Applied Mathematics and Computation 442 (2023) 127706

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

New applications for the Boris Spectral Deferred Correction algorithm for plasma simulations

Kris Smedt [a], Daniel Ruprecht [b,*], Jitse Niesen [c], Steven Tobias [c], Joonas Nättilä [d]

[a] Centre for Doctoral Training in Fluid Dynamics, University of Leeds, Leeds, United Kingdom
[b] Lehrstuhl Computational Mathematics, Institut für Mathematik, Technische Universität Hamburg, Hamburg,Germany
[c] School of Mathematics, University of Leeds, Leeds,United Kingdom
[d] Physics Department and Columbia Astrophysics Laboratory, Columbia University, New York,USA

ARTICLE INFO

Article history:
Received 21 January 2022
Revised 16 September 2022
Accepted 6 November 2022
Available online 5 December 2022

MSC:
65L05
65M06
65M70

Keywords:
Boris integrator
spectral deferred corrections
particle-in-cell (PIC)
relativistic Lorentz equations

ABSTRACT

The paper investigates two new use cases for the Boris Spectral Deferred Corrections (Boris-SDC) time integrator for plasma simulations. First, we show that using Boris-SDC as a particle pusher in an electrostatic particle-in-cell (PIC) code can, at least in the linear regime, improve simulation accuracy compared with the standard second order Boris method. In some instances, the higher order of Boris-SDC even allows a much larger time step, leading to modest computational gains. Second, we propose a modification of Boris-SDC for the relativistic regime. Based on an implementation of Boris-SDC in the RUNKO PIC code, we demonstrate for a relativistic Penning trap that Boris-SDC retains its high order of convergence for velocities ranging from $0.5c$ to $> 0.99c$.

1. Introduction

Movement of charged particles in an electromagnetic field is described by the Lorentz equations

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}, \tag{1a}$$

$$\frac{d\mathbf{v}}{dt} = \frac{q}{m}(\mathbf{E}(\mathbf{x},t) + \mathbf{v}(t) \times \mathbf{B}(\mathbf{x},t)) =: \mathbf{f}(\mathbf{x},\mathbf{v},t), \tag{1b}$$

where $m\mathbf{f}$ is the force on a particle with charge q and mass m, $\mathbf{x}$ is the particle position and $\mathbf{v}$ its velocity. The Lorentz equations have a wide range of applications and are particularly relevant for modeling plasmas. Understanding plasma dynamics is important since an estimated 99% of all visible matter in the universe are in a plasma state [1].

One of the most popular numerical algorithms for solving (1) was introduced by Boris in 1970 [2]. He proposed a Leapfrog method combined with a clever geometric trick to resolve the implicit dependence arising from the $\mathbf{v} \times \mathbf{B}$ term

* Corresponding author.
E-mail address: ruprecht@tuhh.de (D. Ruprecht).

https://doi.org/10.1016/j.amc.2022.127706

in (1b) based on the observation that the magnetic field only rotates the particle trajectory but does not change the magnitude of its velocity. Boris' trick can be applied both to leapfrog integration (velocity defined at half time-steps) and Velocity-Verlet integration (velocity and position both defined on integer time-steps). In either case, the Boris integrator is second order accurate and conserves phase-space volume, giving it favourable long-term energy behaviour [3]. There is also a detailed mathematical analysis available, showing that for spatially varying magnetic fields, Boris can still exhibit linear energy drift [4].

A number of explicit high order integrators for (1) have been developed recently [5–10]. Most are derived from the Hamiltonian of (1) using splitting methods [11]. However, very few studies compare them with respect to computational efficiency and none so far investigates their use as a particle pusher in a particle-in-cell code. Quandt [9] and Li [7] compare computational efficiency of their proposed integrators to classic Boris. High order integrators were found to show the expected order of convergence and could outperform standard Boris in terms work-precision for some configurations.

Winkel et al. introduced Boris-SDC in 2015 [12], a combination of the Boris method with the spectral deferred correction (SDC) algorithm by Dutt et al. [13]. They demonstrate that Boris-SDC delivers high-order accuracy for both a single particle and a particle cloud in a Penning trap and that it leads to less numerical heating than the Boris algorithm. Tretiak and Ruprecht [14] combine Boris-SDC with a GMRES-based convergence accelerator originally proposed by Huang et al. [15]. They show that the resulting BGSDC method can deliver improvements in performance over the standard Boris method when simulating fast ions in idealised magnetic fields. In 2021 they extended these results, showing that BGSDC can improve performance for large ensembles of particles and realistic equilibrium fields of the DIIID and JET Tokamak fusion reactor [16]. However, these three studies consider only non-relativistic cases where particles travel passively through an EM-field guiding them. Here, we extend their results in two ways. First, we investigate numerically the performance of Boris-SDC when used as a particle-pusher in a particle-in-cell code [17,18]. In this case, the particles are no longer passively guided through an electromagnetic field but modify the field. Second, we introduce a modification to Boris-SDC for the relativistic case and demonstrate that it retains high order convergence.

While there is some research about potential benefits of using higher order methods in PIC, the focus is mostly on spatial operations like interpolation, deposition or mesh-based approximations of derivatives. Xiao et al. [19] propose a combination of splitting methods applied to the Hamiltonian version of (1) and specialised finite elements to produce geometric PIC algorithms capable of high order in both space and time. Energy conservation was demonstrated, but no comparison of computational efficiency was made. Shalaby et al. [20] study the performance of an ESPIC code with higher order algorithms for the field interpolations (up to fifth order) while solving for the field exactly. They explicitly highlight the limitation imposed by having only second order time-stepping because of the used Boris/Leapfrog pusher. Moreover, the energy conservation with charge-conserving PIC algorithms is still being actively studied [21].

Contributions. The paper provides the first investigation of how Boris-SDC performs as a particle pusher in a particle-in-cell code. It is also one of very few studies that analyses how a high(er) order pusher affects the overall accuracy of PIC. Furthermore, the paper generalizes the original Boris-SDC algorithm to the relativistic regime. For three benchmark problems, a two-stream instability, Landau damping and a relativistic Penning trap, we show that the higher order of Boris-SDC leads to substantially better accuracy compared to the standard Boris method. However, in the settings that we tested and without further modifications, the better accuracy is not enough to achieve computational gains from being able to take larger time step: in work-precision studies, the additional work per time step mostly offsets the saved cost from taking fewer but larger steps and Boris-SDC and Boris deliver similar performance. Therefore, while Boris-SDC is probably not yet at a stage where its benefits would warrant take-up by users of PIC, our results suggest that its continued development by numerical analysts providing algorithms for PIC is promising. Because Boris-SDC and Boris are similar in performance, any improvements on Boris-SDC, be it from convergence accelerators [15,22] or parallel-in-time approaches [23,24], will immediately translate into speedup over vanilla Boris.

2. Spectral deferred corrections as pusher for particle-in-cell (PIC)

Before we describe the Boris spectral deferred correction (Boris-SDC) algorithm, we briefly revisit the key components of the particle-in-cell (PIC) method. For the sake of simplicity, we restrict our presentation to the one-dimensional case but the generalization to 3D is straightforward, although more complicated in terms of indexing. A detailed overview is provided for example by Verboncoeur [25] and a detailed introduction can be found in the seminal textbook by Birdsall and Landon [26].

2.1. Particle-in-Cell (PIC)

PIC tracks particles in a Lagrangian approach but has the electric and magnetic field they generate "live" on a mesh. This avoids the $\mathcal{O}(N^2)$ bottleneck that emerges if all particle interactions are computed directly. The main assumption of PIC is that the global fields arising from the distribution and movement of the full collection of particles are dominant. A detailed discussion of the main assumptions underlying particle-in-cell methods is given in Chapter 1 of the seminal book by Birdsall and Langdon [26].

Figure 1 sketches the components of one time step in PIC. This paper focuses on the particle velocity and position update where, using the fields computed in the steps before, particles are moved around by numerically integrating the Lorentz

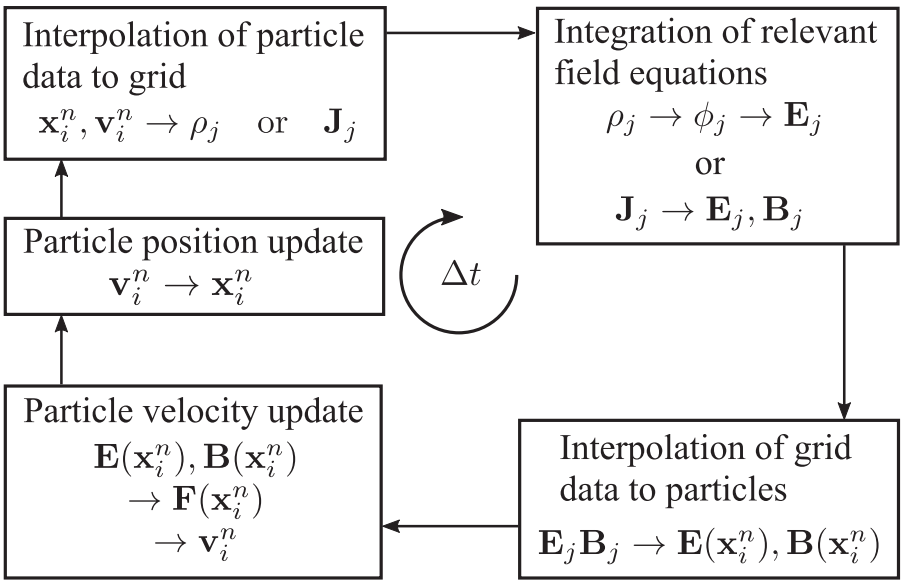

Fig. 1. Steps in a single time step of a general PIC scheme. See e.g. the review by Verboncoeur et al for details [25].

equations. We investigate how the higher order of accuracy provided by Boris-SDC affects the overall approximation quality of the PIC method.

The exact order of operations in PIC depend on the chosen particle integrator for solving the equation of motion. For particle integrators in which the position and velocity are staggered in time, the PIC time-step begins with the velocity update, followed by the position update and field solutions. For particle integrators where position and velocity are both defined at the integer time-steps (synchronised), a PIC time-step begins with the position update, followed by the corresponding field solution and finishes with the calculation of the new velocity. Boris-SDC and Velocity-Verlet, the second order integrator on which it is based, are synchronised particle pushers and so the second type of PIC setup was used throughout this study. Whenever the "Boris integrator" term is applied in this study, it refers to Boris' algorithm applied to the velocity-Verlet integrator unless otherwise noted. However, note that owing to the popularity of the leapfrog integrator in PIC, most existing schemes are of the staggered type.

2.1.1. Integration of relevant field equations

In the electromagnetic case, time derivatives for the electric and magnetic field are present in the Maxwell-Vlasov equations that need to be integrated numerically. While this step is shown in Fig. 1 for the sake of completeness, we only study electrostatic examples in this paper where the electric field is fully reconstructed from the particle charges in every time step. A detailed survey of different numerical approaches to electromagnetic PIC is provided by Birdsall and Langdon [26]. In particular, we do not discuss the issues of divergence correction or cleaning that arises if Gauss' law is not exactly satsified on the discrete level [27] and leave this for future work.

