Electromagnetic Theory Computation Project 1

An Implementation of Relativistic Boris Method

Chang, Heng-Wei

hengweichang@arizona.edu

Department of Physics, The University of Arizona

October 2, 2023

Abstract

We firstly discuss the advantage of the Boris's numerical method for Lorentz force: kinetic energy conservation with only magnetic field appearing. The analytic solution of a charge in a uniform magnetic field is used to test the convergence and accuracy of the implement. After recreating a certain result in Zenitani et al. (2018), we solve for the trajectory of a charge in a magnetic field whose direction is fixed but with gradient of its magnitude perpendicular to itself.

1 Introduction

The theoretical importance and practical use of the Lorentz force cannot be emphasized enough, although the equation itself is as simple as it could be. As we write down the non-gauge-invariant Lagrangian of a charge interacting with the four vector potential from the Lorentz force, it implicitly implies an underlying complicated nature of the gauge theory and we consider non-Abelian gauge redundancy and try to quantize the theory via Fadeev-Popov ghosts. From plasma physics to baryon oscillation, the Lorentz force has been and will still be widely used to discuss a great variety of problems and phenomena.

The necessity of an accurate numerical method to calculate the motion of charged particle consequently appears. However, common methods for second order differential equation do not work impressively on this problem, as the kinetic energy of the charge usually increases with the number of iterations. In face of this problem, the Boris method, published by J.P. Boris (1970), is invented to provide a numerical method to minimize the numerical kinetic energy increase.

In the first section of this report, we show explicitly the kinetic energy increase in the explicit Euclid forward method and how the Boris' method resolves this issue by making the magnitude of the velocity conserved if there is only the presence of the megnetic field. In the beginning of the second section, we test our program by solving the trajectory of a charge particle in a uniform magnetic field and compare the numerical solution to the well-known analytic solution, with L^2 errors calculated under different grid sizes. Specifically, we find that the L^2 error decreases linearly with the grid size, and the time cost increases linearly as we expect. Also, we numerically solve the trajectory with a uniform magnetic field stronger than a uniform electric field and vice verse. As there is a frame where one of them vanishes, we can still predict the trajectories.

In the remaining parts of the second section, we firstly recreate the Fig.3 in Zenitani et al. (2018), which is about the motion of a charge particle in an axially electric field and an axially symmetric magnetic field. As the accuracy of the program is assured, we move on to numerically solve the motion of a charge particle under a magnetic field whose direction is fixed, but the gradient of the magnitude is perpendicular to the magnetic field itself. Since the angular frequency of the trajectory of charge particle is larger on one side, the charge will gradually move in the direction perpendicular both to the magnetic field and the gradient of its magnitude. This phenomenon is conceptually explained in Classical Electrodynamics (Ed. 3) by Jackson, and we numerically verify the inspiring motion.

The time complexity of Boris method is the same with those of explicit forward methods as only simple intermediate steps are used to fixed the increasing energy problem. However, it greatly increases the accuracy and practicality of numerical methods to electrodynamical problems.

2 Theory and Numerical Methods

The relativistic Lorentz force equation is given by

$$\frac{d}{dt}\Big(\gamma(v)m\vec{v}\Big) = \frac{q}{c}\Big(\vec{E} + \vec{v}\times\vec{B}\Big), \tag{1}$$

which is not complicated itself as a first order ordinary differential equation. However, its analytical solutions are difficult to find if we have varying and non-uniform electric and magnetic fields. As we implement the explicit forward method to calculate the motion of a charge particle under electromagnetic field with known trajectory, one usually find that the numerical solution will deviate a lot from the solution in the long run.

The main reason is the kinetic energy of the increase exponentially with the number of iteration. We will firstly show this anomalous numerical defect and that the Boris method does not suffer to this.

2.1 The Problematic Explicit Forward in Magnetic Field

If we want to solve for the trajectory of a charge in a magnetic field numerically, one may implement the following recursive Euclid explicit forward method

$$\vec{v}_{n+1} = \vec{v}_n + \frac{qdt}{mc}\vec{v}_n \times \vec{B}(x_n). \tag{2}$$

However, this could not work since v^2 is not conserved as

$$\begin{aligned} v_{n+1}^2 &= v_n^2 + \Big(\frac{qdt}{mc}\Big)^2 |\vec{v}_n \times \vec{B}(x_n)|^2 \\ &= v_n^2 + \Big(\frac{qdt}{mc}\Big)^2 \Big(v_n^2 B(x_n)^2 - (\vec{v}_n \cdot \vec{B}(x_n))^2\Big) \\ &= v_n^2 \Big(1 + \Big(\frac{qdt}{mc}\Big)^2 \Big(B(x_n)^2 - (\hat{v}_n \cdot \vec{B}(x_n))^2\Big)\Big), \end{aligned} \tag{3}$$

which implies v^2 is monotonically increasing, as $|\vec{v}/c|$ is smaller than one for massive particles. In the special case $\vec{B} = B_0\hat{z}$ with the velocity on the $x-y$ plane, we have

$$v(t+\Delta t)^2 = v(t)^2 \Big(1 + \frac{q^2 B_0^2 dt^2}{m^2}\Big)^N, \tag{4}$$

which increases exponentially with the number of steps. But we expect the kinetic energy to be conserved since the right hand side of the Lorentz force equation is perpendicular to the velocity, so the inner products of both sides with $\gamma(v)\vec{v}$ will give

$$\frac{d}{dt}\Big(\frac{v^2}{1 - v^2/c^2}\Big) = 0. \tag{5}$$

But $x/(1-x)$ is an increasing function on $[0, 1)$, v^2 must consequently be a constant.

