsymbol{u}(t = 0) = (1, 0, 0)$ and the timestep is set to $\Delta t = \pi/60, \pi/20, \pi/6$, or $\pi/2$. We have calculated errors in $\boldsymbol{u}$ from the analytic solutions. We have run the simulations until $t = 12\pi$, and then we have evaluated the maximum errors in a 4-vector, $\{|\delta\boldsymbol{u}|/|\boldsymbol{u}|\}_{\max}$. Since $\omega_c = qB/(m\gamma) = 1/\sqrt{2}$, the interval 12π corresponds to ≈ 4 rotations. This test is a subset of numerical tests in our recent study [8]. Here, almost all of the solvers are combinations of the exact solver for the Coulomb-force parts (Eqs. (3) and (5)) and the second-order solver for the Lorentz-force part. These solvers only differ in the Lorentz-force part.

Fig. 3 shows the results as a function of Δt. Errors by the classical Boris solver and the Umeda solvers are not presented, because the errors are identical to those by the single, double, and quadruple Boris solvers. One can see that the multiple Boris schemes give second-order errors with respect to Δt. The errors also scale like $\propto n^{-2}$. This is because the error simply accumulates in the interval of our interest. From Eq. (34), the errors can be estimated to $\delta\theta \approx (\omega_c 12\pi)\frac{1}{12}(\omega_c \Delta t/n)^2 = (\pi/\sqrt{2})(\omega_c \Delta t/n)^2$. The results are in excellent agreement with the prediction. Meanwhile, the exact-gyration solver only gives small errors of $\mathcal{O}(10^{-13}) \sim \mathcal{O}(10^{-14})$.

In the second test, we have measured elapse times of test-particle simulations. Since algorithms' performance relies on CPU architectures, we have run the code on Intel and SPARC processors. We have compiled the program on an Intel Core i7 processor with Intel Fortran compiler 2019. For each solver, we have run the program for 1.2×10^7 timesteps. We have measured the elapse time six times for every cases. The black bars in Fig. 4 show the results, in normalized units. We have also benchmarked the program on a SPARC processor with FUJITSU Fortran compiler on FX100 supercomputer at the Japan Aerospace Exploration Agency (JAXA). The elapsed times are averaged over three runs, because the results on the SPARC processor have less variations than those on the Intel processor. The results are presented in the gray bars in Fig. 4. To emphasize the difference, the leftmost value is set to 0.5. Note that we show total elapse time of test-particle simulations. The programs contain not only the Lorentz-force part, but also the Coulomb-force parts (Eqs. (3) and (5)) and the position part (Eq. (1)). Thus the results are not proportional to, but highly affected by the computation time of the Lorentz-force solvers.

As far as we have tested, the new solvers are adequately fast. On the Intel processor, they are slower than the classical Boris solver, but faster than the Umeda solvers and the exact-gyration solver except for the $n = 32$ case. The multiple Boris solvers become slower as n increases. On the SPARC processor, the new solvers are more promising than on the Intel processor. Some of them ($n = 1$–4) are 50% faster than the exact-gyration solver and even faster than the classical Boris solver. It is noteworthy that the double Boris solver ($n = 2$) is significantly faster than the Umeda solver ($N = 3$), although the two solvers are equivalent. Similarly, the quadruple Boris solver ($n = 4$) is faster than the Umeda solver ($N = 4$). Surprisingly, even the 32x Boris solver ($n = 32$) looks as good as the Umeda solver ($N = 3$), even though it gives 256 times higher accuracy.

In the third test, we have run test-particle simulations with the single ($n = 1$) Boris solver, the quadruple ($n = 4$) Boris

Table 1
Field settings for test-particle simulations.