2.1.2. Interpolation of grid data to particles

In the electrostatic case, the electric field is given by

$$\mathbf{E} = \nabla\phi \tag{2}$$

where ϕ is the electrostatic potential. The potential depends on the charge density ρ via

$$\nabla^2\phi(\mathbf{x}, t) = \frac{\rho(\mathbf{x}, t)}{\epsilon}, \tag{3}$$

using the permittivity of the plasma ϵ. Now consider N_p particles, where $n = 1, \ldots, N_p$ labels a given particle with position vector $\mathbf{x}_n$ in continuous space and velocity $\mathbf{v}_n$. Let x_i with $i = 1, \ldots, N_x$ be a set of equidistant mesh points with spacing $x_{i+1} - x_i = \Delta x$. To calculate the electric field on the grid, we need to determine the corresponding charge densities ρ_i that are generated by the particles. To do this, particle charges q are interpolated to the surrounding grid nodes via some weighting function $W(\mathbf{x}_n)$. Here, we use linear weighting to "scatter" the particle charge to the two nearest grid nodes so that

$$W(x) = \begin{cases} 1 - \frac{|x - x_i|}{\Delta x} & |x - x_i| < \Delta x \\ 0 & |x - x_i| > \Delta x. \end{cases} \tag{4}$$

Other interpolation schemes exist, such as Nearest-Grid-Point or higher order quadratic or cubic weighting splines, but the linear scheme is most commonly used [25].

Once the charge of the surrounding volume has been assigned to a grid nodes, the charge density ρ_i for the cell is computed as the average over the cell volume. Knowledge of the charge densities ρ_i allows the determination of the electrostatic potential with an appropriate solution scheme. For the current study, 1D second order central finite difference was used for the grid. Discretizing (3) with second order centered finite differences yields

$$\frac{\phi_{i-1} - 2\phi_i + \phi_{i+1}}{\Delta x^2} = \frac{\rho_i}{e}. \tag{5}$$

The resulting linear system is solved for the ϕ_i using the SciPy linear algebra package [28]. From the grid values, the gradient of the electric potential ϕ_i and thus the electric field values are computed with a central difference scheme

$$\mathbf{E}_i = \frac{\phi_{i-1} - \phi_{i+1}}{2\Delta x}. \tag{6}$$

At boundary nodes, forward

$$\mathbf{E}_1 = \frac{\phi_1 - \phi_2}{\Delta x}, \tag{7}$$

or backward finite differences

$$\mathbf{E}_{N_x} = \frac{\phi_{N_x} - \phi_{N_x-1}}{\Delta x}, \tag{8}$$

are used instead.

2.1.3. Particle velocity and position update

The Newton-Lorentz force gives the acceleration exerted on the particles. The corresponding differential Eq. (1) is integrated numerically to update velocity and position of the particles from time t_n to time $t_n + \Delta t = t_{n+1}$. A popular algorithm is the Störmer-Verlet scheme

$$\mathbf{v}_{n+1/2} = \mathbf{v}_{n-1/2} + \frac{\Delta t}{2}\mathbf{f}(\mathbf{x}_n, \mathbf{v}_n) \tag{9a}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t \mathbf{v}_{n+1/2} \tag{9b}$$

where the calculation of position and velocity are offset by $\Delta t/2$. Here, $\mathbf{x}_n \approx \mathbf{x}(t)$ and $\mathbf{x}_{n+1} \approx \mathbf{x}(t + \Delta t)$, etc. Note that some form of interpolation is required to provide $\mathbf{v}_n$ in (9a). Typically, the average of $\mathbf{v}_{n+1/2}$ and $\mathbf{v}_{n-1/2}$ is used. This staggering is advantageous especially on fully electromagnetic PIC loops where the electromagnetic fields can then be evolved with a finite-difference time domain (FDTD) method relying on the so-called Yee lattice [25]. However, for Boris-SDC, staggering was found to increase storage requirements without adding much benefit [14]. We therefore use the second order accurate velocity-Verlet scheme

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_n \Delta t + \frac{1}{2}\mathbf{f}(\mathbf{x}_n, \mathbf{v}_n)\Delta t^2, \tag{10a}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \frac{\mathbf{f}(\mathbf{x}_n, \mathbf{v}_n) + \mathbf{f}(\mathbf{x}_{n+1}, \mathbf{v}_{n+1})}{2}\Delta t, \tag{10b}$$

instead. Both variants are second order accurate and behave very similarly, but they are not equivalent [29]. Boris-SDC, introduced in detail below, is a high-order generalization of (10).

Boris' trick.

While the position update (10a) is explicit, the update for the velocity (10b) is implicit. Boris introduced a simple, geometrical procedure to find $\mathbf{v}_{n+1}$ [2]. We use his trick in a slightly different way than usual, as a generic solver for an equation of the form

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \alpha\mathbf{E} + \beta\frac{\mathbf{v}_n + \mathbf{v}_{n+1}}{2} \times \mathbf{B} + \mathbf{c}, \tag{11}$$

where α and β are some given scalar parameters and $\mathbf{E}, \mathbf{B}$ and $\mathbf{c}$ are some given vectors. Note that $\mathbf{c}$ does not normally feature in most variants of the Boris integrator. However, we will need it later as a "container" for various terms that arise from the Boris-SDC iteration. Typically, α and β are equal to the time step Δt but we will generalise this when deriving the relativistic variant of Boris-SDC. Furthermore, $\mathbf{E}$ will be the average of the electric fields at $\mathbf{x}_n$ and $\mathbf{x}_{n+1}$, $\mathbf{B}$ the magnetic field evaluated at some specific position whereas $\mathbf{c}$ will collect terms related, e.g., to the quadrature needed in Boris-SDC. When used in this form, Boris' trick becomes Algorithm 1. Note that there a multiple slightly different forms of Boris' trick. In the terminology used by Zenitani and Umeda, we use the Boris-B algorithm [30].

Algorithm 1: Boris' trick as a general solver for (11). See Birdsall and Langdon [26, Section 4-4] for the geometric derivation.

input : $\mathbf{v}_{n-1}, \alpha, \beta, \mathbf{B}, \mathbf{E}, \mathbf{c}$
output: $\mathbf{v}_n$ solving $\mathbf{v}_n = \mathbf{v}_{n-1} + \alpha\mathbf{E} + \beta\frac{\mathbf{v}_{n-1}+\mathbf{v}_n}{2} \times \mathbf{B} + \mathbf{c}$

1.1 $\mathbf{t} = \frac{\beta}{2}\mathbf{B}$
1.2 $\mathbf{s} = 2\mathbf{t}/(1 + \mathbf{t} \cdot \mathbf{t})$
1.3 $\mathbf{v}^- = \mathbf{v}_{n-1} + \frac{\alpha}{2}\mathbf{E} + \frac{1}{2}\mathbf{c}$
1.4 $\mathbf{v}^* = \mathbf{v}^- + \mathbf{v}^- \times \mathbf{t}$
1.5 $\mathbf{v}^+ = \mathbf{v}^- + \mathbf{v}^* \times \mathbf{s}$
1.6 $\mathbf{v}_n = \mathbf{v}^+ + \frac{\alpha}{2}\mathbf{E} + \frac{1}{2}\mathbf{c}$

2.1.4. Interpolation of particle data to grid

To compute $\mathbf{E}(\mathbf{x})$ in (1), we need to calculate the electrical field at the position of a particle from the mesh point values computed in Subsection 2.1.2. To do so, linear interpolation is performed to collect the corresponding field value at each particle position. The value will be a sum of contributions from the surrounding nodes, each node contributing a field strength equal to the electric field at the node weighted by Eq. (4). Any imposed, background electric and magnetic field can either be added to the nodes and interpolated or evaluated directly at the particle positions. For the simulations in this study, a neutralising static background electric field is imposed at the grid nodes. In the electrostatic case, this procedure is repeated in every time step.

2.2. Boris-SDC

Boris-SDC is a time integration scheme for (1) that provides tuneable order of accuracy. There are two slightly different versions. The one by Winkel et al. [12] involves a substitution for velocity in the position update which can improve accuracy. The substitution was dropped in the second variant by Tretiak and Ruprecht [14] to allow for the use of a GMRES-based convergence acceleration technique. Both variants are based on collocation: the differential Eq. (1) is turned into an integral equation

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_{t_0}^{t} \mathbf{v}(s)\ ds, \tag{12a}$$

$$\mathbf{v}(t) = \mathbf{v}_0 + \int_{t_0}^{t} \mathbf{f}(\mathbf{x}(s), \mathbf{v}(s))\ ds, \tag{12b}$$

where $\mathbf{x}_0, \mathbf{v}_0$ are approximations of $\mathbf{x}(t_0), \mathbf{v}(t_0)$ brought forward from the previous time step. In the formulation by Winkel et al. [12], the second equation is substituted into the first one so that

$$\mathbf{x}(t) = \mathbf{x}_0 + (t - t_0)\mathbf{v}_0 + \int_{t_0}^{t}\int_{t_0}^{r} \mathbf{f}(\mathbf{x}(s), \mathbf{v}(s))\ ds\ dr, \tag{13a}$$

$$\mathbf{v}(t) = \mathbf{v}_0 + \int_{t_0}^{t} \mathbf{f}(\mathbf{x}(s), \mathbf{v}(s))\ ds. \tag{13b}$$

To compute an update from t_0 to $t_0 + \Delta t =: t_1$, the integrals are approximated using quadrature with respect to nodes $t_0 \le \tau_1 < \ldots < \tau_M \le t_1$. Letting $\mathbf{x}_m, \mathbf{v}_m$ denote approximations for $\mathbf{x}(\tau_m), \mathbf{v}(\tau_m)$ for $m = 1, \ldots, M$, these approximations read

$$\int_{t_0}^{t_1} \mathbf{v}(s)\ ds \approx \sum_{m=1}^{M} q_m \mathbf{v}_m, \tag{14a}$$

$$\int_{t_0}^{t_1} \mathbf{f}(\mathbf{x}(s), \mathbf{v}(s))\ ds \approx \sum_{m=1}^{M} q_m \mathbf{f}(\mathbf{x}_m, \mathbf{v}_m), \tag{14b}$$

where the q_m are quadrature weights. Equations for the approximate values $\mathbf{x}_m, \mathbf{v}_m$ can be derived by inserting $t = \tau_m$ into (13) obtaining

$$\mathbf{x}_m = \mathbf{x}_0 + \mathbf{v}_0 \sum_{j=1}^{m} \Delta\tau_j + \sum_{j=1}^{M} q_{m,j} \sum_{k=1}^{M} q_{j,k} \mathbf{f}(\mathbf{x}_k, \mathbf{v}_k), \tag{15a}$$

$$\mathbf{v}_m = \mathbf{v}_0 + \sum_{j=1}^{M} q_{m,j} \mathbf{f}(\mathbf{x}_j, \mathbf{v}_j), \tag{15b}$$

where $\Delta\tau_j := \tau_j - \tau_{j-1}$ for $j = 1, \dots, M$ and the $q_{m,j}$ are quadrature weights to approximate integrals $\int_{t_0}^{\tau_m}(\cdot)\, ds$. By subtracting the equations for index m and $m-1$, the equations can be written in a node-to-node form

$$\mathbf{x}_m = \mathbf{x}_{m-1} + \Delta\tau_m \mathbf{v}_0 + \sum_{j=1}^{M} sq_{m,j}\mathbf{f}(\mathbf{x}_k, \mathbf{v}_k), \tag{16a}$$

$$\mathbf{v}_m = \mathbf{v}_{m-1} + \sum_{j=1}^{M} s_{m,j}\mathbf{f}(\mathbf{x}_j, \mathbf{v}_j), \tag{16b}$$

where $s_{m,j} = q_{m,j} - q_{m-1,j}$ and the $sq_{m,j}$ can be worked out by rearranging the sum

$$\sum_{j=1}^{M}\left(q_{m,j} - q_{m-1,j}\right)\sum_{k=1}^{M} q_{j,k} = \sum_{j=1}^{M} s_{m,j}\sum_{k=1}^{M} q_{j,k}, \tag{17}$$

see the Appendix in Winkel et al. [12] for details. Note that the equations for the $\mathbf{x}_m, \mathbf{v}_m$ are all coupled so that solving for them directly would require using a Newton iteration for a very large system of equations. Instead, Boris-SDC computes approximations via a different iterative scheme where updates can be computed by a "sweep" of normal Boris integrator steps.