2.2 Boris' Method

To deal with the problem, we alternatively consider the implicit forward recursion

$$\vec{v}_{n+\frac{1}{2}} - \vec{v}_{n-\frac{1}{2}} = \frac{qdt}{2m}\Big(\vec{v}_{n+\frac{1}{2}} + \vec{v}_{n-\frac{1}{2}}\Big) \times \vec{B}(x_n), \tag{6}$$

equivalently,

$$\vec{v}_{n+\frac{1}{2}} - \vec{v}_{n+\frac{1}{2}} \times \vec{t}_n = \vec{v}_{n-\frac{1}{2}} + \vec{v}_{n-\frac{1}{2}} \times \vec{t}_n, \tag{7}$$

where

$$\vec{t}_n = \frac{qdt}{2m}\vec{B}(x_n). \tag{8}$$

To solve this problem, we can define

$$\vec{w}_n := \vec{v}_{n-\frac{1}{2}} + \vec{v}_{n-\frac{1}{2}} \times \vec{t}_n, \tag{9}$$

then we have the recursive equation

$$\begin{aligned} \vec{v}_{n+\frac{1}{2}} &= \vec{w}_n + \vec{v}_{n+\frac{1}{2}} \times \vec{t}_n \\ &= \vec{w}_n + \Big(\vec{w}_n + \vec{v}_{n+\frac{1}{2}} \times \vec{t}_n\Big) \times \vec{t}_n. \end{aligned} \tag{10}$$

The explicit forward method is equivalent the use of this recursive relation and drop the $O(t^2)$ terms. But we shall take a closer look at this equation since it implies then we have the recursive equation

$$\vec{v}_{n+\frac{1}{2}} = \vec{w}_n + \vec{w}_n \times \vec{t}_n - t_n^2 \vec{v}_{n+\frac{1}{2}} + \Big(\vec{t}_n \cdot \vec{v}_{n+\frac{1}{2}}\Big)\vec{t}_n. \tag{11}$$

But (7) implies

$$\vec{t}_n \cdot \vec{v}_{n+\frac{1}{2}} = \vec{t}_n \cdot \vec{v}_{n-\frac{1}{2}}. \tag{12}$$

Hence, (11) can be re-written as

$$\begin{aligned}
(1+t_n^2)\vec{v}_{n+\frac{1}{2}} &= \vec{w}_n + \vec{w}_n \times \vec{t}_n + \left(\vec{t}_n \cdot \vec{v}_{n-\frac{1}{2}}\right)\vec{t}_n \\
&= \vec{w}_n + \vec{w}_n \times \vec{t}_n + \left(\vec{v}_{n-\frac{1}{2}} \times \vec{t}_n\right) \times \vec{t}_n + t_n^2 \vec{v}_{n-\frac{1}{2}} \\
&= \vec{w}_n + \vec{w}_n \times \vec{t}_n + \left(\vec{w}_n - \vec{v}_{n-\frac{1}{2}}\right) \times \vec{t}_n + t_n^2 \vec{v}_{n-\frac{1}{2}} \\
&= \vec{w}_n - \vec{v}_{n-\frac{1}{2}} \times \vec{t}_n + t_n^2 \vec{v}_{n-\frac{1}{2}} + 2\vec{w}_n \times \vec{t}_n \\
&= (1+t_n^2)\vec{v}_{n-\frac{1}{2}} + 2\vec{w}_n \times \vec{t}_n.
\end{aligned} \tag{13}$$

Consequently, we can numerically solve the velocity by forwarding

$$\begin{aligned}
\vec{w}_n &= \vec{v}_{n-\frac{1}{2}} + \vec{v}_{n-\frac{1}{2}} \times \vec{t}_n \\
\vec{v}_{n+\frac{1}{2}} &= \vec{v}_{n-\frac{1}{2}} + \frac{2}{1+t_n^2}\vec{w}_n \times \vec{t}_n.
\end{aligned} \tag{14}$$

However, the same problem in the explicit forward method arises: whether v^2 is conserved in this method? In order to know the answer, we have to calculate its change.