#	$\boldsymbol{B}$	$\boldsymbol{E}$	
1	(0,0,0)	(1,0,0)	Direct acceleration by **E**
2	(0,0,0.1)	(1,0,0)	**E**-dominated
3	(0,0,1)	(1,0,0)	$\lvert\boldsymbol{E}\rvert = c\lvert\boldsymbol{B}\rvert$
4	(0,0,1)	(0.1,0,0)	**E**×**B** drift
5	(0,0,1)	(0,0,0)	Gyration about **B**

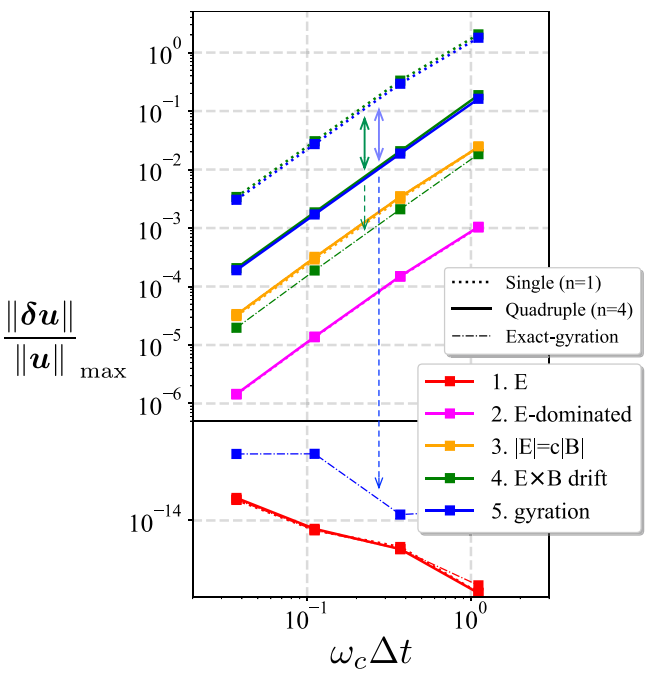


Fig. 5. Normalized errors in $\boldsymbol{u}$ in the test-particle simulations, as a function of $\theta = \omega_c \Delta t$. The single ($n = 1$) Boris solver (dotted lines), the quadruple ($n = 4$) solver (solid lines) and the exact-gyration solver (dashdotted lines) are tested in five configurations (colors; Table 1). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

solver, and the exact-gyration solver in various configurations. We consider five configurations, as shown in Table 1. In cases 1 and 2, the electric field **E** is dominant: $|\boldsymbol{E}| \gg c|\boldsymbol{B}|$. In case 3, the electric and magnetic fields are equal: $|E| = c|B|$. In cases 4 and 5, the magnetic field **B** is dominant: $|\boldsymbol{E}| \ll c|\boldsymbol{B}|$. In particular, case 5 stands for the gyration about **B**, as studied in the first test. Other parameters, timesteps, and the initial conditions are similar to those in the first test. We evaluate maximum values of the normalized error $(|\delta\boldsymbol{u}|/|u|)_{\max}$ during the time interval of $0 < t < 12\pi$. Here, $\delta\boldsymbol{u}$ is deviations from analytic solutions (cases 1 and 5) or numerical reference values (cases 2–4). We obtain the numerical reference values by using the exact-gyration solver with $\Delta t = \pi/240$. These configurations and procedures are the same as in Ref. [8].

Fig. 5 shows the errors as a function of the timestep. The dotted, solid, and dashdotted lines indicate the results by the single ($n = 1$) Boris solver, the quadruple ($n = 4$) solver, and the exact-gyration solver, respectively. In case 1 (in red) by the three solvers and case 5 by the exact-gyration solver, one can essentially see no error. All the other curves suggest the second-order accuracy. In cases 2 and 3, the three solvers return similar results. Meanwhile, in cases 4 and 5, their results are different. The quadruple Boris solver provides 10.9–16.5 times smaller errors than the single Boris solver, as emphasized by the solid arrows in green and blue. The improvement is asymptotic to 16 for $\omega_c \Delta t \ll 1$ in case 5. The exact-gyration solver provides even better results, as indicated by the dashed arrows. The quadruple Boris solver provides similar errors in cases 4 and 5. So does the single Boris solver. It appears that the accuracy in the Lorentz-force solver (case 5) limits the overall accuracy for $|\boldsymbol{E}| \ll c|\boldsymbol{B}|$ (case 4).

