Parallel Particle Dynamics with Spin for Plasma Applications

Afonso Santos, André Filipe, Rafael Fleming
afonso.carmo.santos@tecnico.ulisboa.pt, andre.g.filipe@tecnico.ulisboa.pt,
rafael.fleming@tecnico.ulisboa.pt

Supervised by
Profs. Marija Vranić and Bernardo Barbosa

Instituto Superior Técnico, Lisboa, Portugal

April 2025

Abstract

In this report, we present a C++ numerical framework (with some additional Python scripts) publicly available on GitHub at [1] that accurately solves particle dynamics, including spin, in the vicinity of different electromagnetic fields. We introduce the equations and forces that govern the system, the Lorentz force and BMT equation. We develop and showcase the algorithms used to simulate these systems in constant fields and laser pulses alongside theoretical predictions. We also explain numerical details, such as particle and field initialization, simulation outputs, and the parallel programming components of the script and its benefits for statistical and physical relevancy. Lastly, we extend our approach to include the Landau-Lifhitzs (LL) reduced formula for radiation reaction (RR) in our simulations, comparing it to analytical solutions.
Keywords: Lorentz Force, BMT Equation, Boris Pusher, Radiation Reaction, Parallel Programming

1. Introduction

The complexity of plasma physics problems has created a need for accurate numerical simulation and modeling of the behavior of charged particles in various electromagnetic field types. Numerical simulators allow for the study of problems such as the simulation of plasma-based particle acceleration in the strong-field regime using petawatt-class lasers [2, 3], or the ability to study and simulate astrophysical phenomena, like gamma-ray bursts [4] and supernova explosions [5].

The C++ routine developed in this work allows for a relativistic simulation of non-interacting charged particles in a variety of laser and electromagnetic fields and initial distributions. It relies on the use of a Boris pusher [6] algorithm that tracks the position, momentum and spin of the particles in the system, as well as the optional inclusion of radiation reaction.

The Lorentz force equation governs the movement of relativistic particles in an electromagnetic field. It is given by:

$$\frac{d\mathbf{u}}{dt} = \frac{q}{m}\left(\mathbf{E} + \mathbf{v} \times \mathbf{B}\right) \tag{1}$$

where $\mathbf{u} = \gamma\mathbf{v}$ is the generalized velocity of the particle, m is its mass, and $\gamma = 1/\sqrt{1 - v^2/c^2} = \sqrt{1 + u^2/c^2}$ its Lorentz factor. This equation is the basis for the momentum part of the pusher, whose implementation will be discussed in Section 2.

As for the description of the spin motion of the particles we must include the possibility of having ultra-relativistic particles with $\gamma \gg 1$ and so we cannot restrict ourselves to the non-relativistic equation of spin precession given by $\frac{d\mathbf{s}}{dt} = \boldsymbol{\mu} \times \mathbf{B}$. Because of this we use the Bargmann-Michel-Telegdi (BMT) equation, which describes the spin precession due to the interaction of the magnetic moment of a relativistic particle with the EM field, given by:

$$\frac{d\mathbf{s}}{dt} = -\left[\left(a + \frac{1}{\gamma}\right)(\mathbf{B} - \mathbf{v} \times \mathbf{E}) - \mathbf{v}\frac{a\gamma}{\gamma + 1}(\mathbf{v} \cdot \mathbf{B})\right] \times \mathbf{s} = \boldsymbol{\Omega} \times \mathbf{s} \tag{2}$$

where $\mathbf{\Omega}$ is the precession frequency, $a = (g-2)/2$ is the anomalous magnetic moment of the electron, and $g \simeq 2.0023228$ is its intrinsic momentum.

This equation is the basis for the spin component of our pusher algorithm and will be explained in depth in Section 4.

Lastly, given the use of strong fields in our simulation, the motion of charged relativistic particles will be significantly impacted by the radiation reaction (RR) force and its accompanying energy loss. Therefore, we consider an added term to our pusher, which will be optional and controlled by the user, based on the Landau-Lifhitzs (LL) reduced formula for RR force equation [7]:

$$\mathbf{f}_{RR}(\mathbf{u}) = \sigma_0 \frac{q^2}{\gamma m} \left[F^2 u - (u|F^2 u)u\right]_{spatial} \tag{3}$$

Note that the quantities in this function are expressed in a four-vector notation and are defined in a Minkowski metric $g_{\mu\nu} \equiv G = \text{diag}(1,-1,-1,-1)$. Here $\sigma_0 = \frac{2e^2\omega_0}{3m_e c^2}$ is a dimensionless parameter that depends on the laser frequency, ω_0. We also have that $(\cdot|\cdot)$ is the four-vector inner product defined as $(U|V) = U^T G V$ in matrix form, and $F^\mu_{\ \nu} \equiv F$ is the EM field tensor defined as

$$F = \begin{pmatrix} 0 & E_1 & E_2 & E_3 \\ E_1 & 0 & B_3 & -B_2 \\ E_2 & -B_3 & 0 & B1 \\ E_3 & B_2 & -B1 & 0 \end{pmatrix} \tag{4}$$

The specifics of the implementation of the RR force into the pusher will be discussed in Section 5.