$$\begin{aligned}
v_{n+\frac{1}{2}}^2 - v_{n-\frac{1}{2}}^2 &= \left(\vec{v}_{n-\frac{1}{2}} + \frac{2}{1+t_n^2}\vec{w}_n \times \vec{t}_n\right)^2 - v_{n-\frac{1}{2}}^2 \\
&= \frac{4}{1+t_n^2}\vec{v}_{n-\frac{1}{2}} \cdot \left(\vec{w}_n \times \vec{t}_n\right) + \frac{4}{(1+t_n^2)^2}\left(\vec{w}_n \times \vec{t}_n\right)^2 \\
&= \frac{4}{1+t_n^2}\left(\vec{v}_{n-\frac{1}{2}} \times \vec{w}_n\right) \cdot \vec{t}_n + \frac{4}{(1+t_n^2)^2}\left(\vec{w}_n \times \vec{t}_n\right)^2 \\
&= \frac{4}{1+t_n^2}\left(\vec{v}_{n-\frac{1}{2}} \times (\vec{v}_{n-\frac{1}{2}} \times \vec{t}_n)\right) \cdot \vec{t}_n + \frac{4}{(1+t_n^2)^2}\left(\vec{w}_n \times \vec{t}_n\right)^2 \\
&= \frac{4}{1+t_n^2}\left(\vec{t}_n \times \vec{v}_{n-\frac{1}{2}}\right) \cdot \left(\vec{v}_{n-\frac{1}{2}} \times \vec{t}_n\right) + \frac{4}{(1+t_n^2)^2}\left(\vec{w}_n \times \vec{t}_n\right)^2 \\
&= -\frac{4}{1+t_n^2}\left(\vec{t}_n \times \vec{v}_{n-\frac{1}{2}}\right)^2 + \frac{4}{(1+t_n^2)^2}\left(\vec{w}_n \times \vec{t}_n\right)^2 \\
&= -\frac{4}{1+t_n^2}\left(\vec{t}_n \times \vec{v}_{n-\frac{1}{2}}\right)^2 + \frac{4}{(1+t_n^2)^2}\left(\vec{v}_{n-\frac{1}{2}} \times \vec{t}_n + (\vec{v}_{n-\frac{1}{2}} \times \vec{t}_n) \times \vec{t}_n\right)^2 \\
&= -\frac{4}{1+t_n^2}\left(\vec{t}_n \times \vec{v}_{n-\frac{1}{2}}\right)^2 + \frac{4}{(1+t_n^2)^2}\left((\vec{v}_{n-\frac{1}{2}} \times \vec{t}_n)^2 + ((\vec{v}_{n-\frac{1}{2}} \times \vec{t}_n) \times \vec{t}_n)^2\right) \\
&= -\frac{4t_n^2}{(1+t_n^2)^2}\left(\vec{t}_n \times \vec{v}_{n-\frac{1}{2}}\right)^2 + \frac{4}{(1+t_n^2)^2}\left((\vec{v}_{n-\frac{1}{2}} \times \vec{t}_n)^2 t_n^2 - ((\vec{v}_{n-\frac{1}{2}} \times \vec{t}_n) \cdot t_n)^2\right) \\
&= -\frac{4t_n^2}{(1+t_n^2)^2}\left(\vec{t}_n \times \vec{v}_{n-\frac{1}{2}}\right)^2 + \frac{4t_n^2}{(1+t_n^2)^2}\left(\vec{v}_{n-\frac{1}{2}} \times \vec{t}_n\right)^2 \\
&= 0.
\end{aligned} \tag{15}$$

In the third and fifth line, we use the equality $(\vec{A} \times \vec{B}) \cdot \vec{C} = (\vec{B} \times \vec{C}) \cdot \vec{A}$. In the eighth and second last line, we use the fact that a vector is orthogonal to its cross product to another vector. In the third last equation, we make use the identity $(\vec{A} \times \vec{B})^2 = A^2B^2 - (\vec{A} \cdot \vec{B})^2$.

Now if we want to consider the electric field, we modified the implicit forward method to

$$\vec{v}_{n+\frac{1}{2}} - \vec{v}_{n-\frac{1}{2}} = \frac{qdt}{m}\vec{E}(x_n) + \frac{qdt}{2m}\left(\vec{v}_{n+\frac{1}{2}} + \vec{v}_{n-\frac{1}{2}}\right) \times \vec{B}(x_n). \tag{16}$$

Let $\vec{s}_n = qdt\vec{E}(x_n)/2m$, observe that we can rewrite it as

$$\left(\vec{v}_{n+\frac{1}{2}} - \vec{s}_n\right) - \left(\vec{v}_{n-\frac{1}{2}} + \vec{s}_n\right) = \left(\left(\vec{v}_{n+\frac{1}{2}} - \vec{s}_n\right) + \left(\vec{v}_{n-\frac{1}{2}} + \vec{s}_n\right)\right) \times \vec{B}(x_n), \tag{17}$$

which has the same form without the electric field. In the case, we can solve conduct the forward by

$$\begin{aligned}\vec{u}_{n-\frac{1}{2}} &= \vec{v}_{n-\frac{1}{2}} + \vec{s}_n;\\ \vec{w}_n &= \vec{u}_{n-\frac{1}{2}} + \vec{u}_{n-\frac{1}{2}} \times \vec{t}_n;\\ \vec{u}_{n+\frac{1}{2}} &= \vec{u}_{n-\frac{1}{2}} + \frac{2}{1+t_n^2}\vec{w}_n \times \vec{t}_n;\\ \vec{v}_{n+\frac{1}{2}} &= \vec{u}_{n+\frac{1}{2}} + \vec{s}_n;\end{aligned} \tag{18}$$

