Numerical Study of the Motion of Relativistic Particles in Arbitrary EM Fields

Bottero Luca

Abstract

The accurate prediction of relativistic particle trajectories in electromagnetic (EM) fields is vital for scientific and technological applications in astrophysics, plasma physics, and accelerator physics. This work presents a numerical study of relativistic particle motion in arbitrary EM fields, employing the Boris algorithm for its widespread use and effectiveness in solving equations of motion. By comparing computed results with analytical solutions, we validate the algorithm's accuracy and demonstrate its ability to accurately integrate the equations in a time-accurate and stable manner.

1 Introduction

1.1 Problem statement

The motion of charged relativistic particles subjected to the interaction with an external electromagnetic field (EM) is governed by the Lorentz force (here expressed in SI units):

$$\frac{d(\gamma \mathbf{v})}{dt} = \alpha(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \tag{1}$$

where $\gamma = (1 - v^2/c^2)^{-\frac{1}{2}}$ is the Lorentz' factor and $\alpha = q/m$ with q and m the particle's charge and mass respectively. In the non relativistic limit the LHS of the equation becomes

$$\frac{d\mathbf{v}}{dt} = \alpha(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \tag{2}$$

which is a linear first order differential equation. The trajectory of point particles obeying Eq. (1) can be analytically found for some configurations of the EM fields. However, those scenarios are often an abstraction of much more complex field configurations found in nature and technology, giving rise to results that are only qualitatively correct.

Despite being physically interesting and useful to calculate approximate (sometimes their accuracy can be so close to reality that it coincides with it for every practical purpose) quantities, the analytical solutions are not available for arbitrary EM field configurations. In addition, the motion of charged particles creates itself electrical and magnetic field which will modify the path (and hence the creations of other EM fields) of other particles. This makes the microscopic motion of charged particles chaotic and impossible to solve analytically for every practical situation. Furthermore, additional particle-particle interactions or external fields are needed to correctly model the behaviour of real particles (such as chemical and nuclear reaction and gravity).

If we are interested only in the collective motion of a (big enough) collection of particles, we can resort to the tools of statistical mechanics (leading to plasma physics); however, if we are concerned about the motion of the single particles, we do need to resort to numerical integration of Eq. (1), which is the aim of the present work.

This work is organized as follows: in sections 1.2 and 1.3 we briefly discuss the relevance of the problem for physics and engineering and why we choose the Boris algorithm. In section 2 we outline the Boris Algorithm, stating and commenting the computational steps needed in section 2.1. We then illustrate several numerical experiments in section 3, with increasing field configuration complexity, comparing numerical results with analytical one

(when available). In the final section 4 we discuss the obtained results.

1.2 Physical relevance of the problem

The study of both individual and collective motion of charged particles arises in several key topics of modern physics. The study of plasma has applications in astrophysics: ranging from the physics of solar corona to the motion of individual electrons or ions with extremely high gamma factor inside the galactic magnetic field, charged particles are everywhere in the cosmos and therefore the study of their motion is of top concern as a tool to probe the evolution of the cosmos as well as the working of stellar objects.

Relatively small packets of charged particles are used in particle accelerators to understand the fundamental physics of the universe: the precision needed to create and maintain highly relativistic charged particles in a coherent packet is formidable undertake that needs an accurate computation of several aspects of the motion of particle in complex magnetic fields.

A technological application of plasma physics with the potential to be a long-lasting changing to how the humanity lives is (controlled) nuclear fusion: both experimental and computational efforts are needed to understand the conditions needed to exploit the power of nuclear fusion in a practical and economical manner, since the manipulation of high density, high temperature plasma of different particles immersed in strong non-uniform magnetic field is a very complex task.

1.3 Employed method

In this work we are not concerned with the creation of EM fields from the particles, so that we can concentrate on the study of the motion of the particles inside external EM fields. We will also ignore all other particle-particle interactions as well as any other external force.

Numerically integrating equation 2 is simpler due to its linear nature compared to the relativistic one. However, in real-case scenarios, numerical simulations often involve particle that are at least mildly-relativistic. Therefore, in this study, we examine the solution to equation 1 in both its classical and ultra-relativistic limits. To achieve this, we employ the widely-used Boris algorithm, which is widely used in astrophysical simulations.

2 The Boris Algorithm

The Boris algorithm [Bor+70] is second-order, symplectic algorithm that exhibits desirable properties, such as time-accuracy and stability, which make it suitable for accurately tracking the trajectories of relativistic particles in electromagnetic fields. An highly valuable propriety of this scheme is its time reversibility, which differentiates it from the Runge-Kutta schemes and allows the evolution of a particle's trajectory backward in time. It effectively handles relativistic effects, where the particle's velocity approaches the speed of light. Additionally, the Boris algorithm is closely related to the leapfrog method, as it shares similarities with its structure [Hon13]. A brief description of the computational steps that composes it is provided below.

2.1 Computational steps

Firstly, the midpoint position is obtained drifting the particle's position using its initial velocity:

$$\mathbf{x}_p^{n+\frac{1}{2}} = \mathbf{x}_p^n + \frac{\Delta t^n}{2}\mathbf{v}_p^n$$

Next, a rotation of the half-step velocity is performed using the magnetic field and the electrical field is applied again to get the final value of the 3-velocity. To perform this step, some intermediate variables are useful:

$$\mathbf{u}_p^- = \mathbf{u}_p^n + \frac{h}{2}c\mathbf{E}^{n+\frac{1}{2}}$$

$$\mathbf{u}_p^+ = \mathbf{u}_p^- + 2\frac{\mathbf{u}_p^- + \mathbf{u}_p^- \times \mathbf{b}}{1 + \mathbf{b}^2} \times \mathbf{b}$$

$$\mathbf{u}_p^{n+1} = \mathbf{u}_p^+ + \frac{h}{2}c\mathbf{E}^{n+\frac{1}{2}}$$

where $h = \alpha_p \, \mathrm{d}t$, $\mathbf{b} = \frac{h\mathbf{B}^{n+\frac{1}{2}}}{2\gamma^-}$, $\gamma^- = (1 + (u^-/c)^2)^{\frac{1}{2}}$ and $\mathbf{u}_p = \gamma \mathbf{v}_p$.