The study of complex plasma problems presents some challenges. Many cases of interest cannot be studied through analytical methods alone, as exact solutions may not exist. This, along with the microscopic nature and need for a statistical study of these problems, calls for numerical simulations, which can model the interactions between particles and fields. Additionally, the complexity of these systems, which deal with a large number of particles and variables, demands significant computational resources. To ensure the ability to scale the system and take advantage of the fact that we assume non-interacting particles, we provide an optional parallel computing architecture of single instruction with multiple data (SIMD). This allows for the simultaneous processing of multiple particle calculations, reducing the total simulation time. The numerical foundations of this project, along with the integration of parallel programming, are detailed in Section 3.

2. The Boris Pusher

As a way to numerically solve the motion of a system of particles subject to an EM field, we use the Boris pusher method. This method is a classic second-order leap-frog scheme that advances the position and momentum of a particle in time by discretizing the Lorentz force equation (1) in time. For our scheme we use an "unsynchronized" version of the leap-frog scheme, where the momentum and position are defined with half a time-step difference. *Ripperda et. al.* [8] and *Birdsall and Langdon* [9] give a full and comprehensive description of the Boris scheme and the leap-frog method used. The Boris scheme pushes the reduced momentum through one time step in the following way:

1. First half electric field acceleration:

$$\mathbf{u}^- = \mathbf{u}^n + \frac{q\Delta t}{2m}\mathbf{E}\left(\mathbf{x}^{n+1/2}\right) \tag{5}$$

2. Magnetic field rotation (conserves reduced momentum magnitude, as magnetic field does no work):

$$\mathbf{u}^+ = \mathbf{u}^- + (\mathbf{u}^- + (\mathbf{u}^- \times \mathbf{t})) \times \mathbf{s} \tag{6}$$

3. Second half electric field acceleration:

$$\mathbf{u}^{n+1} = \mathbf{u}^+ + \frac{q\Delta t}{2m}\mathbf{E}\left(\mathbf{x}^{n+1/2}\right) \tag{7}$$

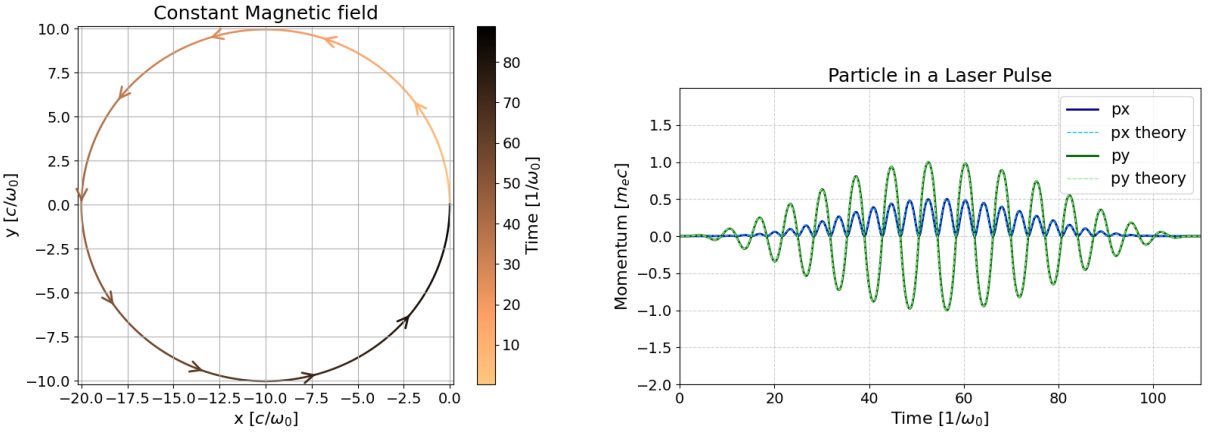
Where $\mathbf{t} = \mathbf{B}(\mathbf{x}^{n+1/2})q\Delta t/(2m\gamma^-)$, $\mathbf{s} = 2\mathbf{t}/(1+t^2)$, $\gamma^\pm = \sqrt{1+(u^\pm/c)^2}$ and Δt is the chosen time step. The position is then updated through the new reduced momentum:

$$\mathbf{x}^{n+\frac{3}{2}} = \mathbf{x}^{n+\frac{1}{2}} + \frac{\Delta t}{\gamma^{n+1} m}\mathbf{u}^{n+1} \tag{8}$$

It may appear somewhat convoluted to divide the schemes into these steps and to handle the magnetic field rotations with equation 6, but it can be shown [8] that the Boris scheme conserves the particle's energy for vanishing electric fields. It does not do so for a non-zero electric field, but it can still be highly accurate in this case, hence our choice to make use of it in our implementation.

