Particle Tracing in Earth's, Saturn's and Jupiter's Magnetospheres. Mat-lab App

Jaime Ruiz Zapatero
Supervisor: Patrick Guio

June 27, 2018

University College London. Physics and Astronomy.

Contents

1 Introduction **1**

2 Physics background **2**
2.1 Single Particle Motion in a constant magnetic field . . . 2
2.1.1 Boris integrator . . . 3
2.2 Single Particle Motion in a Magnetic Dipole . . . 4
2.3 Adiabatic Invariants . . . 4
2.3.1 The loss cone . . . 6
2.4 Magnetodisc structures . . . 7
2.4.1 Caudal's formalism . . . 7

3 App Instruction set **8**
3.1 How to launch the app . . . 8
3.2 How to interact with the app . . . 9
3.2.1 Initial conditions . . . 9
3.2.2 Simulation Configuration . . . 9
3.2.3 Authorship . . . 9
3.3 Understading the graphical output of the app . . . 9
3.3.1 3D Animation . . . 9
3.3.2 2D Comparison . . . 10
3.3.3 Energy Comparison . . . 11
3.3.4 Latitude Comparison . . . 12
3.4 How to change the app's code . . . 12
3.5 Understading the code behind the app . . . 13
3.5.1 Initial Parameters . . . 13
3.5.2 Tracing . . . 14
3.5.3 Plotting . . . 14
3.5.4 Auxiliary functions . . . 14
3.5.5 Jupiter and Saturn modifications . . . 14

4 Acknowledgements **14**

5 References **14**

1 Introduction

The aim of this App is to allow the potential users to gain an understanding of the motion of charged particles in different planetary magnetic structures by providing an interactive graphical representation of the physical theory. The presented app is an adaptation of the code provided by

Professor P. Guio to model the behaviour of charged particles in the magnetospheres of Jupiter, Saturn and Earth that allows the user to explore the variations of the simulation by changing the initial kinetic of the particle, its pitch angle and the length of the simulation. The simulation also offers a comparison between the results obtained when using a Boris algorithm to emulate the particle's trajectory and the solver ode15s built in Mat-lab R2017a.

2 Physics background

2.1 Single Particle Motion in a constant magnetic field

To understand the nature of the simulation first is mandatory to understand the motion of a single particle in the presence of a constant magnetic field, $\mathbf{B}$. Assuming that the particle travels at a speed $\mathbf{v}$, and that there's not electrical field, i.e $\mathbf{E} = \mathbf{0}$, the force experienced by the particle is:

$$m\frac{d\mathbf{v}}{dt} = q(\mathbf{v} \times \mathbf{B}) \tag{1}$$

This is known as Lorentz force. By taking the dot product with the velocity it can be shown that the Kinetic Energy, $\mathbf{W}$, of the particle is conserved under the action of such force.

$$m\frac{d\mathbf{v}}{dt}\mathbf{v} = q(\mathbf{v} \times \mathbf{B})\mathbf{v} \tag{2}$$

since

$$(\mathbf{v} \times \mathbf{B}) \cdot \mathbf{v} = 0 \tag{3}$$

Therefore

$$m\frac{d\mathbf{v}}{dt}\mathbf{v} = m\frac{d|\mathbf{v}|^2}{dt} = \frac{dW}{dt} = 0 \tag{4}$$

The motion of the particle can then be separated in two directions, perpendicular and parallel to the magnetic field. Parallel to the magnetic field, since $\mathbf{v}_\parallel \times \mathbf{B} = 0$ the motion of the particle remains unchanged. Perpendicular to the magnetic field, it follows that

$$(1) = \mathbf{v}_\perp \times \mathbf{B} \tag{5}$$

Due to the conservation of energy, i.e the magnitude of the velocity vector must remain unchanged, and to the perpendicularity of the Lorentz force to the velocity vector emerging from $\mathbf{v} \times \mathbf{B}$,the particle will experience circular trajectory in the plane perpendicular to the magnetic field. Expressions for the radius of the motion and its period can be reached equating (1) to the centripetal force.

$$q|(\mathbf{v}_\perp \times \mathbf{B})| = \frac{|v_\perp|^2 r}{2} \tag{6}$$

$$r_g = \frac{m|v_\perp|}{q|B|} \tag{7}$$

$$\tau_g = \frac{2\pi}{\omega_g} = \frac{2\pi m}{qB} \tag{8}$$

These quantities are known as the the gyro radius and gyro period.