Since $\mathbf{u}^+$ is a rotated version of $\mathbf{u}^-$, we have $\gamma^- = \gamma^+$. The $\mathbf{E}$ and $\mathbf{B}$ fields are calculated at the midpoint position $\mathbf{E}^{n+\frac{1}{2}} = \mathbf{E}(\mathbf{x}^{n+\frac{1}{2}})$ and $\mathbf{B}^{n+\frac{1}{2}} = \mathbf{B}(\mathbf{x}^{n+\frac{1}{2}})$.

The velocity of the particle is then calculated as follows: $\mathbf{v}^{n+1} = \mathbf{u}^{n+1}/\gamma^+$. Finally, the particle's position is updated:

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^{n+\frac{1}{2}} + \frac{\Delta t^n}{2}\mathbf{v}_p^{n+1}$$

The C++ code used to perform one step of the Boris algorithm is reported in appendix C.

3 Numerical experiments

In this section we present some numerical experiments with the goal to asses the correctness of the implementation of the Boris scheme as well as to understand the motion of non-interacting particle in well known EM field configurations which are relevant for physics.

3.1 Gyration

This experiment can be used to understand several different characteristic of the Boris scheme. The electrical field is vanishing everywhere $\mathbf{E} = (0, 0, 0)$ and the magnetic field is uniform in space, directed in the $\hat{z}$ direction $\mathbf{B} = (0, 0, B_0)$. The motion of the particle in this field configuration is well known from undergraduate courses in electromagnetism and it's trajectory is, in general, a circular spiral, with a linear motion in the $\hat{z}$ direction (fig. 4). Without loss of generality, we can neglect the v_z component of the velocity: by equating the modulus of the Lorentz force in absence of the electrical field for a velocity perpendicular to the magnetic field to the modulus of the centripetal force we can get the value for the radius of the trajectory and its period:

$$r = \frac{\gamma v m}{qB} \tag{3}$$

$$T = \frac{2\pi m \gamma}{eB} \tag{4}$$

where m is the rest mass of the particle. The study of this motion is critical to show that the magnetic field does not do work, since a radial motion be only be obtained by modifying the particles' kinetic energy. To test the numerical results against theoretical calculations, we performed different test cases. In each case, a magnetic field $\mathbf{B} = 1\hat{z}$ Tesla was set. Here we schematically include the description and the results of each experiment:

1. **Convergence**: The electron's charge and mass were used for this simulation. We run a simulation with 3 different initial conditions: $\gamma = 10$, $\gamma = 10^4$ and $\gamma = 10^6$. The corresponding cyclotron radius r computed from Eq. (3) was then calculated from the given velocity and the initial position of the particle was set to $\mathbf{r} = (r, 0, 0)$. The simulation duration t_{fin} was set to be equal to one period of the gyration from Eq. (3). We then run several simulation, using each time a different timestep $dt_n = T/N_{steps}$, with $N_{steps} = 4, 8, \ldots, 2^{20}$. In order to show the convergence of the algorithm, we monitored 4 metrics for each simulation:

   (a) The final radius $r_{fin} = \sqrt{x^2 + y^2}$

   (b) The phase offset $\phi = \arctan\left(\frac{y_{fin}}{x_{fin}}\right)$

   (c) The distance from starting point $|\mathbf{r}_{fin} - \mathbf{r}|$

   (d) The final vertical displacement y_{fin}

The results are plotted in fig. 1 - fig. 3.

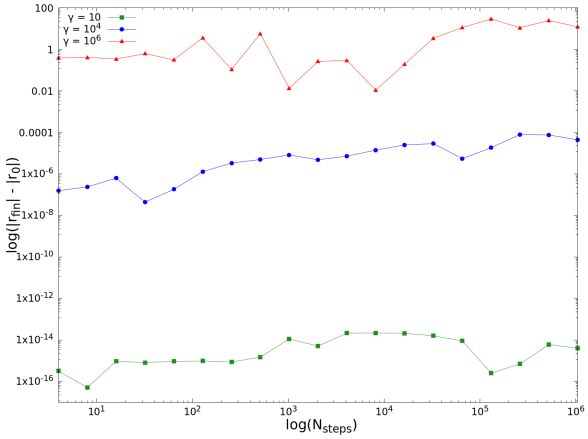

Figure 1: Log-log plot of the final radius $r_{fin} = \sqrt{x^2 + y^2}$ after one turn of the particle as function of the timestep number.

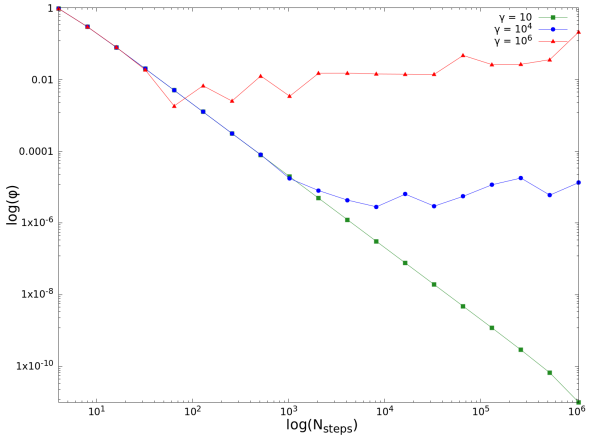

Figure 2: Log-log plot of the phase offset $\phi = \arctan\left(\frac{y_{fin}}{x_{fin}}\right)$ after one turn of the particle.