with

$$\vec{t}_n = \frac{qdt}{2m}\vec{B}(x_n) \tag{19}$$

and

$$\vec{s}_n = \frac{qdt}{2m}\vec{E}(x_n). \tag{20}$$

Also, the velocity change of the charge is given by

$$\begin{aligned}v_{n+\frac{1}{2}}^2 - v_{n-\frac{1}{2}}^2 &= (\vec{u}_{n+\frac{1}{2}} + \vec{s}_n)^2 - (\vec{u}_{n-\frac{1}{2}} - \vec{s}_n)^2\\ &= 2\Big(\vec{u}_{n+\frac{1}{2}} + \vec{u}_{n-\frac{1}{2}}\Big) \cdot \vec{s}_n\\ &= \Big(\vec{v}_{n+\frac{1}{2}} + \vec{v}_{n-\frac{1}{2}}\Big) \cdot \Big(2\vec{s}_n\Big),\end{aligned} \tag{21}$$

which gives the correct answer. Finally, the trajectory can be obtained by numerical integration on the velocity.

2.3 Relativistic Boris Method

The Lorentz force in a relativistic situation is described by

$$\frac{d}{dt}\Big(\gamma(v)m\vec{v}\Big) = \frac{q}{c}\Big(\vec{E} + \frac{\vec{v}}{c} \times \vec{B}\Big). \tag{22}$$

It is similar to solve the equation with Boris' method, though we have to define

$$\vec{U} = \gamma(v)\vec{v} \tag{23}$$

so that

$$\vec{v} = \frac{\vec{U}}{\sqrt{1+U^2}} \tag{24}$$

and the equation can be rewritten as

$$\frac{d}{dt}\Big(m\vec{U}\Big) = \frac{q}{c}\Big(\vec{E} + \frac{1}{c}\vec{U} \times \Big(\frac{1}{\sqrt{1+U^2}}\vec{B}\Big)\Big). \tag{25}$$

As we find that if we replace $\vec{B}$ in the original Boris' method and replace it by $\vec{B}/\sqrt{1+U^2}$, we can solve the equation with exactly the same intermediate steps, thought we have to integrate the displacement by

$$\vec{r}_f = \vec{r}_i + \sum_i \frac{\vec{U}_i}{\sqrt{1+U_i^2}} \tag{26}$$

instead of $\vec{U}$ itself. Also, it's always better to calculate the real velocity in case we make v larger than c but do not recognize it since $|\vec{U}|$ is not bounded, though doing this inspection for every loop would slightly increase the computational cost.

3 Implementations

In this section, we will implement our code in three different kinds of problem. The first one is a test problem. As we are familiar with the circular motion of a charge in a uniform magnetic field, we can compare the analytic solution and the numerical solution solved by the Boris' method, which is helpful for us to understand the error and convergence of the program we write down. Secondly, we examine the situation where the magnetic field is stronger than the electric field and vice verse, with both being uniform. Since we can go to frame where one becomes zero, calculate the trajectory, and go back to the Lab frame, the trajectories are also predictable.

The third implementation is to recreate the Fig. 3 in Zenitani et al. (2018). It's a comparison between the Runge-Kutta method and the Boris' method. In this paper, with Boris' method, it is shown that the orbit of a charge in some axially symmetric electric field and axially symmetric is similar to that of the moon. However, if we use Runge-Kutta instead, the sub-periodic motion of the charge will die out and becomes a circular motion.

The last implementation is to numerically verify Fig.12.2 in Classical Electrodynamics by Jackson. As we mentioned earlier, the angular frequency difference due to a non-uniform magnetic field will cause a charge gradually moving in the direction perpendicular both to the magnetic field and the gradient of its magnitude.

For convenience, we shall change the variables into dimensionless ones so that we will make no trouble in numerical practice. Let $\tau = t/t_0$ for some desired time scale t_0, with the observation that we can write the relativistic Lorentz force equation as

$$\frac{d}{d\tau}\Big(\gamma(v)\vec{\beta}\Big) = \frac{t_0 q}{mc}\Big(\vec{E} + \vec{\beta}\times\vec{B}\Big). \tag{27}$$

As $K = mc/t_0 q$ has the same dimension with the electric and magnetic field, if we let $\vec{\mathcal{E}} = \vec{E}/K$ and $\vec{\mathcal{B}} = \vec{B}/K$, then we can write the equation with dimensionless variable only. With little ambiguity, we keep the notation E and B below. Together with the discussion on Boris' method in last section, we can write a code in Python as

```
import numpy as np
import matplotlib.pyplot as plt

# Initialization
time = 1.0
num_iter = 100000
dt = time/num_iter
pos = np.zeros((3, num_iter)) #Initial Position = (0,0,0)
vel = np.zeros((3, num_iter))
vel[:,0] = [0.1, 0.0, 0.02]

# Prescribed Electric and Magnetic Field
def magnetic_field(pos):
    return np.array([0.0, 0.0, 1.0])
def electric_field(pos):
    return np.array([0.0, 0.0, 0.0])

# Relativistic Boris Solver
for i in range(0, num_iter-1, 1):
    pos[:,i+1] = pos[:,i] + dt*vel[:,i]
    U_vec = vel[:,i]/np.sqrt(1-np.linalg.norm(vel[:,i])**2) #gamma(v)v
    t_vec = (dt/2.0)*magnetic_field(pos[:,i+1])/np.sqrt(1+np.linalg.norm(U_vec)**2)
    s_vec = (dt/2.0)*electric_field(pos[:,i+1])
    u_vec = U_vec + s_vec #Boris Solver
    w_vec = u_vec + np.cross(u_vec, t_vec)
    u_vec = u_vec + (2.0/(1.0+np.linalg.norm(t_vec)**2))*np.cross(w_vec, t_vec)
    vel[:,i+1]= (u_vec + s_vec)/np.sqrt(1+np.linalg.norm(u_vec)**2) #Recover v from gamma(v)v
```