Skipping the derivation, which can also be found in Winkel et al. [12], the Boris-SDC iteration reads

$$\mathbf{x}_m^{k+1} = \mathbf{x}_{m-1}^{k+1} + \Delta\tau_m \mathbf{v}_0 + \sum_{j=1}^{m-1} s^X_{m,j}\left(\mathbf{f}(\mathbf{x}_j^{k+1}, \mathbf{v}_j^{k+1}) - \mathbf{f}(\mathbf{x}_j^{k}, \mathbf{v}_j^{k})\right) + \sum_{j=1}^{M} sq_{m,j}\mathbf{f}(\mathbf{x}_j^{k}, \mathbf{v}_j^{k}) \tag{18a}$$

$$\begin{aligned}\mathbf{v}_m^{k+1} &= \mathbf{v}_{m-1}^{k+1} + \frac{\Delta\tau_m}{2}\left(\mathbf{f}(\mathbf{x}_m^{k+1}, \mathbf{v}_m^{k+1}) - \mathbf{f}(\mathbf{x}_m^{k}, \mathbf{v}_m^{k})\right), \\ &\quad + \frac{\Delta\tau_m}{2}\left(\mathbf{f}(\mathbf{x}_{m-1}^{k+1}, \mathbf{v}_{m-1}^{k+1}) - \mathbf{f}(\mathbf{x}_{m-1}^{k}, \mathbf{v}_{m-1}^{k})\right) + \sum_{j=1}^{M} s_{m,j}\mathbf{f}(\mathbf{x}_j, \mathbf{v}_j), \quad m = 1, \dots, M,\end{aligned} \tag{18b}$$

with k counting iterations. The coefficients $s^X_{m,j}$ can be worked out from the distances $\Delta\tau_j$ between quadrature nodes, see again the Appendix in Winkel et al. [12]. If the iteration converges and $\mathbf{x}_m^{k+1} - \mathbf{x}_m^k \to 0$ and $\mathbf{v}_m^{k+1} - \mathbf{v}_m^k \to 0$, equations (18) reduce to the collocation Eq. (16). Note that the position update is explicit: if we know the values from the previous iteration k and all the values up to $\mathbf{x}_{m-1}^{k+1}$, we can directly compute $\mathbf{x}_m^{k+1}$ and so on. In contrast, the velocity update is implicit, but we can use Boris' trick to compute $\mathbf{v}_m^{k+1}$. To avoid cluttering the notation we assume that the charge-to-mass ratio q/m is equal to one here. If that is not the case, just multiply the $\Delta\tau_m$ factors in front of the electric and magnetic field terms by q/m. Let

$$\mathbf{c}_m^k := -\frac{\Delta\tau_m}{2}\mathbf{f}(\mathbf{x}_m^k, \mathbf{v}_m^k) - \frac{\Delta\tau_m}{2}\mathbf{f}(\mathbf{x}_{m-1}^k, \mathbf{v}_{m-1}^k) + \sum_{l=1}^{M} s_{m,l}\mathbf{f}(\mathbf{x}_l^k, \mathbf{v}_l^k) \tag{19}$$

collect all terms from the previous iteration with index k. Expanding $\mathbf{f}$, the velocity update in the Boris-SDC iteration then reads

$$\mathbf{v}_m^{k+1} = \mathbf{v}_{m-1}^{k+1} + \Delta\tau_m \frac{\mathbf{E}(\mathbf{x}_{m-1}^{k+1}) + \mathbf{E}(\mathbf{x}_m^{k+1})}{2} + \Delta\tau_m \frac{\mathbf{B}(\mathbf{x}_{m-1}^{k+1}) \times \mathbf{v}_{m-1}^{k+1} + \mathbf{B}(\mathbf{x}_m^{k+1}) \times \mathbf{v}_m^{k+1}}{2} + \mathbf{c}_m^k. \tag{20}$$

To bring this into the form (11), we add $-\mathbf{B}(\mathbf{x}_m^{k+1}) \times \mathbf{v}_{m-1}^{k+1} + \mathbf{B}(\mathbf{x}_m^{k+1}) \times \mathbf{v}_{m-1}^{k+1} = 0$ and let

$$\tilde{\mathbf{c}}_m^k := \Delta\tau_m \frac{-\mathbf{B}(\mathbf{x}_m^{k+1}) \times \mathbf{v}_{m-1}^{k+1} + \mathbf{B}(\mathbf{x}_{m-1}^{k+1}) \times \mathbf{v}_{m-1}^{k+1}}{2} + \mathbf{c}_m^k \tag{21}$$

so that, setting $\mathbf{E} := \frac{1}{2}\left(\mathbf{E}(\mathbf{x}_{m-1}^{k+1}) + \mathbf{E}(\mathbf{x}_m^{k+1})\right)$ and $\mathbf{B} := \mathbf{B}(\mathbf{x}_m^{k+1})$, the velocity update becomes

$$\mathbf{v}_m^{k+1} = \mathbf{v}_{m-1}^{k+1} + \Delta\tau_m \mathbf{E} + \Delta\tau_m \frac{\mathbf{v}_{m-1}^{k+1} + \mathbf{v}_m^{k+1}}{2} \times \mathbf{B} + \tilde{\mathbf{c}}_m^k. \tag{22}$$

This can now be solved using Algorithm 1. One time step of Boris-SDC then consists of the following steps:

1. Initialise $\mathbf{x}_m^0 = \mathbf{x}_0$ and $\mathbf{v}_m^0 = \mathbf{v}_0$ for $m = 1, \dots, M$.
2. Perform K sweeps:
   (a) Evaluate $\mathbf{f}(\mathbf{x}^k, \mathbf{v}_m^k)$ for $m = 1, \dots, M$.
   (b) Update $\mathbf{x}_m^k \to \mathbf{x}_m^{k+1}$ and $\mathbf{v}_m^k \to \mathbf{v}_m^{k+1}$ for $m = 1, \dots, M$ using (18).
3. If $\tau_M = t_{n+1}$, that is the end of the step is a quadrature node, set $\mathbf{x}_M^K \to \mathbf{x}_0$ and $\mathbf{v}_M^K \to \mathbf{v}_0$ and start the next time step.

Throughout this paper, we use Gauss-Lobatto nodes for quadrature. If other nodes are used, a final quadrature step is needed to deliver the approximate value at t_{n+1}. Note that because we initialize by spreading the initial value to all nodes, Boris-SDC with $K = 1$ iterations is not equivalent to vanilla Boris. It would also be possible to initialize with a single sweep of Boris instead but in most cases, spreading the initial value before the first iteration seemed to provide slightly better results.

2.3. Relativistic Boris-SDC

The relativistic Newton-Lorentz system in cgs units reads

$$\frac{d\mathbf{x}}{dt} = \frac{\mathbf{u}}{\gamma(\mathbf{u})} =: \mathbf{g}(\mathbf{u}), \tag{23a}$$

$$\frac{d\mathbf{u}}{dt} = \frac{q}{m}\left(\mathbf{E}(\mathbf{x}) + \frac{\mathbf{g}(\mathbf{u})}{c} \times \mathbf{B}(\mathbf{x})\right) =: \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{23b}$$

where $\mathbf{u}$ is the proper velocity (the spatial component of the four-velocity) with the function $\mathbf{g}(\mathbf{u})$ yielding the coordinate velocity $\mathbf{v}$. Following Griffiths [31], the relativistic Lorentz factor calculated from the proper velocity is

$$\gamma(\mathbf{u}) = \sqrt{1 + \mathbf{u} \cdot \mathbf{u}/c^2}. \tag{24}$$

Written in this form, the system has the structure of a general second order initial value problem [32]. We consider the proper velocity $\mathbf{u}$ as the variable to solve for and, as far as the time stepping scheme is concerned, treat the coordinate velocity $\mathbf{v}$ as an auxiliary quantity. In integral form, (23) becomes

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_{t_0}^{t} \mathbf{g}(\mathbf{v}(s))\ ds, \tag{25a}$$

$$\mathbf{v}(t) = \mathbf{v}_0 + \int_{t_0}^{t} \mathbf{f}(\mathbf{x}(s), \mathbf{v}(s))\ ds. \tag{25b}$$

Substituting as for the non-relativistic case would result in

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_{t_0}^{t} \mathbf{g}\left(\mathbf{v}_0 + \int_{t_0}^{r} \mathbf{v}(s)\ ds\right) dr. \tag{26}$$

While this integral can be approximated by quadrature, it is not clear how the resulting SDC iteration can be written in a sweep-like fashion comparable to (18). We therefore use the less accurate formulation without substitution for the relativistic case and leave the derivation of a relativistic sweep with substitution for future work. For the non-relativistic Newton-Lorentz equations (1) the SDC sweep without substitution reads

$$\mathbf{x}_{m+1}^{k+1} = \mathbf{x}_m^{k+1} + \Delta\tau_m(\mathbf{v}_{m+1/2}^{k+1} - \mathbf{v}_{m+1/2}^{k}) + \sum_{j=1}^{M} s_{m,j}\mathbf{v}_j^k, \tag{27}$$

where

$$\mathbf{v}_{m+1/2}^{k} = \mathbf{v}_m^k + \frac{\Delta\tau_m}{2}\mathbf{f}(\mathbf{x}_m^k, \mathbf{v}_m^k). \tag{28}$$

For the velocity, the iterations reads

$$\mathbf{v}_{m+1}^{k+1} = \mathbf{v}_m^{k+1} + \frac{\Delta\tau_m}{2}\left[\mathbf{f}(\mathbf{x}_m^{k+1}, \mathbf{v}_m^{k+1}) + \mathbf{f}(\mathbf{x}_{m+1}^{k+1}, \mathbf{v}_{m+1}^{k+1})\right] - \frac{\Delta\tau_m}{2}\left[\mathbf{f}(\mathbf{x}_m^{k}, \mathbf{v}_m^{k}) + \mathbf{f}(\mathbf{x}_{m+1}^{k}, \mathbf{v}_{m+1}^{k})\right] + \sum_{j=1}^{M} s_{m,j}\mathbf{f}(\mathbf{x}_j^k, \mathbf{u}_j^k). \tag{29}$$

For the relativistic Lorentz equations, the SDC sweep for the position becomes

$$\mathbf{x}_{m+1}^{k+1} = \mathbf{x}_m^{k+1} + \Delta\tau_m\left[\mathbf{g}(\mathbf{u}_{m+1/2}^{k+1}) - \mathbf{g}(\mathbf{u}_{m+1/2}^{k})\right] + \sum_{j=1}^{M} s_{m,j}\mathbf{g}(\mathbf{u}_j^k), \tag{30}$$

where

$$\mathbf{u}_{m+1/2}^{k+1} = \mathbf{u}_m^{k+1} + \frac{\Delta\tau_m}{2}\mathbf{f}\left(\mathbf{x}_m^{k+1}, \mathbf{u}_m^{k+1}\right), \tag{31}$$

and the sweep for the velocity

$$\mathbf{u}_{m+1}^{k+1} = \mathbf{u}_m^{k+1} + \frac{\Delta\tau_m}{2}\left[\mathbf{f}(\mathbf{x}_m^{k+1}, \mathbf{u}_m^{k+1}) + \mathbf{f}(\mathbf{x}_{m+1}^{k+1}, \mathbf{u}_{m+1}^{k+1})\right] - \frac{\Delta\tau_m}{2}\left[\mathbf{f}(\mathbf{x}_m^{k}, \mathbf{u}_m^{k}) + \mathbf{f}(\mathbf{x}_{m+1}^{k}, \mathbf{u}_{m+1}^{k})\right] + \sum_{j=1}^{M} s_{m,j}\mathbf{f}(\mathbf{x}_j^k, \mathbf{u}_j^k). \tag{32}$$