2.1. Algorithm Validation

The theoretical solutions for particle dynamics in constant EM fields are well known. For example, in a constant magnetic field, with no electric field, the particle should undergo cyclotron motion, meaning its trajectory draws a circle with the Larmor radius, $r_L = \frac{p_\perp}{eB}$, and angular frequency $\omega_c = eB/m_e$. This is precisely the setup of figure 1a, which has $\mathbf{B} = 0.1 m_e \omega_0 / c \; \boldsymbol{e}_z$ and the particle is initialized with momentum $\mathbf{p}_0 = m_e c \; \boldsymbol{e}_y$, meaning its trajectory radius should be $r_L = 10c/\omega_0$, as obtained numerically. Using this method, we were also able to reproduce other known constant EM field phenomena such as the $\mathbf{E} \times \mathbf{B}$ drift.



(a) **(b)**

Figure 1: (**a**) cyclotron motion of an electron in a constant magnetic field obtained numerically from the Boris pusher algorithm ($\mathbf{B} = 0.1 m_e \omega_0 / e \; \boldsymbol{e}_z$ and $\mathbf{p}_0 = m_e c \; \boldsymbol{e}_y$); (**b**) x and y components of the momentum as a function of time for values obtained numerically (solid lines) compared to what is expected theoretically (dashed lines), for a laser pulse traveling in the x direction with length $L = 100$ and $E_0 = m_e c \omega_0 / e$ (with electric field polarized in the y direction). The simulated and analytical results show perfect agreement.

Our main objective, however, is to simulate particles in EM fields like those present in laser pulses. These systems also have well-studied solutions [10], which dictate that the particle's momentum should oscillate following the envelope of the pulse. Once again, figure 1b shows such a system: a laser pulse traveling in the x direction with the electric field linearly polarized in the y direction encounters a particle at rest. The image showcases both obtained time evolution of the particle's momentum in the x and y direction (p_z remained null for the entire simulation) both for the simulated and analytical results, which precisely overlap. Our scheme agrees perfectly with the theory.

However, this last simulation raises another question: What is the form of this laser pulse, and how were we able to initialize it? Additionally, we want to expand this scheme to efficiently simulate a large number of particles at a time. How can we do this? Before exploring more complex particle dynamics, it is important to visit the numerical foundations of our project.

3. Numerical Foundations of the Project

One of our key objectives for this framework was for it to handle a large sample of different systems and initial configurations. For this purpose, the script runs on an input deck with several options to choose from. These span the particles' position, momentum, and spin initialization, electromagnetic field configurations, diagnostics, and simulation parameters.

Another main objective for this work was statistical significance. For this, we need to be able to solve systems with a large number of particles, which comes at a great computational cost. We made use of parallel programming to overcome this issue and run large simulations in relatively unimpressive machines in a reasonable run time.

In this section, we will explain how each of these objectives was achieved. Further explanation on the usage of the code can be found in the README file at [1].

3.1. Particle initializations

In our system, each particle has eight degrees of freedom: position, momentum, and normalized spin. Therefore, besides the number of particles, one must also indicate the distribution functions for these quantities.

For momentum and position, we account for three distribution types:

1. Single preferred value for all particles, given in the input;
2. Uniform distribution with given width;
3. Normal distribution with given central value and spread.

The particles' spins are handled similarly, with the single caveat being that the spins must be on the unit sphere. Once again, we have single direction and uniform distributions on the sphere. However, instead of a normal distribution, we include a Von Mises-Fisher distribution, which, for the unit sphere, is given by:

$$f(\mathbf{s};\boldsymbol{\mu},\kappa) = \frac{\kappa}{4\pi\sinh(\kappa)}\exp(\kappa\boldsymbol{\mu}\cdot\mathbf{s}) \tag{9}$$

where $\boldsymbol{\mu}$ is the central spin direction, and κ controls the spread of the distribution. This distribution emulates an intermediate polarization regime, where particles are not perfectly aligned but have a preferred spin direction. It was implemented using a Monte Carlo rejection sampling method. Figure 2 shows this method's resulting initial spin configuration.

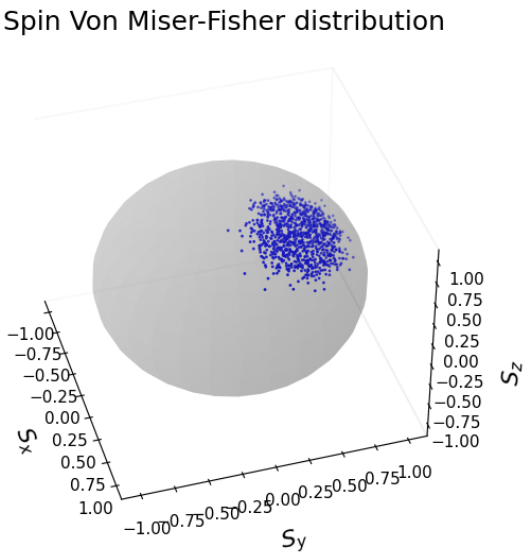


Figure 2: Initial spin configuration of 1000 particles illustrating the Von Miser-Fisher distribution in a simulation, for $\kappa = 50$ and $\boldsymbol{\mu} = (1,1,2)/\sqrt{6}$

3.2. Field Initialization

To handle a great variety of systems, we opted to have a lot of field configurations available. These are initialized as "lasers", which can be of four different types:

1. Constant electric and magnetic fields with chosen values and directions;
2. Spatially constant electric and magnetic fields with chosen maximum values and directions, whose magnitude oscillates with time;
3. Plane waves, initialized with a chosen electric field and wave vector;
4. Laser pulse with a built-in envelope, where the envelope length, L, electric field, and wave vector are provided by the user.