3.1 Task I : Charge in a Uniform Magnetic Field

In this first implementation of the Boris solver, the electric field is set to be vanishing, and magnetic field is prescribed as $\vec{B} = B_0\hat{z}$ with $B_0 = K$ namely ($|\vec{\mathcal{B}}| = 1$). The initial position is set to be the origin without loss of generality, and the initial velocity is set to be to $v = (0.1, 0, 0.02)$.

Since the magnetic field is uniform, namely, $\vec{B} = B_0\hat{z}$, the trajectory is known to be

$$(x, y, z)(\tau) = \Big(\frac{0.1ct_0}{\omega}\sin(\omega\tau), \frac{0.1ct_0}{\omega}\Big(1 - \cos(\omega\tau)\Big), 0.02c\tau\Big), \tag{28}$$

where

$$\omega = \frac{q|\vec{B}|}{mc\gamma(v)} = \Big(\frac{|\vec{B}|}{K\gamma(v)}\Big)\frac{1}{t_0} = \Big(\frac{1}{\gamma(v)t_0}\Big) \tag{29}$$

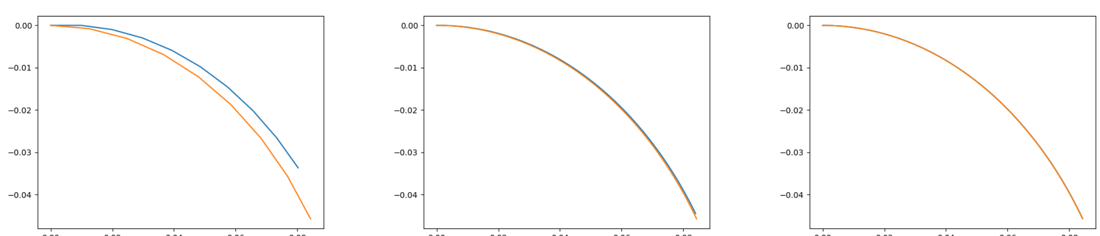
is the cyclone frequency. Also, the position in the program is actually $(x/ct_0, y/ct_0, z/ct_0)$, but we omit ct_0 as there's little ambiguity. As the L^2 norm error is defined as

$$\epsilon_{L^2} = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}\Big(\vec{r}_{num,i} - \vec{r}_{th,i}\Big)^2}, \tag{30}$$

we can execute the code and make the following table.

Partition Number	L^2 norm error	Time Cost (s)
1.0 E+1	1.253 E-2	1.603 E-3
1.0 E+2	1.298 E-3	1.671 E-2
1.0 E+3	1.303 E-4	1.413 E-1
1.0 E+4	1.304 E-5	1.449 E+0
1.0 E+5	1.304 E-6	1.431 E+1
1.0 E+6	1.304 E-7	1.472 E+2

Table 1: L^2 Norm Error and Time Cost of the Boris Solver on the Test Problem.



(a) Partition Number = 10 (b) Partition Number = 100 (c) Partition Number = 1000

Figure 1: The numerical solutions (Blue) compared to analytic solution (Orange) of the test

As one can easily see, the L^2 error is linear to the step size and converges to zero as the step size goes to zero. This means our code is accurate as long as the step size is sufficiently small. Also, as we show in 1, the trajectory also graphically converges to the analytical solution impressively.

3.2 Task II : Stronger Magnetic Field or Stronger Electric Field

In a uniform electric and magnetic field, with the E-field being perpendicular to the magnetic field, if the magnitude of one field is larger than another one, it's possible to find a frame where another field vanishes in that frame. As the Lorentz transformation is linear, we shall still have similar trajectories with some distortion.

If the electric field is stronger, then there's possibly a frame with only electric field, and the trajectory of the charge shall be a parabola on the plane of its initial velocity and the electric field.

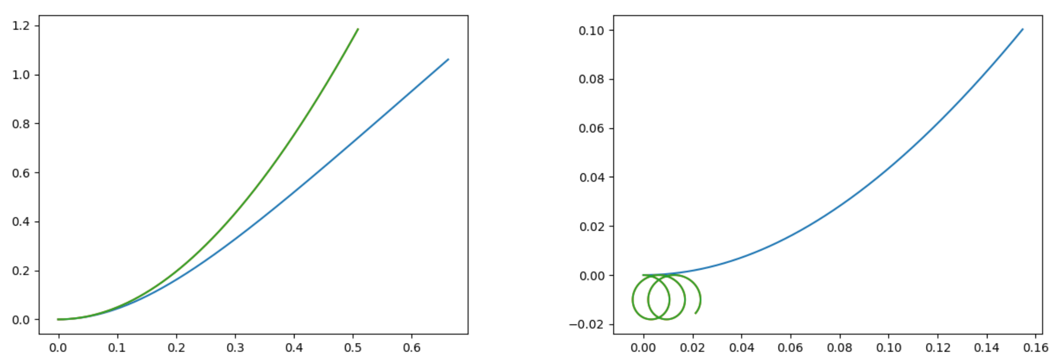
Switching back to the Lab frame, we shall still see a parabola as Lorentz transformation is a linear transformation. If the magnetic field is stronger, then the charge is in a circular motion in the frame where the electric field disappears. As we switch back to the Lab frame, the circular motion shall be constantly drifted.