However, properly applying the Boris trick in the relativistic case requires some care. Expanding $\mathbf{f}$ and defining

$$\mathbf{c}_m^k = \frac{\Delta\tau_m}{2c}\mathbf{g}(\mathbf{u}_m^{k+1}) \times \mathbf{B}(\mathbf{x}_m^{k+1}) - \frac{\Delta\tau_m}{2}\left[\mathbf{f}(\mathbf{x}_m^{k}, \mathbf{u}_m^{k}) + \mathbf{f}(\mathbf{x}_{m+1}^{k}, \mathbf{u}_{m+1}^{k})\right] + \sum_{j=1}^{M} s_{m,j}\mathbf{f}(\mathbf{x}_j^k, \mathbf{u}_j^k) \tag{33}$$

gives the update

$$\mathbf{u}_{m+1}^{k+1} = \mathbf{u}_m^{k+1} + \Delta\tau_m \left(\frac{\mathbf{E}(\mathbf{x}_m^{k+1}) + \mathbf{E}(\mathbf{x}_{m+1}^{k+1})}{2} + \frac{1}{2c}\mathbf{g}(\mathbf{u}_{m+1}^{k+1}) \times \mathbf{B}(\mathbf{x}_{m+1}^{k+1}) \right) + \mathbf{c}_m^k. \tag{34}$$

Setting

$$\mathbf{E} := \frac{\mathbf{E}(\mathbf{x}_m^{k+1}) + \mathbf{E}(\mathbf{x}_{m+1}^{k+1})}{2}, \tag{35}$$

and

$$\mathbf{B} := \mathbf{B}(\mathbf{x}_{m+1}^{k+1}), \tag{36}$$

results in

$$\mathbf{u}_{m+1}^{k+1} = \mathbf{u}_m^{k+1} + \Delta\tau_m E + \frac{\Delta\tau_m}{2} \frac{\mathbf{u}_{m+1}^{k+1}}{c\gamma(\mathbf{u}_{m+1}^{k+1})} \times \mathbf{B} + \mathbf{c}_m^k. \tag{37}$$

In the relativistic Boris algorithm, the Lorentz factor at the end of the time step must be estimated. The best way to estimate $\gamma(\mathbf{u}_{m+1})$ in the classical Boris schemes is still unclear [33,34]. Typically, γ is evaluated using the velocity after it has undergone half of the electric acceleration, that is $\gamma := \gamma(\mathbf{u}^-)$. This constant γ is then included in the parameter β in Algorithm 1. Since $\mathbf{u}^-$ is not an approximation of the velocity at time τ_{m+1}, when using this strategy in Boris-SDC, it will naturally not converge to $\mathbf{u}_m^{k+1}$ as k increases. Therefore, we would have

$$\left\| \frac{\mathbf{u}_{m+1}^{k+1}}{c\gamma(\mathbf{u}^-)} - \frac{\mathbf{u}_{m+1}^{k+1}}{c\gamma(\mathbf{u}_{m+1}^{k+1})} \right\| \geq \delta > 0 \tag{38}$$

for some δ that depends on the time step size and how relativistic the problem is but is independent of k. This prevents the SDC iteration from converging to an accuracy smaller than δ because eventually, as k increases, the constant error in the approximation of the relativistic factor will become dominant.

To prevent this and ensure convergence to the collocation solution we instead use $\gamma(\mathbf{u}_{m+1}^{k+1}) \approx \gamma(\mathbf{u}_{m+1}^k)$ to approximate γ. This ensures that

$$\left\| \gamma(\mathbf{u}_{m+1}^{k+1}) - \gamma(\mathbf{u}_{m+1}^k) \right\| \to 0 \quad \text{as} \quad \left\| \mathbf{u}_{m+1}^k - \mathbf{u}_{m+1}^{k+1} \right\| \to 0 \tag{39}$$

so that the relativistic factor converges to its correct value as k increases. Letting $\gamma := \gamma(\mathbf{u}_m^k)$, Eq. (37) becomes

$$\mathbf{u}_{m+1}^{k+1} = \mathbf{u}_m^{k+1} + \Delta\tau_m E + \frac{\Delta\tau_m}{2\gamma c}\mathbf{u}_{m+1}^{k+1} \times \mathbf{B} + \frac{\Delta\tau_m}{2\gamma c}\left(\mathbf{u}_m^{k+1} \times \mathbf{B} - \mathbf{u}_m^{k+1} \times \mathbf{B}\right) + \mathbf{c}_m^k. \tag{40}$$

Incorporating one term of the added zero into the constant by setting

$$\tilde{\mathbf{c}}_m^k := \mathbf{c}_m^k - \frac{\Delta\tau_m}{2\gamma c}\mathbf{u}_m^{k+1} \times \mathbf{B}, \tag{41}$$

results in

$$\mathbf{u}_{m+1}^{k+1} = \mathbf{u}_m^{k+1} + \Delta\tau_m \mathbf{E} + \frac{\Delta\tau_m}{\gamma c} \frac{\mathbf{u}_m^{k+1} + \mathbf{u}_{m+1}^{k+1}}{2} \times \mathbf{B} + \tilde{\mathbf{c}}_m^k. \tag{42}$$

Now, the update step has again the right form to be solved by Algorithm 1.

3. Numerical Results

We compare performance of Boris-SDC against the Boris integrator for two non-relativistic and two relativistic problems. The first is a two-stream instability, representing a cold plasma with an exponentially growing instability. The second is Landau damping, a hot plasma with an exponentially damped perturbation. Both these problems are electrostatic and use a one-dimensional PIC code. Third, we compute a single relativistic particle in a Penning trap using an implementation of Boris-SDC in the Runko PIC software [35]. The final experiment concerns a single relativistic particle in the special case where the magnetic and electric force exactly cancel out, where we compare the numerical drift for Boris-SDC against standard methods.

For the work-precision studies shown below, we compute the error between a given simulation and a reference simulation as the relative difference in the the norm of the electric fields

$$\mathrm{Err}^{\mathrm{rel}} = \frac{\left| \|E_{\mathrm{ref}}\|_{l_2} - \|E\|_{l_2} \right|}{\|E_{ref}\|_{l_2}} \tag{43}$$

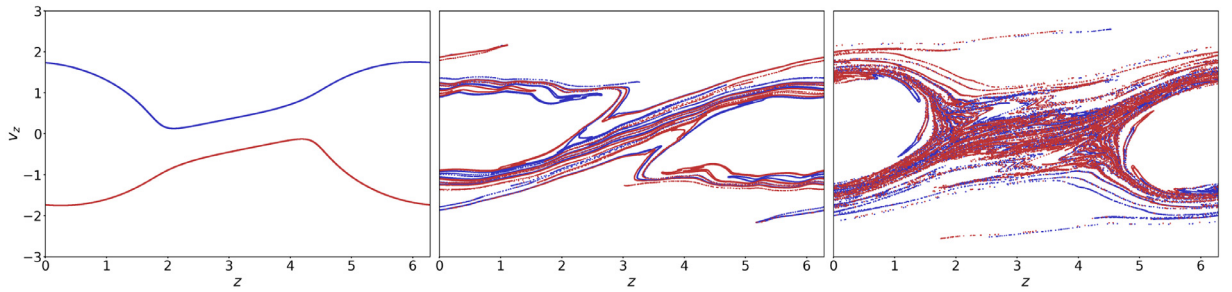

Fig. 2. Two-Stream Instability particle position-velocity phase-space at $t = 60, 180, 300$ for a half period sinusoidal perturbation in the charge density of each beam (magnitude $A = 10^{-1}$).

where

$$||E||_{l_2} = \sqrt{\Delta z \sum_{i=1}^{N_i} E_i^2}, \tag{44}$$

is used to calculate the norm. Here, Δz is the grid spacing, N_i is the number of cell nodes and E_i the electric field at node i. Note that we compute the relative error of the l_2-norms of the electric fields and not the relative error in the electric field directly. In additional experiments not documented here, we found that the gap in accuracy between Boris and Boris-SDC is slightly smaller when comparing the l_2-norms of E-fields than when computing the l2-norm of the difference of the (interpolated) E-fields. Even though the differences are minimal, we use (43) as it is slightly less favorable for Boris-SDC. To measure computational work, we count the number of evaluations of the right hand side function f that each method needs over the whole simulation. In experiments not documented here, we verified that the number of right hand side evaluations is proportional to wall clock time.

For all four numerical experiments, the equations are solved exactly as given in the previous section. The equations are nominally the SI versions for the electrostatic, non-relativistic case and CGS versions for the relativistic study of Boris-SDC. In both cases however, physical constants have been set to simple integer values to make the setup, results and quantities involved more tractable. The used values are shown in tables for each of the four problems. Such arbitrary units are common in purely computational studies [26, Chapter 3, ES 1].

3.1. Two-Stream Instability

The two-stream instability is a type of streaming instability. A beam containing one species of charged particles streams through another. While the instability can occur for counter-streaming beams of any particle mass and charge, for the purposes of this study we focus on beams consisting of the same species with particles of identical charge and mass. The beams are treated as mobile electrons moving around large immobile ions, as seen in a true quasi-neutral plasma. The immobile ions are emulated via a neutralising charge of opposite sign to the beam species added directly to the grid nodes. Such counter-streaming beams are inherently unstable as any perturbation in density or velocity distribution is reinforced by the charge induced in the other beam and vice-versa [1]. The dynamics become increasingly chaotic as the instability develops and the plasma becomes increasingly thermalised until a phase-space structure resembling an eye appears, see Figure 2.

We use the same setup as the ES1 example in the book by Langdon and Birdsall [26, Chapter 5, ES 1] and start with a sinusoidally perturbed particle density distribution

$$n(x) = n_0 + A\cos\left(\frac{2\pi kx}{L}\right), \tag{45}$$

where n_0 is the uniform distribution, A is the initial perturbation magnitude, k is the perturbation mode number and L is the domain length. Assuming that the initial magnitude A of the is small, $A \ll 1$, the early dynamics are linear with the electric field strength growing exponentially at a rate γ. This rate can be calculated analytically. The dispersion relation of the electrostatic wave induced by the perturbation has four roots [26, p. 94] and the growth rate of the instability corresponds to the imaginary root

$$\gamma_{\text{theory}} = \left[k^2 v_0^2 + \omega_p^2 - \omega_p (4k^2 v_0^2 + \omega_p^2)^{1/2}\right]^{1/2}. \tag{46}$$

Here, k is the wave number of the perturbation, v_0 is the initial velocity magnitude of the beams and ω_p is the plasma frequency

$$\omega_p = \sqrt{\frac{nq^2}{m\epsilon}}, \tag{47}$$

Table 1
Parameter used for work-precision study.