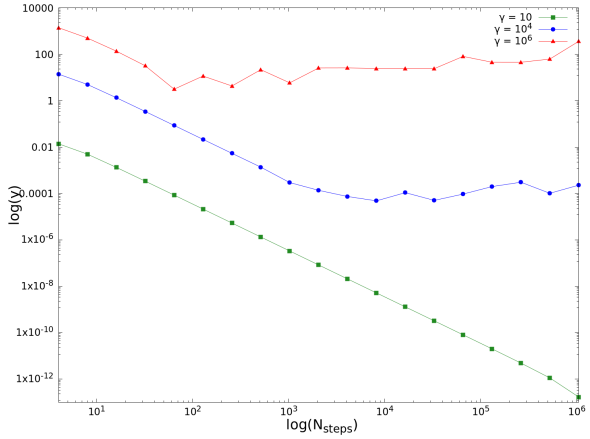

Figure 3: Log-log plot of the final vertical displacement y_{fin} after one turn of the particle.

We observe that the radius is conserved with an error that does not depends on the value of n, with only "noise" superimposed to an otherwise flat error line. The error for the faster particle is several order of magnitude greater than those of the slower particle, indicating that the Boris scheme may not be suitable if high accuracy is needed for ultrarelativistic particles.

The phase offset exhibits an evident (see fig. 2) decrease of the error as the number of timesteps grows. This behaviour is compatible with the expected behaviour of the Boris algorithm, which is 2nd order in the position: the log-log plot exhibits a linear decrease of inclination -2. However, this downward trend stops after some value of N_{steps}: in particular, faster particles suffer earlier of this phenomena. This poses a limitation of the Boris scheme since, after reaching a critical value of the number of timesteps, increasing it further does not yield an increased accuracy.

This relationship between accuracy and number of timesteps is further verified by looking at the final y displacement y_{fin}, as shown in fig. 3: the downward trend has a linear relationship with an inclination of -2 in the log-log plot.

In appendix B the plot for the conservation of γ (fig. 10) confirms the result of the radius conservation, since the error can be attributed to a non-perfect conservation of the γ value.

Since the distance from the starting point is closely related to the angular displacement of the particle at the final moment of the simulation, the result follows closely those of the angular displacement (see 9 in appendix B).

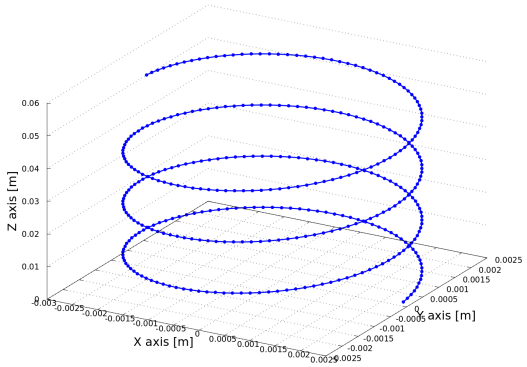

Figure 4: Trajectory of an electron subject to a 1 Tesla upward magnetic field with initial velocity $\mathbf{v} = (0, 0.9c, 0.9c)/\sqrt{2}$ evolved for $t_{final} \sim 2.9 \cdot 10^{-11}$ s.

2. **Parallel transport**: The initial conditions are the same of the previous example except for the fact that the velocity was set to $\mathbf{v} = (0, v_0/\sqrt{2}, v_0/\sqrt{2})$. The radius and period are calculated using v_z as the speed value. The measured metric is the ratio between $v_0 t_{fin}/\sqrt{2}$ and the final displacement in the $\hat{z}$ direction. No deviations from the target value of 1 for γ up to 10^7 were observed: all the deviations from a purely unitary value can be explained

with numerical precision issues. The scheme fails for $\gamma = 10^8$.

3. **Radius conservation**: in order to show that the motion is indeed circular the distance of the particle's position from the center must be constant: $dr/dt = 0$, with $r = \sqrt{x^2 + y^2}$ (since the initial position is defined as $(r, 0, 0)$, forcing the center to be the origin of the reference frame). We run each simulation with the particle's initial gamma factor equal to $\gamma = 10^i$ for $i = 1, \ldots, 8$. For each run of the simulation we calculate the mean relative absolute error between the particle's distance from the center of the reference frame and the actual radius:

$$\text{MRAE} = \frac{1}{N_{steps}} \sum_{i=1}^{N_{steps}} \left| \frac{\sqrt{x_i^2 + y_i^2}}{r} - 1 \right|$$

The result for $N_{steps} = 360$ is listed in table 3. The MRAE raises linearly as a function of $\log_{10} \gamma$. The scheme fails for $\gamma = 10^8$.

$\log_{10} \gamma$	MRAE
1	5.85723×10^{-14}
2	7.98274×10^{-12}
3	4.30407×10^{-10}
4	1.63462×10^{-07}
5	3.54517×10^{-06}
6	0.0021256
7	0.0391391

Table 1: Mean absolute relative error of the distance of particle from the correct value as a function of the initial γ factor.