The chosen envelope for the last option can be easily changed within the code. For this project, we chose an envelope of the form:

$$E_{\text{Env}}(\phi) = \begin{cases} E_0 \sin^2\left(\frac{\phi\pi}{L}\right) & \text{if} \quad 0 < \phi < L \\ 0 & \text{otherwise} \end{cases}, \qquad \text{where } \phi = \boldsymbol{k}\cdot\boldsymbol{r} - \omega t \tag{10}$$

where $\boldsymbol{k}$ and ω are the laser's wave vector and frequency. The user does not need to initialize the laser's frequency due to our choice of units in this work, as we will see in the next subsection.

Lastly, for any single simulation, any number of lasers. There is also a possibility to have an extra phase of $\frac{\pi}{2}$ in a laser to recreate circular polarization.

3.3. Workspace Units

To do numerical physics, we must choose the units we will work with throughout our simulations. In our case, it is only natural to normalize most quantities to those of the particles and lasers we are simulating. Our time scales are given by the inverse of the laser frequency, ω_0, mass and charge are normalized to the electron's mass and charge magnitude, m_e and e, velocities to the speed of light, c, etc. This way, the same simulation results can be used for systems with different laser frequencies.

Table 1 showcases the units of some relevant physical quantities are measured within this system:

Quantity	Units
Generalized velocity	$[\mathbf{u}] = c$
Linear Momentum	$[\mathbf{p}] = m_e c$
Distance	$[\Delta x] = c/\omega_0$
Number Density	$[n] = (\omega_0/c)^3$
Energy	$[U] = m_e c^2$
Electric Field	$[\mathbf{E}] = m_e c\omega_0/e$
Magnetic Field	$[\mathbf{B}] = m_e\omega_0/e$
Vector Potential	$[\mathbf{A}] = m_e c/e$

Table 1: Workspace units for several relevant physical quantities.

The only exception is spin, which we treat as an adimensional quantity. This is because we are only interested in the direction of the vector, not its magnitude. All spin vectors are normalized to have a unit norm, $||\mathbf{s}|| = 1$.

Lastly, the particle's linear momentum and generalized velocity differ only by its mass. Since we only worked with electrons throughout this project, we normalized our units to their mass, these two quantities are essentially the same inside our framework. Hence, throughout this report, we use these terms interchangeably.

3.4. Diagnostics

After running the simulation, our tool provides several options for diagnostics, which can be chosen and personalized in the input file.

For example, in simulations involving a large number of particles, one can opt to visualize the distribution using one or two-dimensional histograms of selected components from the particle's full nine-dimensional phase space position. Which components (of either $\mathbf{x}$, $\mathbf{p}$ or $\mathbf{s}$) are chosen, as well as the number and size of the histogram's bins, are set in the input file. Each simulation produces histograms of the same selected components at different moments inside the simulation. The number of iterations between histograms is also selected by the user.

Another option provided is to retain the particle's position, momentum, and spin evolution in a single file to analyze its trajectory in phase space. For this purpose, the program may follow a selectable number of randomly chosen particles or a single chosen particle.

Lastly, it is also possible to produce a simple histogram that dictates the chosen electric or magnetic field values throughout time and space. We found that this facilitates the physical interpretation of the simulation results. The field component, bin size and limits, and time evolution can all be chosen in the input file.

To visualize the outputs from the simulation, our framework also includes a couple of Python scripts. These handle reading the output and input files and plotting the simulation data. A lot of plotting options are provided, from particle trajectories to histograms. All figures in this report were made using the functions from these scripts.

3.5. Code Parallelization

To support users interested in high-performance simulations, one of the branches of the numerical framework provided in [1] is fully parallelized using the OpenMPI C++ library [11]. This functionality is kept separate from the main branch to avoid imposing the OpenMPI dependency on users who do not require parallel execution.

Adapting the main branch's code to parallel programming is simple since we are only considering non-interacting particles. We can give each process different particles so that each one handles different calculations with no need to communicate with the rest until it's time to write diagnostics. Essentially, the program follows these steps

1. The program initializes the particles of the system according to the chosen initial configuration;
2. The particles are divided evenly across the various processes, with the last process taking the remainder particles;
3. Each process runs the simulation for its particles until it is time to write diagnostics;
4. The processes communicate to write the histogram of the ensemble of particles at that instant;
5. Repeat steps 3 and 4 until the simulation is complete.

To evaluate the performance of this parallelization scheme, we ran the same simulations with a different number of processors and compared the time they took to run. Results can be seen in 3, where the speedup for N processors is given by the ratio between the total time it took to run the simulation for one and N processors.