Parameter	Key	Values
Integrator	-	[Boris-SDC ($M = 3, K = 2$), Boris]
Time steps	N_t	[10, 20, 40, 50, 80, 100, 200, 400, 500, 1000]
Mesh resolution	N_z	[10, 100, 1000]
Particle count	N_q	$2 \cdot 10^5$
End time	T_E	10

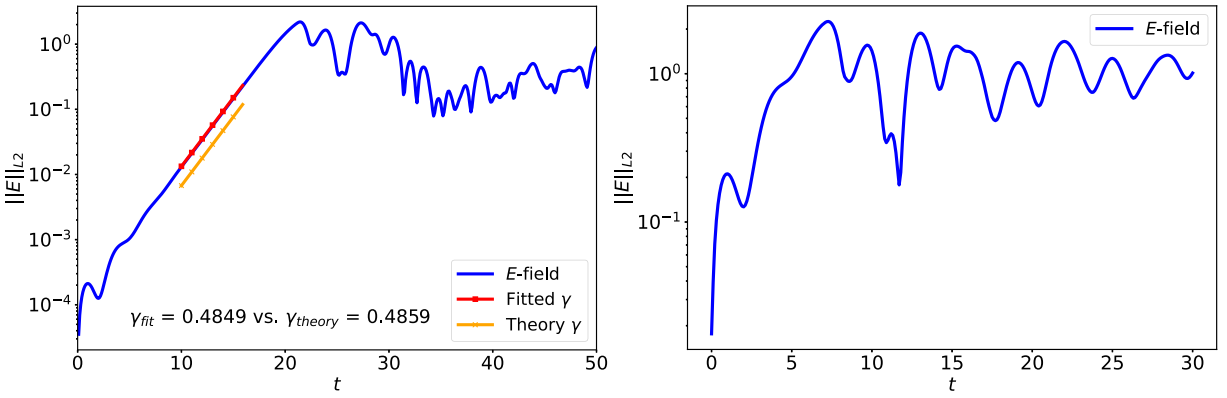

Fig. 3. E-field l_2 norm growth of weak (left) and strong (right) two-stream instability.

using the plasma density n, particle charge q and mass m as well as the permittivity ϵ.

To show that Boris-SDC captures the early dynamics correctly, we simulate the initial growth of the electric field for a weak ($A = 10^{-4}$) and strong ($A = 10^{-1}$) perturbation. In both cases we expect the field to grow exponentially with rate γ, up to a saturation point, followed by transition to a chaotic, nonlinear regime. The simulation uses $N_q = 10^4$ particles, $N_z = 100$ grid nodes and a time-step size $\Delta t = 0.1$. A periodic domain of length $L = 4\pi$ is used, a perturbation with wave number $k = 1$, beam velocity is set to $|v_0| = 1$, the particle/mass ratio to $q/m = 1$, and the permittivity to $\epsilon = 1$. Particle charge was calculated so that $\omega_p = 1$ using a plasma density defined by $n = N_q/L$.

Figure 3 shows the electric field strength given by (44) over time for the weak (left) and strong perturbation (right). For the weak perturbation case, linear growth can be seen up to a time of around $t = 20s$. Between $t = 10s$ and $t = 16s$, a line with slope γ_{theory} is shown. A best fit of the electric field data between 12s and 18s gives a growth rate of $\gamma_{\text{fit}} = 0.4849$ which matches the rate predicted by theory ($\gamma_{\text{lit}} = 0.4859$) to within 0.19%. A simulation using classic Boris produces $\gamma = 0.4855$.

To investigate performance we compare the computational effort in terms of right hand side evaluations required by Boris and Boris-SDC to reach a certain error. To minimise noise so that we can clearly assess the error from numerical discretisation, we use $N_q = 2 \times 10^5$ particles in all simulations. Meshes with $N_z = 10, N_z = 100$ and $N_z = 1000$ points were used to analyse the interplay of spatial and temporal discretisation errors. Simulation parameters are summarised in Table 1. The reference solution for both cases is a high accuracy Boris-SDC ($M = 3, K = 3$) simulation using 5 times the maximum time- and space resolution, thus $N_t = 5000$ and $N_z = 5000$. Boris-SDC uses $M = 3$ Gauss-Lobatto nodes so that the underlying quadrature is fourth order accurate.

Figure 4 shows the relative error at 1s (left) and 10s (right) versus time-step size for the strongly perturbed two-stream instability. At 1s simulation time, both Boris-SDC with $K = 1$ iteration and standard Boris are second order accurate, but Boris-SDC has a slightly smaller error constant. For $K = 2$ and $K = 3$ iterations, Boris-SDC is fourth order accurate. Here, the limiting factor is the order $p = 2M - 2 = 4$ of the underlying quadrature rule. By contrast, at 10s simulation time, the dynamics are strongly nonlinear and convergence orders are less clear. Accuracy is mostly determined by the spatial error: for $N_z = 10$ and $N_z = 100$ mesh points, there is little impact from the time step size for both Boris and Boris-SDC anymore.

Figure 5 shows error versus computational work, measured by the number of required right hand side evaluations, for the weakly perturbed two-stream instability at 1s (left) and 10s (right). For errors above 1%, the Boris integrator is the more efficient choice as Boris-SDC will require more computational work. If errors of 1% or below are required, Boris-SDC becomes more efficient. To achieve an error of, say, 10^{-4} at 10s simulation time, Boris-SDC requires about 200 evaluations of the right hand side whereas Boris requires around 800. Because spatial resolution remains fixed, eventually there are no more gains for both methods from decreasing Δt further, as the error becomes dominated by the contributions from spatial errors.

Figure 6 shows error at 1s (left) and 10s (right) versus computational effort for the strongly perturbed case. Performance at the beginning of the simulation is clearly equivalent to the same time in the weakly perturbed case. At 10 seconds, the

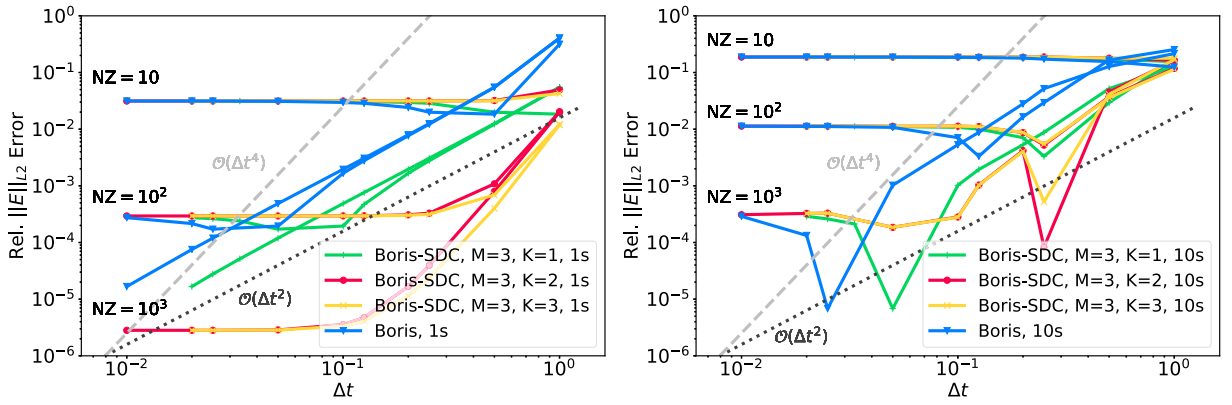

Fig. 4. Accuracy comparison of Boris and Boris-SDC for the strongly perturbed two-stream instability at 1 and 10s simulation time across 3 Boris-SDC iteration counts. Dashed/dotted lines are guide lines for fourth and second order convergence respectively.

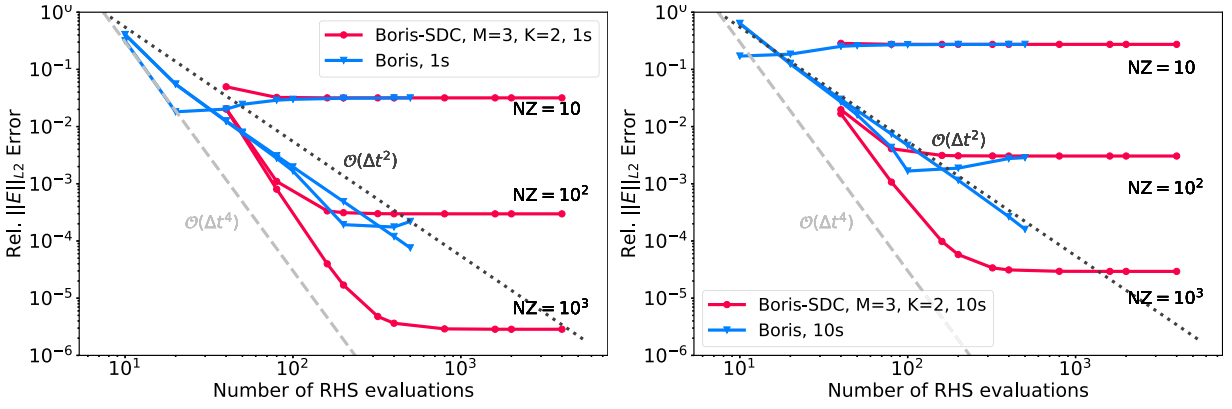

Fig. 5. Performance comparison of Boris and Boris-SDC for the weakly perturbed two-stream instability at 1 and 10s simulation time. Dashed/dotted lines are guide lines for fourth and second order convergence respectively.

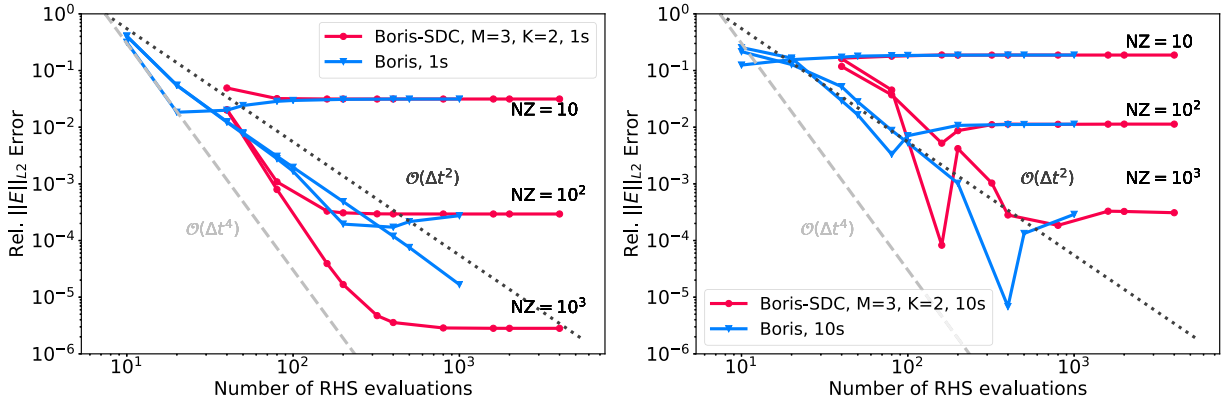

Fig. 6. Performance comparison of Boris and Boris-SDC for the strongly perturbed two-stream instability at 1 and 10s simulation time. Dashed/dotted lines are guide lines for fourth and second order convergence respectively.

two integrators deliver roughly comparable performance. The higher order of Boris-SDC allows one to take fewer, larger time steps but this gain is off-set by the increased per-time step cost of Boris-SDC.