To verify the argument, we consider three different values of E and B field strength, whose directions are in the y and z axis respectively. The initial velocity is always set to be $v_0 = 0.1\hat{x}$, and the reference case is set to be the equal field strength condition $|\vec{E}| = |\vec{B}| = 0.1$. These values are chosen to make the effects clear but with enough time for the charge to move since the charge particle could easily get accelerated to nearly the speed of the light if $|\vec{E}| \geq 1$ within $t < 10$.

For the stronger electric field case, we set $\vec{B} = 0.01\hat{z}$ and $\vec{E} = 0.1\hat{y}$. In Fig. 2 (a), we run the code with time span from 0 to 5 with step size $dt = 1.0E - 5$, which is chosen such that the difference from the reference case can be easily seen. As the figure shows, the trajectory is likely a parabola, which obeys our expectation.

For the stronger magnetic field case, we set $\vec{B} = 10\hat{z}$ and $\vec{E} = 0.1\hat{y}$. In Fig. 2 (b), we run the code with time span from 0 to 1.5 with step size $dt = 1.0E - 5$, which is chosen such that the drifted circular motion can be easily observed. As the figure shows, the trajectory is really a drifted circular motion along the direction perpendicular to the electric field. We do expect this since the charge has larger velocity at larger y, so it has an average velocity in the positive x direction.

Furthermore, when $|\vec{E}| = |\vec{B}|$, it seems that the parabolic acceleration to $+y$ and the circular motion to $-y$ cancel each other, and the trajectory seemingly converges to a straight line, but we do not have further evidence.



(a) Trajectory as $|\vec{E}| > |\vec{B}|$

(b) Trajectory as $|\vec{E}| < |\vec{B}|$

Figure 2: The trajectories with different $|\vec{B}|/|\vec{E}|$ (Green) compared to the $|\vec{E}| = |\vec{B}|$ trajectory (Blue)

3.3 Task III : Sustained Details in a Long Run

In the paper S. Zenitani and T. Umeda, On the Boris Solver in particle-in-cell simulation, Phys. Plasmas 25: 112110 (2018), they consider the following axially symmetric and afar-decaying magnetic and electric field:

$$\begin{aligned}\vec{B} &= (0, 0, \sqrt{x^2 + y^2}) \\ \vec{E} &= (-\frac{0.01x}{\sqrt{x^2 + y^2}^3}, -\frac{0.01y}{\sqrt{x^2 + y^2}^3}, 0),\end{aligned} \tag{31}$$

and initial velocity $\vec{u} = \gamma(v)\vec{v} = (0.1, 0, 0)$ and initial position $(0.9, 0, 0)$.

In the end of the fourth section, with this setup, they show the advantage of the Boris' method compared to the Runge-Kutta solver. Specifically, they run the program with $t \approx 2 \times 10^5, dt = \pi/10$, and show that the Boris solver gives a stable trajectory where the charge particle is orbiting around a point which is also moving circularly moving around the origin. However, with Runge-Kutta method, at the 300th turn, or around the $t = 2 \times 10^5$, the smaller circular motion of the charge particle disappears, which leaves a simple circle trajectory. This implies, in the long run, the Boris' method can keep the details and accuracy relatively impressive in comparison with the widely used Runge-Kutta integration, as they show in Fig.4.

To verify their result, we implement our solver with the same electric and magnetic field within time span $t = 2 \times 10^5$ but with a smaller time step $dt = 0.01$. Since it takes around $\Delta t = 800$ for the charge to complete circulating about the origin, we take the first and last 800 seconds (in fact t_0) and plot them on the same graph Fig.3 so that it's more easy to see the difference.