3.2 $\mathbf{E} \times \mathbf{B}$ drift

We will now relax the vanishing electrical field constraint to introduce an uniform field of the form $\mathbf{E} = (0, E_0, 0)$ and keep the uniform, unitary upward magnetic field $\mathbf{B} = (0, 0, B_0)$. The study of this motion is easy to solve in the classical regime. In this situation the particle experiences a *drift* due to the superposition of the acceleration caused by the electrical field and the rotation induced by the Lorentz force. Starting at rest, the particle acquires velocity due to the $\mathbf{E}$ field; then particle is subject to a perpendicular Lorentz force that curves it trajectory away from the electrical field, resulting in a diminishing acceleration: When the particle's velocity is perpendicular to the $\mathbf{E}$ field the direction of motion is inverted until the particle returns to the original y position, displaced by a certain amount on the x axis.

The described motion results in *cycloidal* motion: it is the superposition of a gyration motion and a lateral drift. A derivation of the particle's speed for the non-relativistic case is reported in appendix A:

$$\mathbf{v}_d = \frac{\mathbf{E} \times \mathbf{B}}{B^2} \tag{5}$$

For the relativistic case, the derivation of the drift speed and trajectory is quite technical and not reported here. We use the result of [Tak02]. The main difference is that the motion is the same of the classical case iff $c\mathbf{E}_0 < \mathbf{B}$, which can be understood as the fact that Eq. (5) would yield speed exceeding c otherwise. If that is the case, the particle does not drift anymore and accelerates indefinitely: the electrical force is stronger than the Lorentz's one and the resulting path is an arc (if $\mathbf{B}_0 = c\mathbf{E}_0$) or an asymptotically flat line (if $c\mathbf{E}_0 > \mathbf{B}$).

In our experiment, we check that our implementation gives the correct value for the drifting speed as a function of the E_0/B_0 ratio. In addition, we evaluate qualitatively the motion of the particle in the relativistic case to check that the correct behaviour briefly described before is reproduced.

1. **Correctness of $\mathbf{v}_d$**: In order to evaluate the value of $\mathbf{v}_d$ and compare it to the theoretical value given by Eq. (5), we need a way to eliminate the contribution of the gyration motion from the particle's trajectory. In all the experiments the particle starts at rest. The magnetic is $\mathbf{B} = (0, 0, -1)$. We run the simulation with different values of the electrical field $\mathbf{E} = (E_0, 0, 0)$: to explore "weak" electrical field we use $E_0 = 10^i c, i = -9, \ldots, -1$; after that we switch to $E_0 = c(1 - 10^{-i}), i = 2, \ldots, 3$ to explore the region where the electrical field induces relativistic effect to the test particle. We then let the simulation run as long as the

particle completes one cycle of the rotatory motion: the speed is then obtained simply by dividing the particle's final y position y_{fin} and the time needed to get into that position t_y.

As in our previous tests, the particle is an electron. We use 10^6 steps in each situation. For the "weak field" value of **E**, the final time is set to $t_{fin} = 10 * T$ with T the gyration period. For the "relativistic" field region, we use a final time equal to $T = 10^5 T$. The agreement between theory and the result of the simulation is reported in table 1. We conclude that the drift velocity is correctly simulated, the errors being the result of the method chosen to deal with the non-drift velocity components.

$E_0[V]$	$\left\|\frac{E_0}{B_0}\right\| - \left\|\frac{y_{fin}}{t_y}\right\|$	n_{eff}
2.998	1×10^{-5}	90910
29.979	1×10^{-5}	90910
299.792	1×10^{-5}	90910
2997.925	9.999×10^{-6}	90910
29979.246	9.985×10^{-6}	90910
299792.458	8.5×10^{-6}	90910
2997924.58	2.98×10^{-6}	90923
29979245.8	2.83×10^{-7}	92290
269813212.2	-0.006749989	30
296794533.42	-0.0001904	839
299492665.542	5.7×10^{-8}	26313
299762478.754	0.000259552	99999
299789460.075	0.000418725	99999

Table 2: Precision of the drift velocity as a function of electrical field strength. The last column is the effective number of time steps used in the simulation needed by the particle to reach the inversion point.

2. **Different types of drifts**: We performed the simulation for a more general motion and qualitatively compared the result with the theoretical expectations. We initialize a particle with velocity $\mathbf{v} = (0.3c, 0.4c, 0.1c)$. We used $\mathbf{B} = (0, 0, -1)$. We then performed the simulation for three different values of $\mathbf{E} = (E_0, 0, 0)$, namely the "weak" field $E_{0,w} = 0.1c$, the "equilibrium" filed $E_{0,e} = c$ and the "strong" field $E_{0,s} = 3c$. The result is shown in fig. 5 and fig. 6. The expected behaviour is correctly reproduced by our simulation.

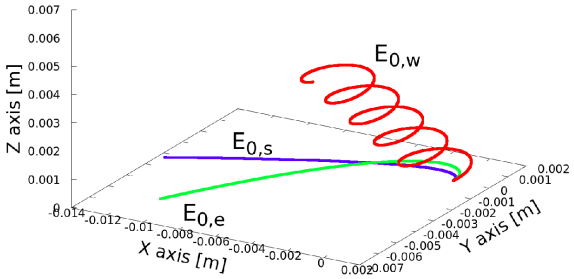

Figure 5: Simulated trajectory of an electron for different strength of the electrical field displaying the theoretically predicted behaviour.