In the fourth test, we study the long-term stability of the particle motion in a non-uniform electromagnetic field. As done in Refs. [4,8], we have run test-particles in the following field,

$$\boldsymbol{B} = (x^2 + y^2)^{1/2}\boldsymbol{e}_z, \quad \phi = 0.01(x^2 + y^2)^{-1/2}. \tag{53}$$

The conditions are set to $\boldsymbol{u}(t = 0) = (0.1, 0.0)$, $\boldsymbol{x}(t = 0) = (0.9, 0, 0)$, $m = c = 1$, and $\Delta t = \pi/10$. We compare the single ($n = 1$) Boris solver, the quadruple ($n = 4$) Boris solver, the exact-gyration solver, and the fourth order Runge–Kutta solver. The blue and gray lines in Fig. 6 show the trajectories by the quadruple Boris solver and by the Runge–Kutta solver, at (a) an initial stage and at (b) a late stage. The time history of the total energy is presented in Fig. 6(c). It combines the particle kinetic energy and the electric potential energy at the midpoint positions ($\boldsymbol{x}(t + \frac{\Delta t}{2})$). The values are normalized by the initial values.

One can see typical trajectories in Fig. 6(a). It consists of a fast small-scale gyration and a slow large-scale rotation. The large-scale rotation is due to the ∇B drift and the **E**×**B** drift. In the case of the quadruple Boris solver, the particle keeps gyrating even after a long time (300th turn; $t \approx 2 \times 10^5$) in Fig. 6(b). The particle energy is excellently conserved. We only recognize a relative error of $\mathcal{O}(10^{-4})$. The trajectories (not shown), the time histories, and the error amplitudes (not shown) by the single ($n = 1$) Boris solver and the exact-gyration solver are similar to those of the quadruple ($n = 4$) Boris solver. In contrast, as evident in the gray line in Fig. 6(b), the particle gyroradius becomes smaller and smaller in the case of the Runge–Kutta solver. The particle numerically lost 30% of its total energy (Fig. 6(c)).

These results are attributed to the volume-preserving property of the numerical schemes. It is known that the Runge–Kutta solver is not volume preserving. The solver is incapable of preserving the energy of the small-scale oscillation. On the other hand, as discussed in Section 3, the multiple Boris solvers are volume-preserving. The other two solvers are also known to be volume-preserving [4,5,8]. These three schemes are capable of preserving the energy of the small-scale oscillation, even in a nonuniform field.

5. Discussion and summary

We have proposed the multiple Boris integrators that virtually use the Boris procedure n times in the Lorentz-force part. The schemes provide second-order but n^2-times smaller errors (Eq. (34)) than the classical Boris solver. The coefficients are given by either the generic forms of p (Eqs. (23)–(25)), the generic forms of t (Eqs. (27)–(29)), or the explicit forms (Eqs. (30)–(33)). In our experience, the generic forms with p are useful for large n. For small n ($n \leq 4$), we prefer the explicit form, because we can further optimize the code. In our implementation for $n = 1 \ldots 4$, we have calculated the numerators and denominators in Eqs. (30)–(33) separately, because the explicit forms share the same denominator, $(1 + t^2)^n$. As $\boldsymbol{t}$ contains $1/\gamma$ (Eq. (17)), we multiply the numerators and denominators by γ^{2n}. By doing so, we reduce the number of division operations to save the total computation cost.

The new solvers are developed in the same spirit as in the multistep Boris solvers [6,7], which considered 2^{N-2}-times smaller timesteps in the Lorentz-force part. In the multiple Boris solvers, we virtually divide the rotation angle into n Boris units. This n is arbitrary and not limited to 2^{N-2}. Thus one can choose the