3.2. Landau Damping

Landau damping refers to the attenuation of electrostatic waves in a collisionless plasma from energy transfer between particles and the electric field. A physical description of the phenomenon is given by Chen [1] while a detailed mathematical

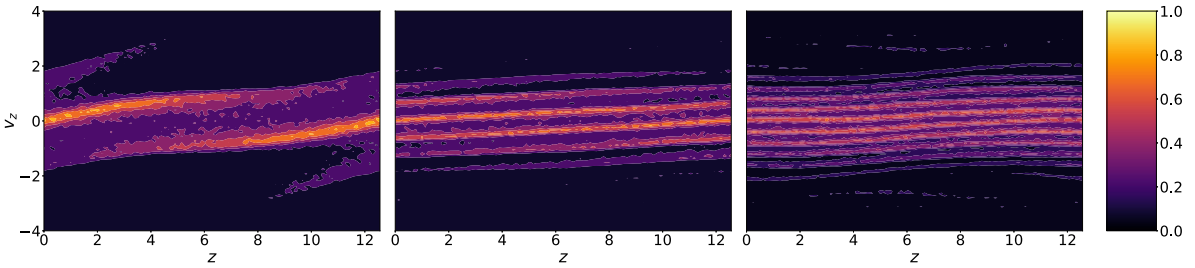

Fig. 7. Landau damping density distribution function at $t = 60, 180, 300$ for a single period sinusoidal perturbation in the charge density of each beam (magnitude $A = 0.5$).

analysis can be found in the original study by Landau [36]. Figure 7 shows the evolution of phase space density for a strongly perturbed density wave in a negatively charged particle distribution on a neutralising background. The plasma slowly returns towards an equilibrium state due to Landau damping.

As for the two-stream instability we first demonstrate that the code captures correctly the evolution of the electric field strength in both weakly and strongly perturbed simulations. Similar to the two-stream instability, the magnitude of a an imposed sinusoidal density perturbation will determine the linearity of the ensuing wave damping. Both linear and non-linear Landau damping has been studied extensively [37–41] for initial density distributions n, f in position x and velocity v space of the form

$$n(x,0) = \rho(x,0) = 1 + A\cos(kx), \tag{48}$$

and

$$f(v,0) = \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}}, \tag{49}$$

with magnitudes $A = 0.01$ and $A = 0.5$ for the weakly and strongly perturbed regimes respectively used in all studies. Note that alternatively the velocity could be perturbed as $v = v(x)$ instead of the position.

The studies mentioned above apply numerical methods directly to the Maxwell-Vlasov system, which allows direct use of (48) and (49) as initial conditions. To realise the same setup in an ESPIC code requires a particle distribution that corresponds to the initial density distribution. The average unit density of particles n_0 can be calculated from (48) using

$$n_0 = \frac{1}{L}\int^L n(x,0)\,dx = \frac{1}{L}\int^L 1 + A\cos(kx)\,dx. \tag{50}$$

Since the charge density of the particle species is given globally, the charge of the macro-particles must be assigned based on this. Unlike the two-stream instability, this problem does not use the plasma frequency as an independent quantity. The particle charge was calculated for a given simulation by dividing the global charge of the species by the desired quantity of computational particles N_q:

$$q = \frac{Q}{N_q} = \frac{\int^L \rho(x)\,dx}{N_q} = \frac{\int^L 1 + A\cos(kx)\,dx}{N_q}. \tag{51}$$

Finally, the particle velocities can be distributed randomly to fit the Maxwellian defined by (49).

A perturbation mode $k = 0.5$ and domain length $L = 4\pi$ are used and leads to a single perturbation period in space. The damping rate predicted by linear theory is $\gamma_{\text{theory}} = -0.1533$ for the weak case while numerical studies from literature [37–41] report values from $\gamma = -0.145$ to $\gamma = -0.1533$. For the strongly perturbed case, values in the literature [37,39–41] range from $\gamma = -0.280$ to $\gamma = -0.292$. We use $\gamma_{\text{lit}} = -0.1533$ for the weakly perturbed case and $\gamma_{\text{lit}} = -0.292$ for the strongly perturbed case, since both values are found by multiple and more recent studies. Rapid oscillation of the electric field is expected with an overall exponential damping of the perturbation until a saturation point is reached. In the weakly perturbed case, the plasma should continue irregular oscillation after the saturation point, while a phase of slight growth should be observed in the strongly perturbed dynamics. Simulation parameter are summarised in Table 2.

Weak Landau damping is difficult to capture with PIC unless a large number of particles is used, since the driving interaction is the energy exchange between wave and particles in the trapping range, close to the phase velocity v_ϕ. For the studied setup, the phase velocity is placed toward the tail end of the velocity distribution, meaning only a small proportion of the particles are in the trapping range. We found that using $A = 0.01$ did not induce noticeable damping and thus we increased the magnitude of the perturbation to 0.05 to place more particles in the trapping range. Each simulation used $N_q = 10^5$ particles, $N_z = 100$ grid nodes and a time-step size $\Delta t = 0.1$.

Figure 8 shows the evolution of the magnetic field for the weak (left) and strong (right) perturbation. The dynamics match the expected behaviour: for the weak perturbation, the electric field decreases exponentially until a saturation point

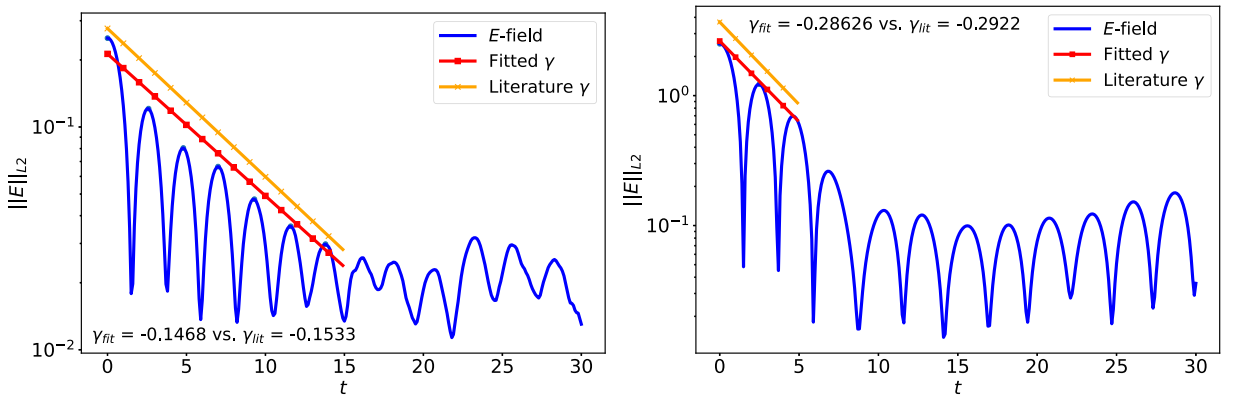

Fig. 8. E-field l_2-norm evolution for weak (left) and strong (right) Landau damping.

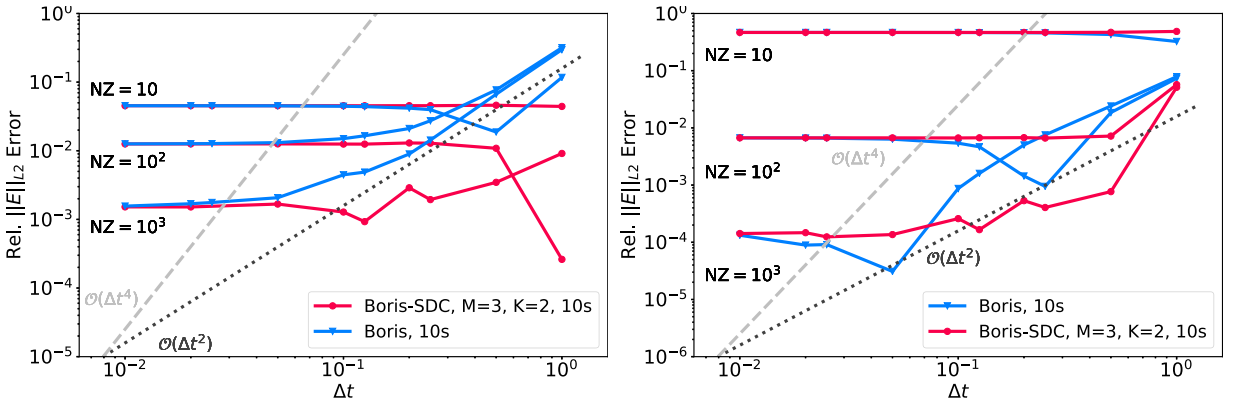

Fig. 9. Error at 10s versus time step size for weak (left) and strong (right) Landau damping.

at around $t = 15$. After that, the field continues to oscillate without a clear change in magnitude. For the strong perturbation, the damping phase is shorter and after saturation, a slight growth of the electric field sets in.

To evaluate the damping rates, best fit lines were drawn through the relevant oscillation peaks on each graph, the first seven peaks were used for the linear data ($t \sim [0, 15]$) and first three used for the non-linear data ($t \sim [0, 5]$). The relative error of the simulated linear damping rate $\gamma_{\text{fit}} = -0.147$ to the theoretical value was found to be approximately 4.1%, thought to be reasonable agreement considering the rapidly oscillating dynamics and difficulties related to the scheme. Furthermore, the simulated damping rate in the non-linear case $\gamma_{\text{fit}} = -0.28825$ was firmly within the range of values reported in literature.

Figure 9 shows the error in electric field norm compared to the reference simulation at 10s against time step size for weak (left) and strong (right) Landau damping. In most cases, the spatial error dominates and there is little effect from varying Δt. However, for $N_z = 10^3$, Boris-SDC is more accurate for the same Δt and reaches the saturation error set by the spatial resolution earlier than the Boris method. For example, Boris-SDC reaches an error of 10^{-3} for weak Landau damping with a step size of around $\Delta t = 10^{-1}$, compared to the Boris integrator which requires $\Delta t = 10^{-2}$. This effect

Table 2
Physical setup parameters for the Landau damping cases.

Parameter	Weak perturbation	Strong perturbation
L	4π	4π
k	0.5	0.5
A	0.05	0.5
v_{th}	1	1
ϵ	1	1
ω_p	1	1
n_0	1	1
q	L/N_q	L/N_q

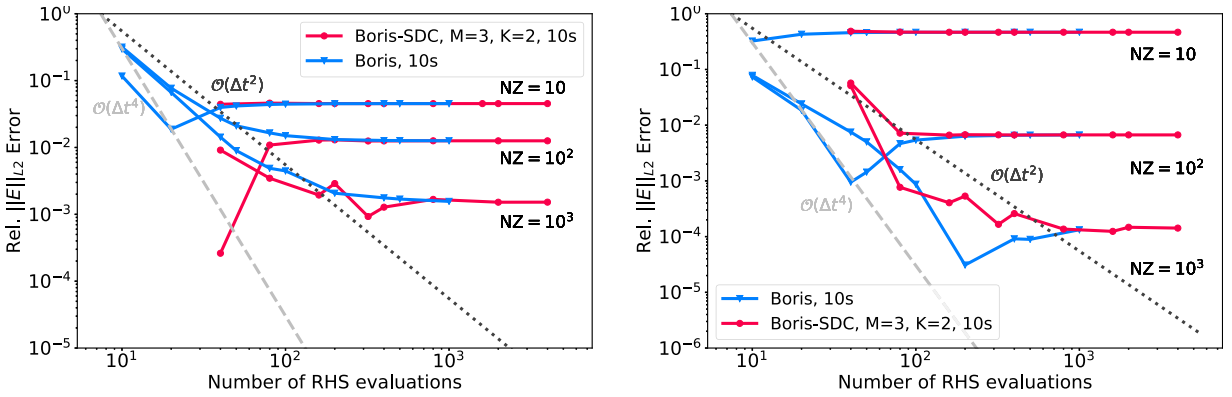

Fig. 10. Error versus number of right hand side evaluations for weak (left) and strong (right) Landau damping.

Table 3
Validation study parameters.