Combining the two motions, the resulting trajectory of the particle is a helix contained in a cylinder of radius r_g and axis parallel to $\mathbf{v}_\parallel$.
Furthermore, another quantity can be defined relating $\mathbf{v}_\perp$ and $\mathbf{v}_\parallel$; perpendicular and parallel components of the velocity to the magnetic field respectively.

$$\alpha = \arctan\frac{\mathbf{v}_\perp}{\mathbf{v}_\parallel} \tag{9}$$

We will refer to this quantity as pitch angle.

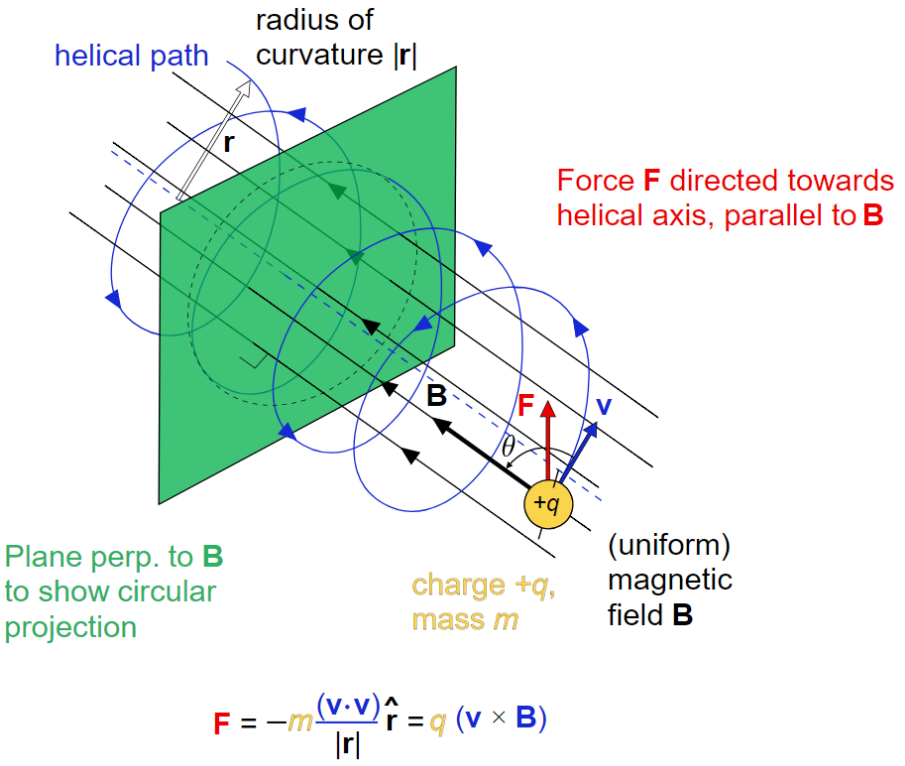


Figure 1: Trajectory of a charged particle in a simple Constant magnetic field

2.1.1 Boris integrator

This equations of motion can solved numerically applying a simple Euler forward integrating scheme. However this will result in unstable behavior in the first instances of the simulation for magnetic fields greater than the magnitude of the perpendicular component of the velocity. For a more faithful representation it is necessary to implement a so more complex integrating scheme that we will refer as Boris algorithm. Boris algorithm is an application of the leapfrog method [2] to Lorentz forces. As in leapfrog schemes, in Boris algorithms the space and the velocity variables of integration are half a step off phase. This can be more clearly appreciated by looking at the equations of the integrating scheme.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{v}_{k+1/2} \tag{10}$$

$$\mathbf{v}_{k+1/2} = \mathbf{u}' + q'\mathbf{E}_k \tag{11}$$

$$\mathbf{u}' = \mathbf{u} + (\mathbf{u} + (\mathbf{u} \times \mathbf{h})) \times \mathbf{s} \tag{12}$$