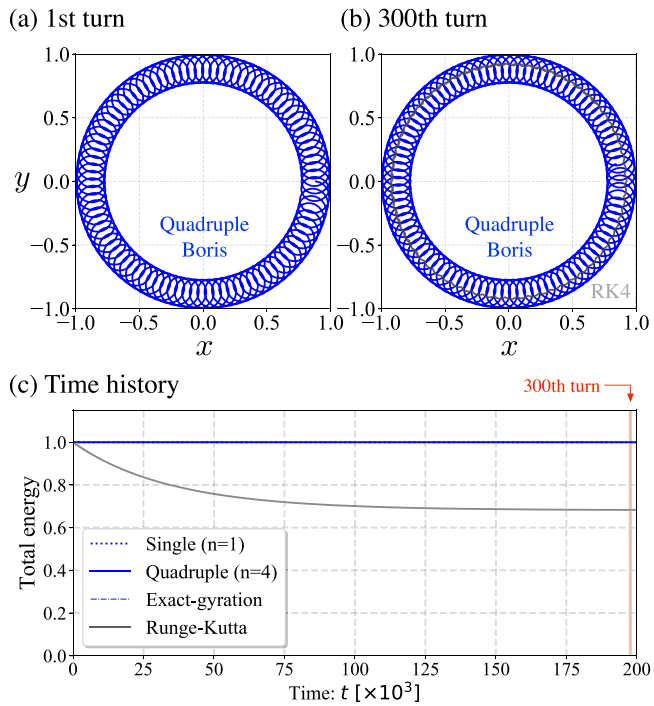


Fig. 6. Particle trajectories by the quadruple Boris solver at (a) the initial stage and (b) the late stage. A trajectory by the RK4 solver is overplotted in gray at (b) the late stage. (c) Energy histories by the four solvers. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

best n for one's application. In addition, we calculate the vector equation (Eq. (20)) only once, because we prepare the coefficients in advance. The coefficients are rational fractions, and so it is easy to calculate them. Consequently, the program runs efficiently, as evident in the benchmark results.

We have also evaluated numerical accuracy of several solvers in general cases of $\boldsymbol{B} \neq 0$ and $\boldsymbol{E} \neq 0$. Fig. 5 demonstrates the second-order accuracy in most cases, because the second-order Lorentz-force solvers are combined with the Coulomb-force procedures (Eqs. (3) and (5)) in an operator-splitting manner. In this study, since all the Lorentz-force solvers are combined with the same Coulomb-force procedures, the numerical accuracy for $|\boldsymbol{E}| \ll c|\boldsymbol{B}|$ is essentially controlled by the Lorentz-force solvers, even though the resulting schemes provide the same second-order accuracy. This is evident in cases 4 and 5 in Fig. 5 — the quadruple ($n = 4$) Boris solver provides $\approx$ 16 times better accuracy, and the exact-gyration solver, which corresponds to the multiple Boris solver in the $n \to \infty$ limit, provides even better accuracy. In general, the multiple Boris solvers have higher accuracy than the classic Boris solver, and then they bring better accuracy in the $|\boldsymbol{E}| \ll c|\boldsymbol{B}|$ cases. Importantly, we often consider $|\boldsymbol{E}| \ll c|\boldsymbol{B}|$ in practical applications in PIC simulation.

The new solvers may further reduce the computation cost, by allowing larger timesteps. In PIC simulation, the timestep Δt is limited by the plasma frequency ω_p and the gyrofrequency ω_c. The ratio of ω_p to ω_c depends on applications, and both frequencies are limited to $\omega\Delta t \lesssim \mathcal{O}(0.1)$ due to the numerical accuracy and $\omega\Delta t < 2$ due to the numerical stability [1]. If the gyrofrequency limits the timestep, the multiple Boris solvers can relax the limitations for the accuracy. As evaluated in Eq. (35), the n-tuple Boris solver allows n-times larger timestep, because it repeats the classical Boris solver with a smaller timestep ($\Delta t/n$) in the Lorentz-force part. Actually, in the other parts of PIC simulation, the timesteps are unchanged. For example, we update the position (Eq. (1)) for Δt, not for $\Delta t/n$. This fact makes the timestep condition more complicated. The new condition is given by Eq. (40). This is more restrictive than Eq. (35).

In the nonuniform electromagnetic field, the multiple Boris solvers have a favorable property of the long-term stability. Since they inherit a volume-preservation property from the Boris-type element, the new solvers are capable of preserving a small-scale oscillation, as they should be. As far as we have tested, they conserve the total energy very well. These facts give us further confidence to use the new solvers in PIC simulations for practical applications.