$\hat{c}$	0.45
t_{end}	45
$\hat{\mathbf{x}}(0)$	$(7.5, 5, 7.5)^{\mathrm{T}}$
$\hat{\mathbf{v}}(0)$	$(0.7\hat{c}, 0, 0.7\hat{c})^{\mathrm{T}}$
N_x, N_y, N_z	10
$\lvert E\rvert$	0.1
$\lvert B\rvert$	1

is more pronounced for the weak Landau damping, most likely because the smoother dynamics lead to a smaller spatial discretisation error.

Figure 10 shows error at 10s against computational effort, measured by the number of right hand side evaluations. For weak Landau damping, both methods perform similarly. Although Boris-SDC allows one to achieve a given accuracy with a larger time step size, the reduced computational effort from computing fewer time steps is counterbalanced by the increased workload per step. Only minimal gains are achieved for weak Landau damping for errors between 10^{-2} and 10^{-3} where Boris-SDC is marginally more efficient. For strong Landau damping, we do not see efficiency gains from Boris-SDC, despite its better accuracy.

3.3. Relativistic Penning Trap

We compare performance of Boris and Boris-SDC in terms of work-precision for the Penning trap, similar to the non-relativistic test cased used by Winkel et al. [12]. Here, however, we translate the parameters into the units used by Runko. The simulation parameters are summarised in Table 3. The fields at a grid node with index (i, j, k) are

$$E_{i,j,k} = |E| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{bmatrix} x_{i,j,k}, \tag{52}$$

and

$$B_{i,j,k} = |B|, \tag{53}$$

where $x_{i,j,k}$ is the position of node. The result is a homogeneous magnetic field pointing up along the z-axis and an electric field pushing towards and along the xy-plane centred on $z = 0$. Charges caught in the fields are pushed towards this plane and away from the centre by the electric field, with the magnetic field curving the trajectories back to keep the charges trapped. We use the linear interpolation to compute field values at particle positions and the Boris integrator as a reference particle pusher. The resulting trajectory and trajectory projection on the $x - y$-plane can be seen in Figs. 11 and 12.

Figs. 13 shows the SDC residual in position (left) and velocity (right) for Boris-SDC for $K = 0$ up to $K = 4$ iterations. Because of the choice of the Lorentz parameter described in Subsection 2.3, γ converges to the correct values as K increases and the residual goes down to more or less machine precision as the iteration converges to the collocation solution.

To compare Boris and Boris-SDC in terms of work-precision, we compute a reference solution using the same $10 \times 10 \times 10$ grid, $NT = 3200$ and $M = 5$ quadrature nodes. Fig. 14 shows error against time step (left) and error against computational cost (right), measured again by the number of required right hand side evaluations. As a guide to the eye, gray lines with slopes of minus one, two, four and eight are also shown.

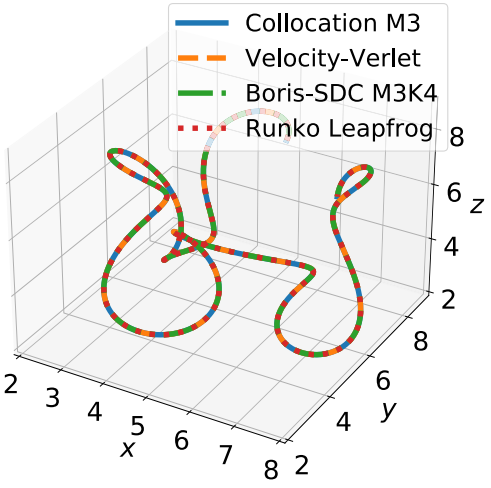

Fig. 11. Runko and Boris-SDC M3K4 Penning trajectory for $t = [0, 45]$.

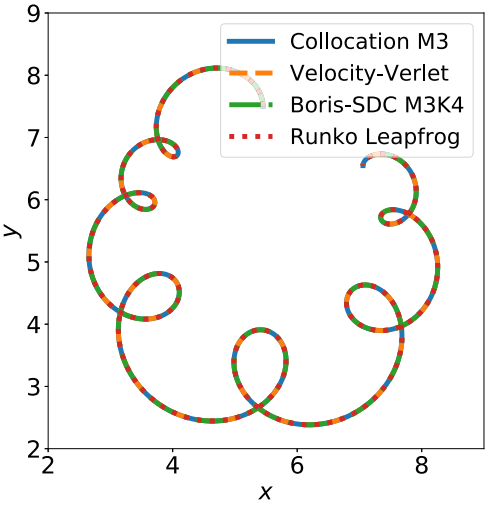

Fig. 12. Runko and Boris-SDC M3K4 Penning xy-plane trajectory for $t = [0, 45]$.

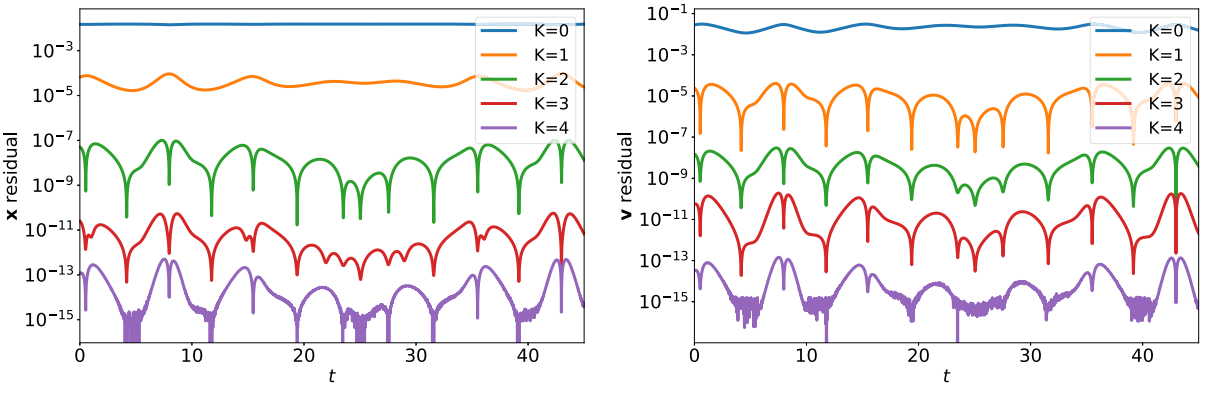

Fig. 13. Boris-SDC residual in position (left) and velocity (right) for the relativistic Penning trap.

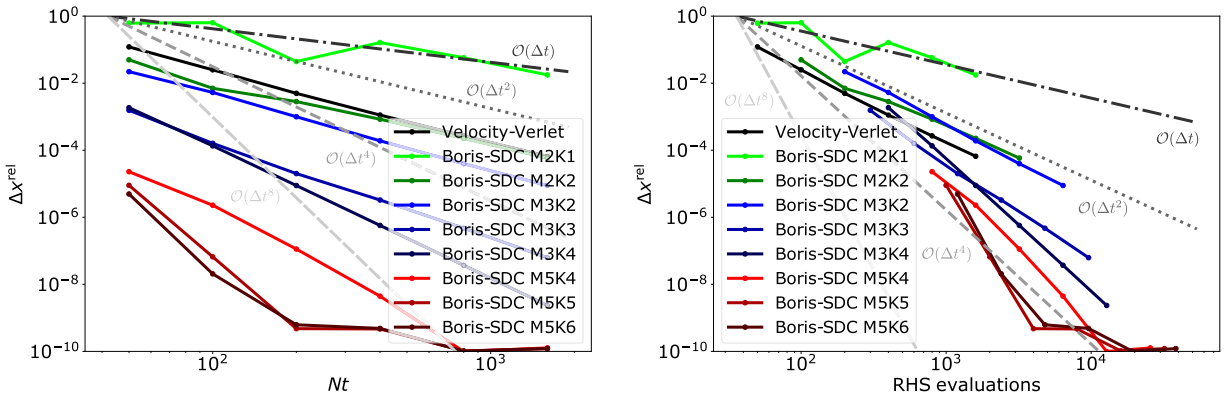

Fig. 14. Error versus number of time steps (left) and error versus computational cost (right) for the relativistic Penning trap.

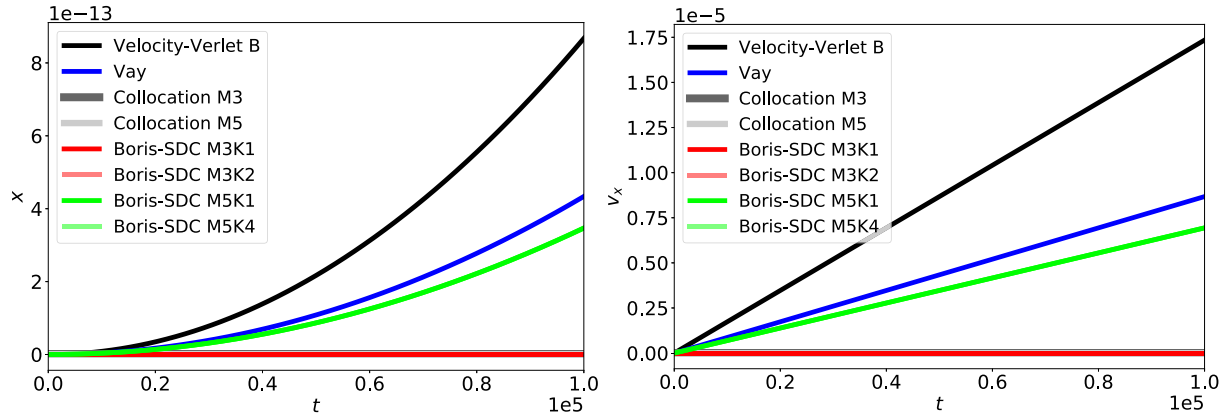

Fig. 15. Error in position (left) and velocity (right) for the force-free case. Note that the lines for Collocation M3 and M5 and Boris-SDC M3K2 and M5K4 all overlap.

As expected, Boris/velocity-Verlet converges with order two. For Boris-SDC, the order increases with K, although the precise impact of an additional iteration is not clear. While Winkel et al. [12] observed numerically for the non-relativistic case that a sweep increases order by two (until reaching the order of the collocation method), the picture is less clear in the relativistic case. For both $M = 3$ and $M = 5$ nodes, Boris-SDC reproduces the order four or eight, respectively, of the underlying collocation method for sufficiently many iterations. Furthermore, every iteration leads to a significant gain in accuracy, even when it fails to fully increase the order by two.

3.4. Numerical drift in the force-free case

If the acceleration from the magnetic and electric field in (23) cancel, there is no net force acting on the particle and it should continue to travel into its original direction without changing velocity. However, due to round-off error, acceleration from the fields will not cancel out exactly on the discrete level and the particle will undergo numerical drift. For the test case studied by Ripperda et al. [42], we will compare numerical drift for the Boris integrator, the Vay integrator [34] and Boris-SDC with $M = 3$ and $M = 5$ nodes and varying iteration counts. The test cases uses a particle with an initial velocity that is parallel to the y-axis and a magnetic field with field lines oriented along the z-axis. Then, the electric field is set to

$$\mathbf{E} = -\mathbf{v} \times \mathbf{B} = \begin{pmatrix} v_y B_z \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} E_x \\ 0 \\ 0 \end{pmatrix} \tag{54}$$

so that the net acceleration is zero. Parameters for the problem are summarised in Table 4.

Figure 15 shows the resulting drift in position (left) and velocity (right) over time for Boris, Vay and Boris-SDC with $M = 3$ nodes and $K = 1, 2$ iterations and $M = 5$ nodes and $K = 1, 4$ iterations. For both $M = 3$ and $M = 5$ nodes, the collocation solution exhibits no numerical drift. Boris-SDC with $M = 3$ nodes and $K = 1$ iteration is drift-free. By contrast, when $M = 5$ nodes are used, Boris-SDC with $K = 1$ iteration suffers from drift. However, its drift is already less than that of the Vay

Table 4
Force-free test parameters.