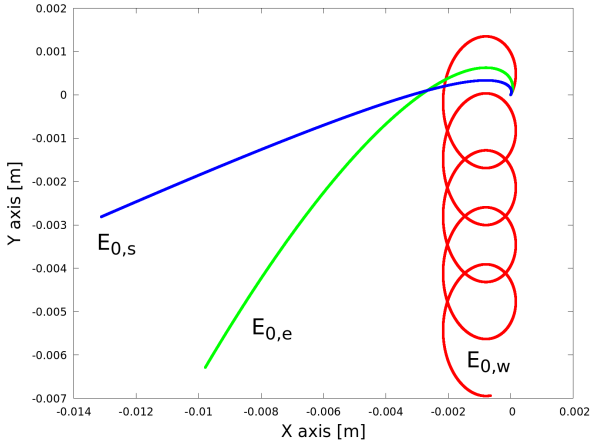

Figure 6: Projection on the xy plane of figure 5.

3.3 Magnetic X-point

In the previous two example we run simulations where the EM filed were constant and uniform: we will now relax the uniformity constraint and study the particular case of an inhomogeneous magnetic field. The field configurations of this experiment are $\mathbf{E} = (0, 0, 0.5c)$ and $\mathbf{B} = (y/L, x/L, 0)$. The magnetic field lines shows what is called an "*x-point*", that is the confluence of field lines towards the origin, where the **B** field has a *null point*.

We performed a numerical experiment using charge and mass for protons: this choice is mainly due to the fact that this field configuration is an approximate model for the field found in both astrophysical (on star objects) and artificial (particularly in tokamaks) situations, where the protons are the main constituent of the plasma.

Protons: we created $N = 1000$ protons randomly placed inside a square region of side $2L = 20$ meters centered on the origin. The velocity magnitude was set to $|\mathbf{v}| = 0.1c$ for each particle and was set to lie on the xy plane with a random orientation. We let the simulation run for $2 \cdot 10^{-4}$ seconds divided in $N_{iters} = 2000$ timesteps. The result of the simulation can be seen in fig 7 for a sample of 10 particles. In addition, we provide the density plot of the region explored by the particles in appendix B (fig. 12 and fig. 13.

We observe that the particle tend to follow the magnet field lines and to avoid the x-point: two distinct "guiding path" can be seen in the y^+ and negative y^- parts of the space, respectively. The particles display and oscillatory motion that follows the hyperbolic magnetic field lines. In addition, each particle has a net motion along the $\hat{y}$ direction. Finally, a net upward motion is gained for each particle, although the magnitude , for most particles, is small compared to that of the oscillatory motion. A "spike" develops for at the beginning of the simulation, indicating an acceleration in the $\hat{z}$ direction.

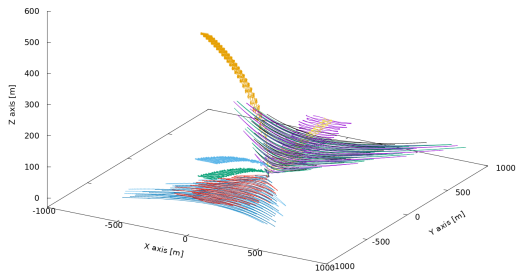

Figure 7: Trajectory for a sample of 10 particle in the x-point field configuration. The particle display evident individual differences in the relative importance of the described motion components

3.4 Magnetic Dipole approximation of Earth's Magnetic Field

We performed a simulation of the motion of charged particles subject to a magnetic dipole field. In cartesian coordinates, the field can be approximately expressed as

$$\mathbf{B} = \frac{3\mu_0 M}{4\pi r^5} \begin{cases} B_x = xz \\ B_y = yz \\ B_z = (3z^2 - r^2) \end{cases} \tag{6}$$

where $r = \sqrt{x^2 + y^2 + z^2}$ is the distance from the dipole and M is its magnetic moment. In order to simulate the behaviour of charged particle trapped in the magnetic field of Earth (which results in the Van Allen radiation belt), we approximate the planet's magnetic field with that of a dipole placed at its center, with $M = 8 \cdot 10^{22}$ A·m^2. Although this is a crude approximation, it nonetheless yields a qualitatively correct description of the coarse structure of the geomagnetic field.

We choose protons as the test particles. We initialized the particles' positions at a random distance from the dipole respecting $1.2R_\oplus \leqslant r \leqslant 3R_\oplus$, with $R_\oplus = 6.371 \cdot 10^6$ m being the radius of Earth in meters, at a random position in the xy plane (in other words: the particles initially lie in the equatorial plane). This roughly corresponds to the so called "inner Van Allen belt". Their velocity is obtained by setting a random energy between 800 keV and 1.5 MeV, which correspond roughly to the upper part of the energy spectrum of the protons found the inner Van Allen belt.

We run the simulation with 10 particles evolved for 1.5 seconds (simulation time) with $N_{steps} = 10^5$. The simulation results is displayed in figure 8. The motion of the particle can be interpreted using the "guiding center" approach: due to the fact that the magnetic field strength varies slowly as the particle moves, the resulting motion can be thought as a "gyration" motion around a field line *that is not straight*. The particle trajectory is the curved in order to "follow" the field line. However, as the field line become denser and denser (hence the magnetic gradient becomes stronger), the gyration radius of the particle begins to shrink and, due to the conservation of angular momentum, its speed increases. In addition, the gyration plane rotates to remain perpendicular to the field line.

When the rotation plane is sufficiently inclined and the particle is sufficiently fast, its motion reverses: it has reached a "magnetic mirror point". The process is then repeated in a specular fashion while going in the opposite direction. The particle oscillates back and forth from pole to pole while spiraling. The particle also drifts eastward (if negatively charged) or westward (if positively charged) due to asymmetries during the velocity reversal.

Our simulation shows the expected motion of the proton in the inner Van Allen belts with realistic simulation parameters.