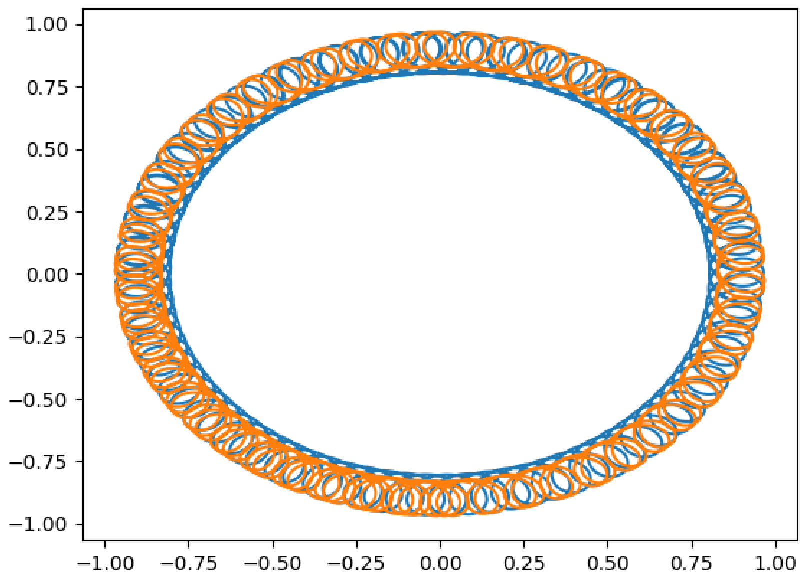


Figure 3: First (blue) and last (orange) 800 seconds of the trajectory with Boris Method

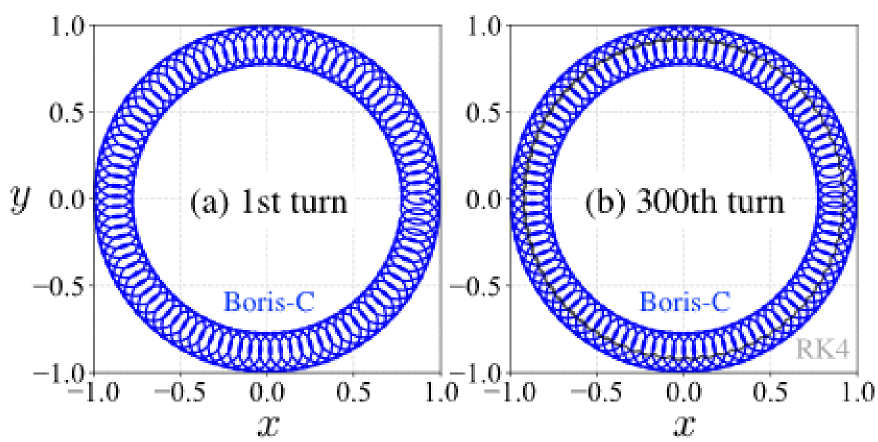


Figure 4: Fig. 3 in Phys. Plasmas 25: 112110 (2018)

The result is pretty close to what is obtained in Phys. Plasmas 25: 112110 (2018), as the details are well kept in such a long run, as the time cost is roughly 1300 seconds. There's a little difference between the blue and the orange curves, as the latter one have a larger minor circular motion radius. This is seemingly not seen in the reference paper, but as they did not plot the trajectory in a single graph and the difference is not sufficiently obvious, we do not know the difference is a numerical error or not. However, we still succeed to recreate the result in the referred paper.

3.4 Task IV : Non-uniform B-field with Gradient Perpendicular to Itself

For the last task to test out our program, we consider the following magnetic field

$$\vec{B} = B\hat{z} = (x\cos\theta + y\sin\theta)\hat{z}, \tag{32}$$

where θ is a free parameter to decide the direction of the gradient of the magnetic field as

$$\hat{n} = \vec{\nabla}B = (\cos\theta, \sin\theta, 0) \perp \hat{z}. \tag{33}$$

Also, the Maxwell equation is still obeyed as

$$\vec{\nabla}\cdot\vec{B} = \frac{\partial}{\partial z}(x\cos\theta + y\sin\theta) = 0. \tag{34}$$

Although with a vanishing electric field, we must have a current

$$\vec{J} \propto \vec{\nabla}\times\vec{B} = (\sin\theta, -\cos\theta, 0). \tag{35}$$

We set the initial position to be the origin and initial velocity $\vec{v}_0 = (0.1, 0, 0)$. The we run the program with time span $t = 100$ and number of grids 100000 so that each time step is $dt = 0.001$. With $\theta = 0, \pi/4$, and $\pi/2$, we run the Boris' solve with the same setup above, and the trajectories for different θ are plot in 5.

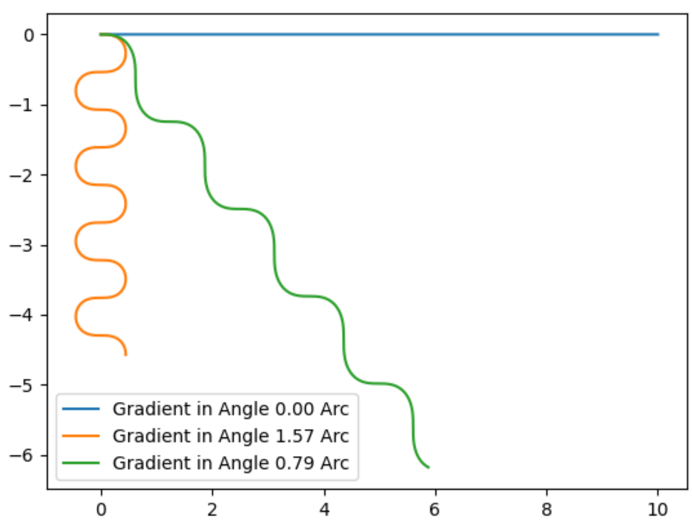


Figure 5: Trajectories when the gradient of B-field is perpendicular to itself

As we previously discussed, due to the increase of the magnetic field in the direction of the gradient of the magnitude of the magnetic field, the angular frequency (which is perpendicular the the B-field) difference in the direction perpendicular to the gradient of magnitude of the magnetic field will cause the circular motion to be drifted gradually.