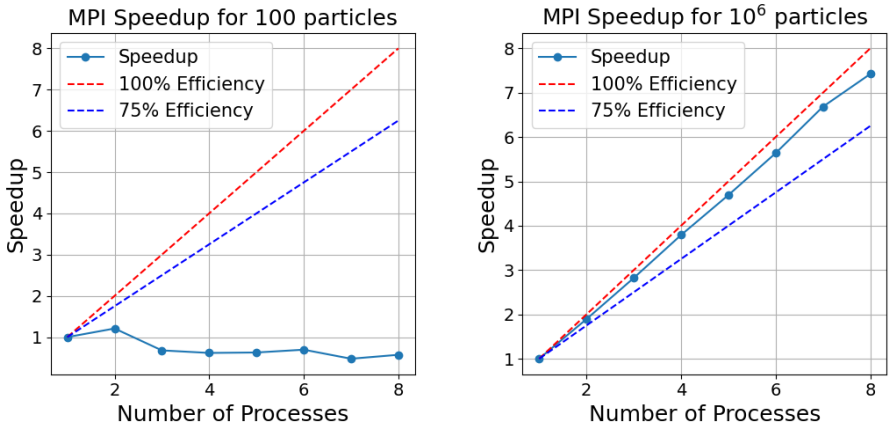


Figure 3: MPI parallelization speedup compared to 75% and 100% efficiencies, for 100 particles [left] and for 10^6 particles [right]. The achieved efficiency for this second case was of $\approx 94.94\%$. Code parallelization seems to only be beneficial for larger numbers of particles.

As expected, our parallel programming scheme can never achieve 100% efficiency [12], meaning the workload is perfectly divided between the processors. This is because of the non-parallelized portions of the code, such as particle initialization and histogram writing. Even still, the case of 10^6 particles on the right of figure 3 showcases good efficiency ($\approx 94.94\%$). For a lower number of particles, however, most of the computing time is spent on initializing and dividing the particles between processors instead of the parallelized calculations. This means that increasing the number of processors can slow down the simulation, as the left plot of figure 3 illustrates. Using the parallelized branch of our code is only beneficial to simulate systems with large amounts of particles.

4. The Spin Dynamics Solver

Now that the statistical foundation of our approach is well established, we can begin to explore the spin dynamics of our particles. For this purpose, we will treat each particle as having a well-defined spin vector with no quantum uncertainties. In reality, we must interpret this as an expected value for the particle's spin vector after repeating the simulation multiple times, $\langle \boldsymbol{s} \rangle = \boldsymbol{s}$. Through this approach, the equation that governs spin dynamics is the BMT equation [13].

4.1. the Bargmann-Michel-Telegdi (BMT) Equation

The covariant form of the BMT equation is written as [7]:

$$\frac{ds^\mu}{d\tau} = \frac{q}{m}\left[\frac{g}{2}F^\mu{}_\nu s^\nu - a(u_i F^i{}_j s^j)u^\mu\right] \tag{11}$$

where latin indices (i and j) only sum through spatial components, we have taken $c = 1$, $F^{\mu\nu}$ is the Faraday tensor and s^μ is the spin 4-vector, defined as:

$$s^\mu = \left(\mathbf{s}\cdot\mathbf{u},\, m\mathbf{s} + \frac{\mathbf{s}\cdot\mathbf{u}}{\gamma+1}\mathbf{u}\right) \tag{12}$$

However, we are only interested in the vector $\mathbf{s}$, whose equation of motion can be recovered from the two above. This yields equation 2, presented above, which can be written in terms of generalized velocity $\mathbf{u}$ as:

$$\frac{d\mathbf{s}}{dt} = \mathbf{s}\times\left[\left(a+\frac{1}{\gamma}\right)(\gamma\mathbf{B} - \mathbf{u}\times\mathbf{E}) - \mathbf{u}\frac{a}{\gamma+1}(\mathbf{u}\cdot\mathbf{B})\right]/\gamma = \mathbf{s}\times\boldsymbol{\Omega} \tag{13}$$

The form of the equation makes it clear that $\mathbf{s}$ is only getting rotated at each time, according to some axis and angular frequency given by $\boldsymbol{\Omega}$. The magnitude of the spin vector is conserved as expected. The complexity in this equation is hidden inside $\boldsymbol{\Omega}$, which depends non-trivially on the particle's momentum and the EM fields.

For constant magnetic fields, this equation describes spin precession, as any quantum formalism predicts. However, the usefulness of equation 13 becomes clear once we attempt to tackle complex electrodynamic field configurations. Once again, besides constant EM fields, there are known analytical solutions for plane waves and laser pulse configurations. These are complex expressions, written in terms of the vector potential, $\mathbf{A}$, and the particle's momentum [14].

In this report, however, we will explore these equations numerically.

4.2. Solving the BMT Equation Numerically

Considering that at any given instance we know the EM field configuration of a system and the particle's momentum, $\boldsymbol{\Omega}$ is just some known vector field. Thus, equation 13 has the same form as the Lorentz force (1) if we set $\mathbf{E} = \mathbf{0}$, and make the change: $\mathbf{u} \to \mathbf{s}$, $\mathbf{B} \to \Omega$ (apart from some multiplicative constants). This means that we can reuse the Boris pusher algorithm and even ignore the first and last steps, which account for the change in the magnitude of the vector.