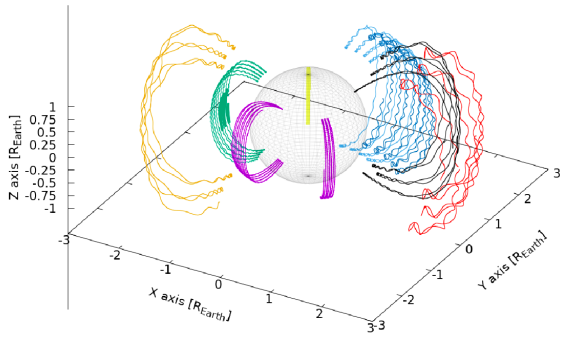

Figure 8: Trajectories of the simulated protons in the dipole field approximation of the Earth's magnetic field. A reference shadowed sphere of radius $R_\oplus$ is also displayed for reference.

4 Discussion

4.1 Overwiev

We have implemented and tested the relativistic Boris Algorithm, which allows the accurate simulation of the motion of charged particles in arbitrary EM field configurations even at relativistic speed. We used our implementation in different EM field configurations, demonstrating that our implementation (and the Boris Algorithm) can effectively be used as tools to understand the physical dynamics of relativistic charged particles. Our implementation is capable of handling simulations with the direct insertion of the values of the particles and fields in SI units, eliminating the need for the scaling of physical quantities.

4.2 Additional features explored during development

In order to speed the the simulations, especially when using high accuracy for extend periods of time for an high number of particles, we investigated the use of parallel computation through the use of OpenMP. We found (with no surprise) that the slowest part of the algorithm was the saving of the positions to file; thus the implementation of the parallel computation was useful only in specific use case and it was not adopted in the described computations.

In order to avoid this slowdown the position of the particles can be saved to file every $1 << N_{file} << N_{iters}$ time steps. This result in a dramatic improvement of simulation speed while improving accuracy (since a shorter time interval dt can be used) and saving disk space.

4.3 Future work

It would be interesting to extend the present work to allow the effects of the particle on the surrounding field. A wast literature is present on the argument and the resulting code would allow the numerical study of much more complex phenomena, such as plasma physics. By including also approximations of quantum effects, it should be possible to simulate the fusion of nuclei inside, as an example, fusion devices.

4.4 Source Code

The complete source code to run all the experiments described in this work can be found on GitHub at the link *https://github.com/lucabottero/ChargedParticleSimulator*.

A Derivation of the non-relativistic E × B drift velocity

Let us consider the motion of a non-relativistic ($\gamma \approx 1$) charged particle inside mutually perpendicular, constant and uniform electrical and magnetic field: $\mathbf{E} = (0, E_0, 0)$ and $\mathbf{B} = (0, 0, B_0)$, with $cE_0 \ll B_0$.

Qualitatively, we expect the particle's trajectory to be the superposition of a drifting motion, a parallel (to the magnetic field lines) motion and a gyration motion: $\mathbf{v} = \mathbf{v}_d + \mathbf{v}_{\parallel} + \mathbf{v}_g$.

Let the particle's initial velocity be $\mathbf{v}_0 = (v_0, 0, 0)$. If we find $\mathbf{v}_d$ such that $\mathbf{E}+\mathbf{v}_d \times \mathbf{B} = \mathbf{0}$, then the acceleration of Eq. (2) vanishes and the motion is cycloid. We have:

$$\mathbf{a} = \alpha(\mathbf{E} + \mathbf{v}_d \times \mathbf{B}) = \mathbf{0}$$

$$\mathbf{E} = -\mathbf{v}_d \times \mathbf{B}$$

Taking the cross product of both sides with the magnetic field we obtain

$$\mathbf{E} \times \mathbf{B} = -\mathbf{B}(\mathbf{B} \cdot \mathbf{v}_d) + \mathbf{v}B^2$$

where we used the well know triple product formula $(\mathbf{a} \times \mathbf{b}) \times \mathbf{c} = \mathbf{a} \times (\mathbf{b} \times \mathbf{c}) - \mathbf{b} \times (\mathbf{a} \times \mathbf{c})$ with $\mathbf{a} = \mathbf{v}_d$ and $\mathbf{b} = \mathbf{c} = \mathbf{B}$. Since the drifting speed is perpendicular to the magnetic field $\mathbf{v}_d \cdot \mathbf{B} = 0$ and we obtain Eq. (5)

$$\mathbf{v}_d = \frac{\mathbf{E} \times \mathbf{B}}{B^2}$$

It is interesting to note that the drifting velocity is independent of the particle's properties (charge and mass): in particular, the direction of motion will be the same for both negative and positive particles.

B Additional plots

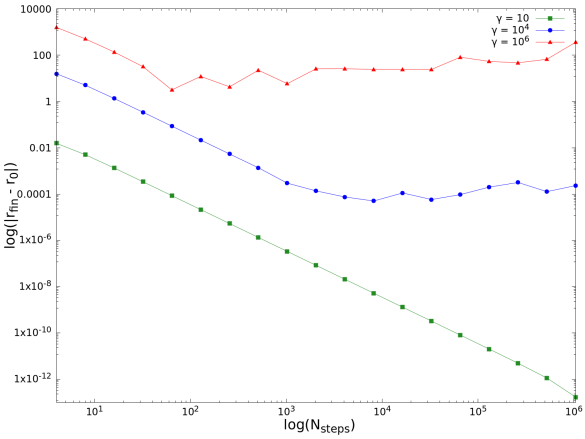

Figure 9: Log-log plot of the distance from starting point $|\mathbf{r}_{fin} - \mathbf{r}|$ after one turn of the particle.