4 Conclusion and Discussion

In the first part of the report, we analytically discuss the energy from the void problem in the Euclid explicit forward method, and we discuss how the Boris' method resolve this problem via some intermediate steps, which makes the kinetic energy conserved as there's only the magnetic field presenting. This also corresponds to the third task we conduct with the Boris' solver.

In the first task, we run the Boris' solver solely under a uniform magnetic field, which analytically gives a trajectory of circular motion. The numerical result nicely matches the analytical solution, as the L^2 norm error decreases linearly with the time step size, which supposedly converging to zero as the time step does. In fact, we roughly have $\epsilon_{L^2} = 0.13dt$. Graphically, we also show that the trajectory gets unrecognizably close with merely $dt = 0.001$.

In the second task, we use the Boris' solver for $|\vec{E}| > |\vec{B}|$ and $|\vec{E}| < |\vec{B}|$ and compare the numerical solutions to the case $|\vec{E}| = |\vec{B}|$. For the former case, we expect the trajectory to be a parabola as a trajectory does with only electric field, and the numerical result is consistent with the argument. In the second case, we expect the trajectory to be a drifted circular motion, and the numerical result does exhibit such a motion.

In the third task, we recreate a result in Phys. Plasmas 25: 112110 (2018). In this paper, they consider an axially symmetric electric and magnetic field, where the Runge-Kutta integration will lose the details of the motion terribly in a long run. However, they show that the Boris' method can impressively cure this problem. With the same time span but a much smaller time step, we find a similar result that the details of the motion in the early time are sustained despite the long iteration time.

In the last task, we consider the movement of a charge in a non-uniform B-field whose magnitude has a gradient perpendicular to the magnetic field itself. With different direction of the gradient of the magnitude of the magnetic field, we plot the trajectories and find that the numerical result is consistent with the discussion in Sec. 12.4 in Classical Electrodynamics (Ed.3) by J. D. Jackson.

Summarizing, we make a rigorous discussion to answer why the Boris' method behaves better than the common forward method since it does not create kinetic energy from the numerical void. The Boris' method also impressively complete four tasks in a satisfying manner: the error is verified to be small and controllable, it can predict the trajectory consistent with our discussion on the ratio between the electric and magnetic field strength, and it can also overwhelmingly sustain the details of the motion in comparison with the Runge-Kutta solver. As a result, we can confidently say the Boris' solver is an excellent method as the Lorentz force is concerned.

A The Full Python Code

Here we provide the full Python after we complete all the tasks, though it's not made readable.

```python
import numpy as np
import matplotlib.pyplot as plt
import time
# Initilization
start_time = time.time()
time_interval = 100.0
num_iter = 100000
dt = time_interval/num_iter
pos = np.zeros((3, num_iter))
vel = np.zeros((3, num_iter))
pos[:,0] = [0.0, 0.0, 0.0]
vel[:,0] = [0.1, 0.0, 0.0]

# Given Electric and Magnetic Field
def magnetic_field(pos):
    return np.array([0.0, 0.0, pos[0]])
def electric_field(pos):
    return np.array([0.0, 0.0, 0.0])

# Relativistic Boris Solver
for i in range(0, num_iter-1, 1):
    pos[:,i+1] = pos[:,i] + dt*vel[:,i]
    U_vec = vel[:,i]/np.sqrt(1-np.linalg.norm(vel[:,i])**2)
    t_vec = (dt/2.0)*magnetic_field(pos[:,i+1])/np.sqrt(1+np.linalg.norm(U_vec)**2)
    s_vec = (dt/2.0)*electric_field(pos[:,i+1])
    u_vec = U_vec + s_vec
    w_vec = u_vec + np.cross(u_vec, t_vec)
    u_vec = u_vec + (2.0/(1.0+np.linalg.norm(t_vec)**2))*np.cross(w_vec, t_vec)
    vel[:,i+1]= (u_vec + s_vec)/np.sqrt(1+np.linalg.norm(u_vec)**2)

end_time = time.time()
plt.plot(pos[0,:],pos[1,:])

#Exact Solution
#time_set = np.linspace(0,time_interval,num_iter-1)
#omega = -1.0*np.sqrt(1-0.1**2-0.02**2)
#x_exact = (0.1/omega)*np.sin(omega*time_set)
#y_exact = (0.1/omega)*(1-np.cos(omega*time_set))
#z_exact = 0.02*time_set

#mse_error = np.linalg.norm(x_exact-pos[0,0:num_iter-1])**2
#mes_error += np.linalg.norm(y_exact-pos[1,0:num_iter-1])**2
#mes_error += np.linalg.norm(z_exact-pos[2,0:num_iter-1])**2
#mes_error = np.sqrt(mes_error)
print(end_time-start_time)
#print(mse_error)

#plt.legend(loc="lower left")
# Plot
#plt.plot(pos[0,19920000:19999998],pos[1,19920000:19999998])
#plt.plot(x_exact[:],y_exact[:])
plt.show()
```