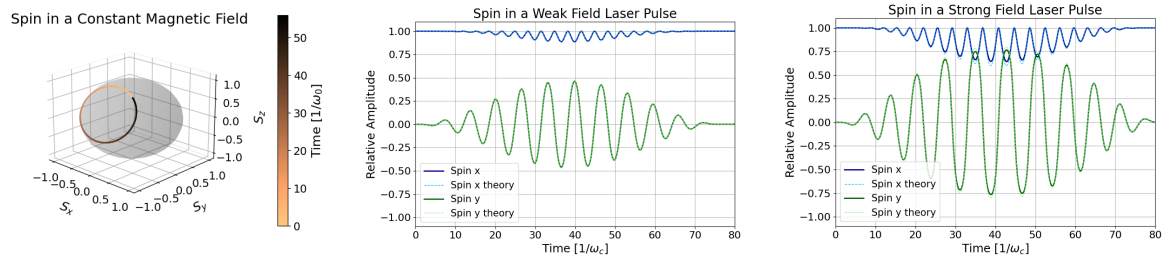
Essentially, after the Boris algorithm described in section 2, we update the particle's spin by:

$$\mathbf{s}^{n+1} = \mathbf{s}^n + (\mathbf{s}^n + \mathbf{s}^n\times(\boldsymbol{\Omega}\Delta t/2))\times\frac{\boldsymbol{\Omega}\Delta t}{1+(\boldsymbol{\Omega}\Delta t/2)^2} \tag{14}$$

where $\boldsymbol{\Omega}$ is calculated using $\mathbf{E}$, $\mathbf{B}$ and $\mathbf{u}$ at time step $n+1$.

4.3. Algorithm Validation

Finally, from the theoretical solutions arrived at by *Greger Torgrimsson* [14], we can make a comparisson to the results from our simulations for several systems. Figure 4 shows three such systems.



(a) **(b)** **(c)**

Figure 4: Three spin dynamics simulation results for initially at-rest particles. (**a**) Three-dimensional plot of spin precession for a constant magnetic field $\mathbf{B} = (0, -0.1, 0.05)m_e\omega_0/e$ and initial spin $\mathbf{s}_0 = (1, -1, 2)/\sqrt{6}$. (**b**), (**c**) Time evolution of spin for a strong and weak ($E_0 = 0.5m_e c\omega_0/e$ and $E_0 = m_e c\omega_0/e$, respectively) laser pulse with length $L = 75$ traveling in the x direction (aligned with initial spin) with $\mathbf{E}$ in the y direction. Simulation results have good agreement with theory, but there is some discrepancy, especially noticeable for the stronger field.

The constant magnetic field simulation (Figure 4a) showcases the predicted spin precession. The direction of spin rotates on the plane perpendicular to the magnetic field, never changing magnitude, as expected.

As for the laser pulse simulations (Figures 4b and 4c), we chose to portray time evolution plots of the first components of $\mathbf{s}$. This makes the comparison with theory clearer and hides s_z, which remained at zero throughout the simulation. While the figure demonstrates good agreement between theory and simulation, we do have some discrepancy for stronger fields. It seems that in these cases, theory expects slightly larger peak spin direction changes than our simulations can offer. This may be due to some small inconsistency in our scheme or a slight misinterpretation of the analytical solutions.

One last important detail regarding spin dynamics that we found for the regimes we explored was the relative unimportance of EM fields. Both the analytical and simulation results seem to depend mainly on the particle's momentum. Of course, the momentum itself is completely determined by the fields, so unimportance is somewhat misleading. What we observed is that, whether we considered the fields direct contributions to the spin in the theoretical calculations or not, we obtain essentially the same analytical solution. In fact, even considering the electron's anomalous magnetic momentum to be null seemed to have little effect in either theory or simulation.

5. Introducing Radiation Reaction

With a working pusher for momentum, position, and spin, we can now set out to include the effects of radiation reaction (RR) to our system. This would allow us to emulate systems within very high-energy regimes.

5.1. Numerical Radiation Reaction Contribution

The two possible places where a RR contribution could be added to out pusher are as a half-push with the electric field acceleration in equations 5 and 7, or with the magnetic field rotation term in equation 6 as a "full push". As the full weight of the RR term over a single time-step would consider only the contribution of an intermediate momentum value ($\mathbf{u}^-$), a more realistic approach would be to spread it out in two separate momentum time steps ($\mathbf{u}^-$ and $\mathbf{u}^{n+1}$), before and after the magnetic field rotation, i.e. in the electric field acceleration terms. The addition of the RR terms to equations 5 and 7 are as follows:

$$\mathbf{u}^- = \mathbf{u}^n + \frac{q\Delta t}{2m}\mathbf{E}\left(\mathbf{x}^{n+1/2}\right) + \frac{\Delta t}{2}\mathbf{f}_{RR}(\mathbf{u}^n), \tag{15}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^+ + \frac{q\Delta t}{2m}\mathbf{E}\left(\mathbf{x}^{n+1/2}\right) + \frac{\Delta t}{2}\mathbf{f}_{RR}(\mathbf{u}^+) \tag{16}$$