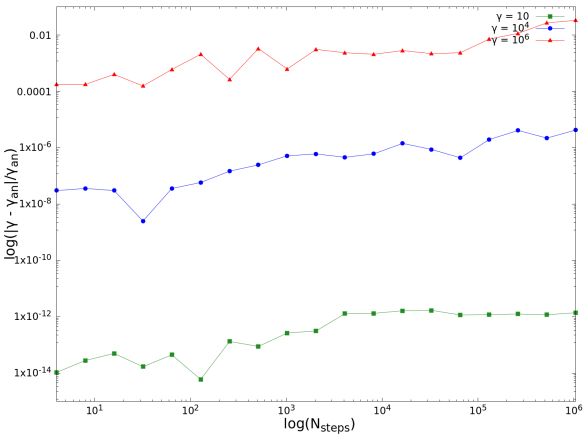

Figure 10: Log-log plot of $\log(\frac{|\gamma-\gamma_{an}|}{\gamma_{an}})$ after one turn of the particle. Here γ_{an} is the γ value at the beginning of the simulation.

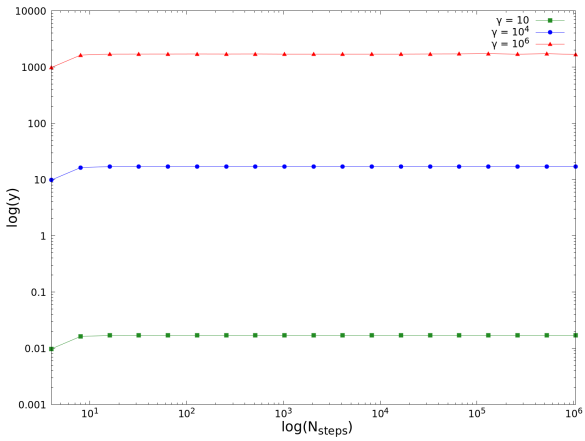

Figure 11: Log-log plot of the final horizontal displacement $x : fin$ after one turn of the particle.

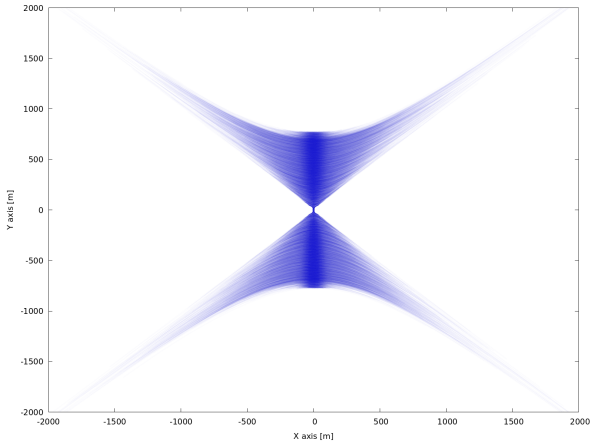

Figure 12: Trajectory densities for the simulation described in section 3.3 projected on the xy plane. Darker area means that more particle passed there during the simulation. The plot was generated as a superposition of all the particles trajectories plotted with an high transparency.

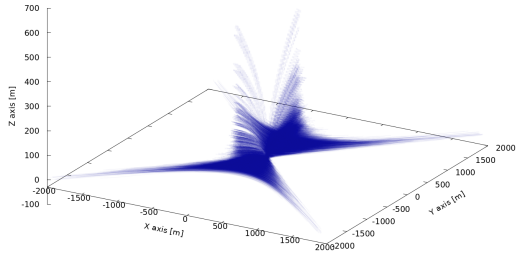

Figure 13: Trajectory densities for the simulation described in section 3.3 in full 3-dimensional space. Darker area means that more particle passed there during the simulation. The plot was generated as a superposition of all the particles trajectories plotted with an high transparency.

C Source code for Boris Algorithm Step

C.1 Simulation Structure

Each simulation needs 3 functions to be defined: the **E** and the **B** fields, both as function of position and time, and the initial condition for the particle. Those three functions are must be passed by reference to the Boris Algorithm Step function. This simple structure allows a great degree of customization with minimal effort for generic EM field configurations.

While theoretically capable of performing simulations of time-varying fields, we have not performed any experiment to validate this feature.