Parameter	Value
c	1
q/m	1
γ	10^6
B_z	1
v_y	$\sqrt{1-\frac{1}{\gamma^2}}\cdot c$
E_x	$-v_x B_z$
T_{end}	10^5

method and much less than that of classical Boris. While these results are very preliminary and a deeper analysis is needed, they at least hint at the possibility that Boris-SDC could reduce numerical drift compared to Vay and classical Boris.

4. Conclusions and Outlook

The paper explores the effect of using Boris spectral deferred corrections (Boris-SDC), a high order generalisation of the Boris algorithm, as particle pusher in a particle-in-cell code. Accuracy and performance in terms of work-precision is investigated for simulations of a two-stream instability and Landau damping. A modification for Boris-SDC is proposed to make it applicable to the relativistic Lorentz equations. Performance of relativistic Boris-SDC is then studied for a relativistic Penning trap and numerical drift is assessed for a particle in a scenario where accelerations from the electric and magnetic field should cancel out.

Throughout, we found that Boris-SDC is more accurate than Boris at the same time step size, yielding lower error for any given time-step size, up to the limits imposed by the accuracy of the spatial discretisation. Alternatively, a given accuracy could be reached with a much larger time step compared to Boris. However, for the non-relativistic cases, these gains were offset by the much higher computational cost per time step; as a result, Boris and Boris-SDC performed very similarly in terms of computational cost. A key problem was that mostly errors from the spatial discretisation were dominating, limiting the gains from the higher order of Boris-SDC. Results were somewhat more positive for the relativistic case, partly because the imposed electric and magnetic fields are relatively simple and can be interpolated exactly. We confirmed that relativistic Boris-SDC delivers higher order of convergence, up to the order of the underlying quadrature method. This also translated into computational gains compared to relativistic Boris for errors below 10^{-3}. Finally, tests for the force-free case suggest that Boris-SDC might produce less numerical drift than classical Boris or the Vay integrator. Some configurations of Boris-SDC were even completely drift free, but we do not have, at the moment, a theoretical understanding why this is.

Although the results so far do not show clear efficiency gains from Boris-SDC, we expect that PIC/Boris-SDC could outperform PIC/Boris when combined with high order spatial approximations as well as techniques to further improve performance of SDC [15,22,23] and in situations where high accuracy is required. There also has been an interest recently by method developers in developing fully implicit PIC pushers [43]. Since the collocation approach underpinning Boris-SDC corresponds to a fully implicit Runge-Kutta method, our results could also be of interest for further developments into that direction.

Potential practical applications of Boris-SDC include, for example, studies of growth rate and saturation of especially weak plasma instabilities in laboratory and astrophysical context. In this case the algorithm can rely on accurate field information and therefore vastly outperform standard Boris method by being more accurate and computationally efficient. Topics of interest for further exploration of ESPIC/Boris-SDC include relativistic test cases, longer simulations and high accuracy. Development of a fully electromagnetic Boris-SDC/PIC scheme would also be an interesting direction for future work.

Data availability

Data will be made available on request.

Acknowledgments

K. S. thanks NORDITA for the hospitality during his visit during which part of this work was initiated. K. S. was supported by the Engineering and Physical Sciences Research Council (EPSRC) Centre for Doctoral Training in Fluid Dynamics(EP/L01615X/1). S. M. T. would like to acknowledge support from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation program (grant agreement no. D5S-DLV-786780)

References

[1] F.F. Chen, Introduction to Plasma Physics, Springer, New York, 1974.

[2] J.P. Boris, Relativistic plasma simulation-optimization of a hybrid code, in: Proc. Fourth Conf. Num. Sim. Plasmas, Naval Res. Lab, Wash. DC, 1970, pp. 3–67.
[3] H. Qin, S. Zhang, J. Xiao, J. Liu, Y. Sun, W.M. Tang, Why is Boris algorithm so good? Physics of Plasmas 20 (8) (2013) 084503.
[4] E. Hairer, C. Lubich, Energy behaviour of the Boris method for charged-particle dynamics, BIT Numerical Mathematics 58 (4) (2018) 969–979.
[5] E. Hairer, C. Lubich, Symmetric multistep methods for charged-particle dynamics, SMAI J. Comput. Math. 3 (2017) 205–218.
[6] Y. He, Y. Sun, J. Liu, H. Qin, Higher order volume-preserving schemes for charged particle dynamics, Journal of Computational Physics 305 (2016) 172–184, doi:10.1016/j.jcp.2015.10.032.
[7] T. Li, B. Wang, Arbitrary-order energy-preserving methods for charged-particle dynamics, Applied Mathematics Letters 100 (2020) 106050.
[8] J. Qiang, High order numerical integrators for relativistic charged particle tracking, arXiv preprint: 1702.04486 (2017).
[9] M. Quandt, High order particle transport for PIC simulations of plasma flows, University of Stuttgart, 2010 Ph.D. thesis.
[10] M. Tao, Explicit high-order symplectic integrators for charged particles in general electromagnetic fields, Journal of Computational Physics 327 (2016) 245–251.
[11] E. Hairer, C. Lubich, G. Wanner, Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations, volume 31, Springer Science & Business Media, 2006.
[12] M. Winkel, R. Speck, D. Ruprecht, A high-order Boris integrator, Journal of Computational Physics 295 (2015) 456–474, doi:10.1016/j.jcp.2015.04.022.
[13] A. Dutt, L. Greengard, V. Rokhlin, Spectral deferred correction methods for ordinary differential equations, BIT Numerical Mathematics 40 (2) (2000) 241–266, doi:10.1023/A:1022338906936.
[14] K. Tretiak, D. Ruprecht, An arbitrary order time-stepping algorithm for tracking particles in inhomogeneous magnetic fields, Journal of Computational Physics: X 4 (2019) 100036, doi:10.1016/j.jcpx.2019.100036.
[15] J. Huang, J. Jia, M. Minion, Accelerating the convergence of spectral deferred correction methods, Journal of Computational Physics 214 (2) (2006) 633–656, doi:10.1016/j.jcp.2005.10.004.
[16] K. Tretiak, J. Buchanan, R. Akers, D. Ruprecht, Performance of the BGSDC integrator for computing fast ion trajectories in nuclear fusion reactors, Computer Physics Communications 264 (2021) 107876, doi:10.1016/j.cpc.2021.107876.
[17] R.W. Hockney, J.W. Eastwood, Computer Simulation Using Particles, CRC Press, 1988.
[18] A. Pukhov, Particle-in-cell codes for plasma-based particle acceleration (2015). arXiv preprint arXiv: 1510.01071.
[19] J. Xiao, H. Qin, J. Liu, Structure-preserving geometric particle-in-cell methods for Vlasov-Maxwell systems, Plasma Science and Technology 20 (11) (2018) 110501, doi:10.1088/2058-6272/aac3d1.
[20] M. Shalaby, A.E. Broderick, P. Chang, C. Pfrommer, A. Lamberts, E. Puchwein, Sharp: A spatially higher-order, relativistic particle-in-cell code, The Astrophysical Journal 841 (1) (2017) 52.
[21] I.V. Sokolov, Alternating-order interpolation in a charge-conserving scheme for particle-in-cell simulations, Computer Physics Communications 184 (2) (2013) 320–328, doi:10.1016/j.cpc.2012.09.015.
[22] M. Weiser, Faster SDC convergence on non-equidistant grids by DIRK sweeps, BIT Numerical Mathematics 55 (4) (2014) 1219–1241, doi:10.1007/s10543-014-0540-y.
[23] M. Emmett, M.L. Minion, Toward an efficient parallel in time method for partial differential equations, Communications in Applied Mathematics and Computational Science 7 (2012) 105–132, doi:10.2140/camcos.2012.7.105.
[24] B.W. Ong, R.D. Haynes, K. Ladd, Algorithm 965: RIDC methods: A family of parallel time integrators, ACM Trans. Math. Softw. 43 (1) (2016) 8:1–8:13, doi:10.1145/2964377.
[25] J.P. Verboncoeur, Particle simulation of plasmas: review and advances, Plasma Physics and Controlled Fusion 47 (5A) (2005) A231.
[26] C.K. Birdsall, B.A. Langdon, Plasma Physics Via Computer Simulation, McGraw-Hill, Inc., New York, NY, 1985.
[27] C.-D. Munz, P. Omnes, R. Schneider, E. Sonnendrücker, U. Voß, Divergence correction techniques for Maxwell solvers based on a hyperbolic model, Journal of Computational Physics 161 (2) (2000) 484–511, doi:10.1006/jcph.2000.6507.
[28] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental algorithms for scientific computing in Python, Nature Methods 17 (2020) 261–272, doi:10.1038/s41592-019-0686-2.
[29] A.K. Mazur, Common molecular dynamics algorithms revisited: Accuracy and optimal time steps of Störmer-leapfrog integrators, Journal of Computational Physics 136 (2) (1997) 354–365, doi:10.1006/jcph.1997.5740.
[30] S. Zenitani, T. Umeda, On the boris solver in particle-in-cell simulation, Physics of Plasmas 25 (11) (2018) 112110, doi:10.1063/1.5051077.
[31] D.J. Griffiths, Introduction to Electrodynamics, 4th, Cambridge University Press, 2017.
[32] E. Hairer, C. Lubich, G. Wanner, Geometric numerical integration illustrated by the Störmer-Verlet method, Acta Numerica 12 (2003) 399-450, doi:10.1017/S0962492902000144.
[33] A.V. Higuera, J.R. Cary, Structure-preserving second-order integration of relativistic charged particle trajectories in electromagnetic fields, Physics of Plasmas 24 (5) (2017) 052104.
[34] J.-L. Vay, Simulation of beams or plasmas crossing at relativistic velocity, Physics of Plasmas 15 (5) (2008) 056701.
[35] J. Nättilä, Runko: Modern multi-physics toolbox for simulating plasma (2019). arXiv preprint arXiv: 1906.06306.
[36] L.D. Landau, On the vibrations of the electronic plasma, J. Phys.(USSR) 10 (1946) 25–34. [Zh. Eksp. Teor. Fiz.16,574(1946)]
[37] B. Ayuso de Dios, S. Hajian, High order and energy preserving discontinuous Galerkin methods for the Vlasov-Poisson system (2012). arXiv preprint arXiv: 1209.4025.
[38] J. Canosa, J. Gazdag, J. Fromm, B. Armstrong, Electrostatic oscillations in plasmas with cutoff distributions, Physics of Fluids 15 (12) (1972) 2299–2305.
[39] C.-Z. Cheng, G. Knorr, The integration of the Vlasov equation in configuration space, Journal of Computational Physics 22 (3) (1976) 330–351.
[40] T. Nakamura, T. Yabe, Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov-Poisson equation in phase space, Computer Physics Communications 120 (2-3) (1999) 122–154.
[41] J.A. Rossmanith, D.C. Seal, A positivity-preserving high-order semi-Lagrangian discontinuous Galerkin scheme for the Vlasov-Poisson equations, Journal of Computational Physics 230 (16) (2011) 6203–6232.
[42] B. Ripperda, F. Bacchini, J. Teunissen, C. Xia, O. Porth, L. Sironi, G. Lapenta, R. Keppens, A comprehensive comparison of relativistic particle integrators, The Astrophysical Journal Supplement Series 235 (1) (2018) 21.
[43] F. Bacchini, J. Amaya, G. Lapenta, The relativistic implicit particle-in-cell method, Journal of Physics: Conference Series 1225 (1) (2019) 012011, doi:10.1088/1742-6596/1225/1/012011.