In order to process the LL equation (eq. (4)) into the framework we expressed it in vector form as follows:

$$(\mathbf{f}_{RR})_i(\mathbf{u}) = \sigma_0 \frac{q^2}{\gamma m}\left[(F^2\mathbf{u})_i - \left(\gamma((\mathbf{B}\times\mathbf{E})\cdot\mathbf{u}) + \mathbf{E}^2\gamma^2 - u_j(F^2\mathbf{u})_j\right)u_i\right] \tag{17}$$

Where $(F^2\mathbf{u})_j$ is defined as:

$$(F^2\mathbf{u})_j = \gamma(\mathbf{E}\times\mathbf{B})_j + E_j(\mathbf{E}\cdot\mathbf{u}) + B_j(\mathbf{B}\cdot\mathbf{u}) - \mathbf{B}^2 u_j \tag{18}$$

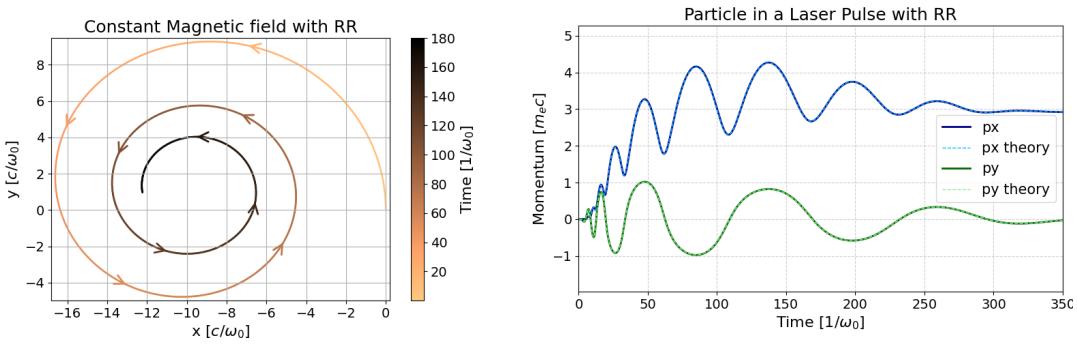
The effects of this implementation to the circular motion of an electron using a linearly polarized laser pulse with a constant magnetic field can be seen in Figure 5a (the same setup as in figure 1a). Instead of simple cyclotron motion, we now observe that the particle spirals inward instead. This happens because the particle is emitting radiation at the cost of its energy, which means its momentum diminishes at every time step, and thus so does its Larmor radius, creating a spiral instead of the circle we had in figure 1a.

5.2. Algorithm Validation

To compare the numerical results with theory, we use the reduced Landau-Lifshitz model to calculate particle momentum considering RR, where we neglect forces that depend on field derivatives. A paper by *A. Di Piazza* [15] gives a complete formulation of the theory used for particle dynamics in plane waves according to the LL model.

Figure 5b shows perfect agreement between numerical simulation and theory. Here, we consider an initially at-rest particle and a laser pulse traveling in the x direction with length $L = 40$, electric field in the y direction, and $E_0 = m_e c\omega_0/e$. Particles gain momentum from the laser pulse and seem to remain inside it for longer then when RR is neglected (figure 1b). We can also observe an elongated attenuation

of the momentum which does not even converge to zero, as opposed to the "symmetric" loss of momentum in the case of no RR.



(a) **(b)**

Figure 5: (**a**) Cyclotron-like motion of an electron in a constant magnetic field obtained numerically from the Boris pusher algorithm with the effects of radiation reaction ($\mathbf{B} = 0.1 m_e \omega_0 / e \; \boldsymbol{e}_z$ and $\mathbf{p}_0 = m_e c \; \boldsymbol{e}_y$). These effects take energy from the particle, diminishing the Larmor radius of the movement at every instance; (**b**) Comparison between theoretical (dashed lines) and simulated (solid lines) values of p_x and p_y, for a particle in a laser pulse that travels in the x direction with the electric field initialized in the y direction ($L = 40$ and $E_0 = m_e c \omega_0 / e$). There is perfect agreement with theory, and it can be seen just how much these results differ from the systems with no radiation cooling (Figure 1b).

Additionally, since there is no radiation cooling correction term in the BMT equation, the analytical solutions for the spin dynamics in this regime are the same as in section 4, except that the momentum used to calculate them is altered by the RR terms, as already discussed. When we compared these solutions to the numerical results, we found discrepancies in the peak changes of spin components as in figure 4c, but significantly more pronounced. These discrepancies likely stem from the same unresolved issue as the ones from section 4 and are thus not analyzed further.