Among the new solvers, considering the accuracy, the timestep, and computation costs, we recommend the quadruple ($n = 4$) Boris solver with $\omega_c\Delta t \lesssim 0.5$ to the readers. One can develop higher-order solvers ($n \gg 32$) to obtain even better accuracy. In such a case, we recommend the exact-gyration solver [8] to the readers. As shown in Fig. 4, the 32x Boris solver ($n = 32$) is already more expensive than the exact-gyration solver on Intel processors. It is likely that higher-order solvers are as expensive as the exact-gyration solver on SPARC processors for $n \gg 32$. In any case, since the extra cost is limited, the exact-gyration solver will be another choice for $n > 10$. Finally, we note that our recommendation is partly based on the benchmarks on our specific systems. Performance may be different on other systems such as GPU-based systems and FGPA systems. In general, since it is impossible to predict future computing platforms, we can only prepare as many particle solvers as possible. Our solvers will be the latest additions.

In summary, we have proposed the multiple Boris solvers that virtually combine multiple Boris units in the Lorentz-force part. We have derived one-step expressions with polynomial-based coefficients for arbitrary n divisions. The new solvers provide second-order, n^2-times smaller errors than the classical Boris solver. They also allow larger timesteps. Their performance is

promising. They are stable over a long time. We hope that the proposed solvers will be useful in studies with PIC simulations.

Acknowledgments

This work was partially carried out on facilities of the JSS2 system at Japan Aerospace Exploration Agency (JAXA). This work was supported by Grant-in-Aid for Scientific Research (C), Japan 17K05673, (S) 17H06140, and (B) 17H02877 from the Japan Society for the Promotion of Science (JSPS).

Appendix. Third coefficients for $n = 2k + 1$

We recall the following relation of trigonometric functions.

$$\sin(A+\alpha) + \sin(A-\alpha) = 2\sin A\cos\alpha \tag{A.1}$$

We substitute $(2k+1)\alpha$ to A, where k is a positive integer, $k \geq 1$. Using Chebyshev polynomials, we obtain

$$\begin{aligned} 2\sin\big((2k+1)\alpha\big)\cos\alpha &= \sin\big(2(k+1)\alpha\big) + \sin\big(2k\alpha\big) \\ &= \sin 2\alpha\Big(U_k(\cos 2\alpha) + U_{k-1}(\cos 2\alpha)\Big) \end{aligned} \tag{A.2}$$

$$\sin\big((2k+1)\alpha\big) = \sin\alpha\Big(U_k(\cos 2\alpha) + U_{k-1}(\cos 2\alpha)\Big) \tag{A.3}$$

With help from Eqs. (7), (8), and (22), we obtain

$$\begin{aligned} c_{n3} &= \frac{2}{t^2}\sin^2\big((2k+1)\alpha\big) \\ &= \frac{2}{1+t^2}\left\{U_k\Big(\frac{1-t^2}{1+t^2}\Big) + U_{k-1}\Big(\frac{1-t^2}{1+t^2}\Big)\right\}^2 \\ &= (1+p)\Big\{U_k(p) + U_{k-1}(p)\Big\}^2 \end{aligned} \tag{A.4}$$

References

[1] C.K. Birdsall, A.B. Langdon, Plasma Physics Via Computer Simulation, McGraw-Hill, New York, 1985.
[2] R.W. Hockney, J.W. Eastwood, Computer Simulation using Particles, McGraw-Hill, New York, 1981.
[3] J.P. Boris, Proceedings of 4th Conference on Numerical Simulation of Plasmas, Naval Research Laboratory, Washington D. C., 1970, pp. 3–67.
[4] H. Qin, S. Zhang, J. Xiao, J. Liu, Y. Sun, W.M. Tang, Phys. Plasmas 20 (2013) 084503.
[5] R. Zhang, J. Liu, J. Liu, H. Qin, Y. Wang, Y. He, Y. Sun, Phys. Plasmas 22 (2015) 044501.
[6] T. Umeda, Comput. Phys. Commun. 228 (2018) 1.
[7] T. Umeda, Comput. Phys. Commun. 237 (2019) 37.
[8] S. Zenitani, T. Umeda, Phys. Plasmas 25 (2018) 112110.
[9] T.J. Rivin, Chebyshev Polynomials : From Approximation Theory to Algebra and Number Theory, second ed., Wiley, New York, 1990.