C.2 Boris Algorithm's step for one particle

```
void Boris2ndOrderStep(double t, double dt, double alpha, double * Y, int Neq,
                       void (*E_Field)(double, double *, double *), void (*
    B_Field)(double, double *, double *)){
    // Calculates one iteration of the Boris-push algorithm of the 2nd order for
     a single particle

    double x_half[3], u_minus[3], u_plus[3];
    double E[3], b[3];          // electrical field at n+1/2; scaled magnetic
    field

    double gamma = 1./sqrt(1. - (Y[3]*Y[3] + Y[4]*Y[4] + Y[5]*Y[5])/(C*C));
    // Lorentz factor

    // drift
    for(int i=0; i<3; i++) x_half[i] = Y[i] + 0.5*dt*Y[i+3];

    // Get fields values at x_half
    double pos[3] = {x_half[0], x_half[1], x_half[2]};
    E_Field(t + 0.5*dt, pos, E);
    B_Field(t + 0.5*dt, pos, b);

    // kick
    for(int i=0; i<3; i++) u_minus[i] = Y[i+3]*gamma + 0.5*alpha*dt*E[i];

    double gamma_minus = sqrt(1. + (u_minus[0]*u_minus[0] + u_minus[1]*u_minus
    [1] + u_minus[2]*u_minus[2])/(C*C));

    // Get the b field which is defined as h*B/(2*gamma)
    for(int i=0; i<3; i++) b[i] *= alpha*dt*0.5/gamma_minus;

    // rotate
    double u_minus_cross_b[3], fraction[3], rotation_cross[3];
    double correction_factor = 2. / (1. + b[0]*b[0] + b[1]*b[1] + b[2]*b[2]);
    cross_prod(u_minus, b, u_minus_cross_b);
    for(int i=0; i<3; i++) fraction[i] = (u_minus[i] + u_minus_cross_b[i]) *
    correction_factor;
    cross_prod(fraction, b, rotation_cross);
    for(int i=0; i<3; i++) u_plus[i] = u_minus[i] + rotation_cross[i];

    //gamma = 1./sqrt(1. - (Y[3]*Y[3] + Y[4]*Y[4] + Y[5]*Y[5])/(C*C));

    // kick
    for(int i=0; i<3; i++) Y[i+3] = u_plus[i] + 0.5*alpha*dt*E[i];

    double gamma_plus = sqrt(1. + (Y[3]*Y[3] + Y[4]*Y[4] + Y[5]*Y[5])/(C*C));
```

```
40
41      // Transform the relativistic corrected velocity back to velocity
42      for(int i=0; i<3; i++) Y[i+3] /= gamma_plus;
43
44      // drift
45      for(int i=0; i<3; i++) Y[i] = x_half[i] + 0.5*dt*Y[i+3];
46  }
```

C.3 General simulation code

```
1  int main(){
2      cout << setiosflags(ios::scientific) << setprecision(10);
3
4      double t;
5      double t_in = 0., t_fin = 3.5*8.19562e-11;
6      int n = 360;
7
8      double dt = (t_fin - t_in)/(double) n;
9
10     int N_PARTICLES = 1;
11     int Neq = 6 * N_PARTICLES;
12
13     double Y[Neq];
14
15     double alpha = -e/(m_e);
16
17     ofstream fdata;
18     fdata.precision(20);
19     fdata.open("output.dat");
20
21     t = t_in;
22     Set_Initial_Values(Y);
23
24     for (int i = 0; i < n; i++){
25         Boris2ndOrderStep(t, dt, alpha, Y, Neq, E_Field, B_Field);
26         t += dt;
27         fdata << t << " " << Y[0] << " " << Y[1] << " " << Y[2] <<  endl;
28     }
29
30     fdata.close();
31 }
```

C.4 E, B fields and initial conditions for the Dipole approximations of Earth's Magnetic Field

```
1  void Set_Initial_Values(double * Y){
2      double K = 100e6;           // 100 MeV
3      double M_eV = 938.272e6;    // Proton mass in eV
4
5      double theta_pos = (double)rand()/(double) RAND_MAX*2*M_PI;
6      double r = RADIUS_EARTH*(1.2 + 1.8*(double)rand()/(double)RAND_MAX); //
       Inner van Allen Belt
7
8      double theta_vel = (2. - 2.*(double)rand()/(double) RAND_MAX)*M_PI;
9      double phi_vel = (double)rand()/(double) RAND_MAX*2*M_PI;
10     K *= (0.8 + 0.7*(double)rand()/(double) RAND_MAX);      // Randomize kinetic
       energy of particle
```

```
    double speed = C*sqrt(1. - 1./((1 + K/(M_eV))*(1 + K/(M_eV))));

    Y[0] = r * cos(theta_pos);  // x
    Y[1] = r * sin(theta_pos);  // y
    Y[2] = 0. ;  // z
    Y[3] = speed * sin(theta_vel)*cos(phi_vel);  // v_x
    Y[4] = speed * sin(theta_vel)*cos(theta_vel);  // v_y
    Y[5] = speed * cos(theta_vel);  // v_z

    double B[3];
    double pos[3] = {Y[0], Y[1], Y[2]};
    B_Field(0., pos, B);
    r = sqrt(pos[0]*pos[0] + pos[1]*pos[1] + pos[2]*pos[2]);
}

void E_Field(double t, double * pos, double * E){
    E[0] = 0.;
    E[1] = 0.;
    E[2] = 0.;
}

void B_Field(double t, double * pos, double * B){
    double r = sqrt(pos[0]*pos[0] + pos[1]*pos[1] + pos[2]*pos[2]);
    double coupling_const = MAG_MOMENT/(r*r*r*r*r)*mu_0*0.25/M_PI;

    B[0] = 3. * pos[0] * pos[2] * coupling_const;
    B[1] = 3. * pos[1] * pos[2] * coupling_const;
    B[2] = 3. * (3.*pos[2]*pos[2] - r*r) * coupling_const;
}
```

C.5 Constants

```
# define C 299792458.
# define e 1.602176634e-19
# define m_e 9.1093837015e-31
# define m_p 1.67262192369e-27
```

References

[Bor+70] Jay P Boris et al. “Relativistic plasma simulation-optimization of a hybrid code”. In: *Proc. Fourth Conf. Num. Sim. Plasmas*. 1970, pp. 3–67.

[Tak02] Satoshi Takeuchi. “Relativistic $E \times B$ acceleration”. In: *Phys. Rev. E* 66 (3 Sept. 2002), p. 037402. DOI: 10.1103/PhysRevE.66.037402. URL: https://link.aps.org/doi/10.1103/PhysRevE.66.037402.

[Hon13] et al Hong Qin. “Why is Boris Algorithm So Good?” In: (Mar. 2013). DOI: 10.2172/1090047. URL: https://www.osti.gov/biblio/1090047.