6. Conclusions

The implementation of an algorithm for the simulation of charged particle and spin dynamics in different types of electromagnetic fields was successfully achieved and can be accessed on GitHub at [1]. The algorithm is based on a Boris pusher leap-frog method to advance particle momentum and position, and the spin dynamics are included using the BMT equation. The effects of radiation reaction are also successfully taken into account. The simulation initializes the momentum, position, and spin of particles with three different distributions and follows their evolution through time. We also created 4 different EM field types, ranging from constant fields to laser pulses. The framework also provides several highly customizable plotting options for data visualization through Python scripts. Additionally, in a separate branch of the numerical framework, the evolution scheme was parallelized using the OpenMPI C++ library, allowing for high-performance computing. A speedup efficiency of 94.94% was achieved for 10^6 particles.

Theoretical validation of the numerical simulations was successful for most explored cases. For the momentum and position portion of the pusher, we observe exactly the expected cyclotron motion for a constant magnetic field, and the particle's momentum follows the envelope of the laser pulse also as expected (Figures 1a and 1b). For the spin implementation, we successfully observed the spin precession in a constant magnetic field as well as the expected spin amplitudes, with small disagreement with theory for strong field laser pulses (Figures 4a, 4b and 4c). Lastly, with the implementation of radiation reaction, we observed the expected diminishing of the Larmor radius and the elongated attenuation of particle momentum that is predicted from the loss of energy due to electromagnetic radiation emission (Figures 5a and 5b).

The developed code can work as a benchmark for more complex systems since it is simpler and easier to operate than other such codes, like PIC codes, for example. Having this in mind and thinking of future work, the project has room for several improvements, some of which are simple and others more complex. Simple improvements include: The integration of other types of laser envelopes; Using the vector potential $\mathbf{A}$ rather than the electric field $\mathbf{E}$ for ease of comparison with theoretical results; The addition of the effects of the Stern-Gerlacht force. Other, more challenging implementations include: Introducing particle interactions; Including other types of relativistic particle pusher integrators, such as the Vay or the Higura-Cary schemes [8], which have different accuracies depending on the type of calculation performed.

References

[1] André Filipe Rafael Fleming, Afonso Santos. Parallel particle dynamics with spin. https://github.com/figipef/Parallel-spin-particle-dynamics/tree/main, 2025. Last updated: 2025-04-02.

[2] Toshiki Tajima and John M Dawson. Laser electron accelerator. *Physical review letters*, 43(4):267, 1979.

[3] Pisin Chen, J. M. Dawson, Robert W. Huff, and T. Katsouleas. Acceleration of electrons by the interaction of a bunched electron beam with a plasma. *Phys. Rev. Lett.*, 54:693–696, Feb 1985.

[4] S Chintalwad, S Krishnamurthy, S Morris, Lap Van Dao, and B Ramakrishna. Simulation studies of γ-ray radiation in laser-plasma interactions with structured targets. *Fundamental Plasma Physics*, 10:100038, 2024.

[5] Martin Pohl, Masahiro Hoshino, and Jacek Niemiec. Pic simulation methods for cosmic radiation and plasma instabilities. *Progress in particle and nuclear physics*, 111:103751, 2020.

[6] JP Boris. Relativistic plasma simulation-optimization of a hybrid code, paper presented at fourth conference on numerical simulation of plasmas, naval res. *Lab., Washington, DC*, 1970.

[7] Fei Li, Viktor K. Decyk, Kyle G. Miller, Adam Tableman, Frank S. Tsung, Marija Vranic, Ricardo A. Fonseca, and Warren B. Mori. Accurately simulating nine-dimensional phase space of relativistic particles in strong fields. *Journal of Computational Physics*, 438:110367, August 2021.

[8] B. Ripperda, F. Bacchini, J. Teunissen, C. Xia, O. Porth, L. Sironi, G. Lapenta, and R. Keppens. A comprehensive comparison of relativistic particle integrators. *The Astrophysical Journal Supplement Series*, 235(1):21, March 2018.

[9] Charles K Birdsall and A Bruce Langdon. *Plasma physics via computer simulation*. CRC press, 2018.

[10] Paul Gibbon. *Short Pulse Laser Interactions with Matter: An Introduction*, chapter 3.1, pages 30–37. Imperial College Press, London, 2005.

[11] Edgar Gabriel, Graham E. Fagg, George Bosilca, Thara Angskun, Jack J. Dongarra, Jeffrey M. Squyres, Vishal Sahay, Prabhanjan Kambadur, Brian Barrett, Andrew Lumsdaine, Ralph H. Castain, David J. Daniel, Richard L. Graham, and Timothy S. Woodall. Open MPI: A high-performance, portable implementation of the MPI standard. https://www.open-mpi.org/, 2024.

[12] *AFIPS '67 (Spring): Proceedings of the April 18-20, 1967, spring joint computer conference*, New York, NY, USA, 1967. Association for Computing Machinery.

[13] Timofey Zolkin. Bmt equation analysis and spin in accelerator physics. 2011.

[14] Greger Torgrimsson. Loops and polarization in strong-field qed. *New Journal of Physics*, 23(6):065001, may 2021.

[15] A. Di Piazza. Exact solution of the landau-lifshitz equation in a plane wave, 2008.