$$\mathbf{u} = \mathbf{v}_{k-1/2} + q'\mathbf{E}_k \tag{13}$$

$$\mathbf{h} = q'\mathbf{B}_k \tag{14}$$

$$\mathbf{s} = 2\mathbf{h}/(1 + h^2) \tag{15}$$

$$q' = \Delta t \times (q/2m) \tag{16}$$

In this App the user will be provided with information of the simulation as represented by the Boris algorithm and using the MATLAB built in solver ODe15s to solve the equations of motion.

The code used to implement this method can be found at the bottom of the call back function of each of the planet buttons referred as "Borisintegrator". For further information of how to access this code please go to section 3.3.

2.2 Single Particle Motion in a Magnetic Dipole

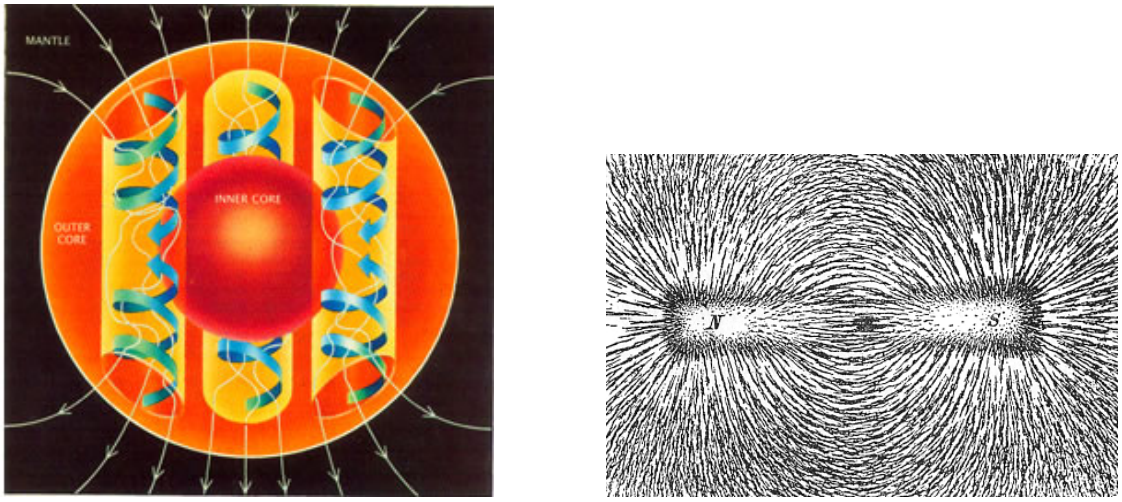
When studying the behavior of charged particles in the atmosphere of a planet the constant magnetic field applied in the previous section needs to be replaced by a magnetic dipole.

$$\mathbf{B}_{dip}(\mathbf{r}) = \frac{\mu_0}{4\pi r^3}(3(M \cdot \hat{\mathbf{r}})\hat{\mathbf{r}} - \mathbf{M}) \tag{17}$$

Planetary magnetic fields can generally be explained using dynamo theory which states that celestial bodies with convecting electrically conducting fluids can hold magnetic fields over astronomical time scales. Dynamo Theory describes that electrically conduction fluids flow inside celestial bodies due to heat convection from the inner core to the outer. Due to the rotation of the body, the convection currents form helical structures aligned with the axis of rotation of the body due to Coriolis forces. This solenoid like currents vary in time generating a time varying magnetic field similar to the one of pure magnetic dipoles governed by the induction equation.

$$\frac{\partial \mathbf{B}}{\partial t} = \mu \nabla^2 \mathbf{B} + \nabla \times (\mathbf{u} \times \mathbf{B}) \tag{18}$$

Where $\mathbf{u}$ is the velocity of the fluid and μ the magnetic diffusivity.



(a) Planetary Dipole

(b) Simple Dipole

Figure 2: Comparison between simple and planetary dipole

In this scenario the motion of the particle is essentially helical however the non-uniformity of the field rises two new modes of motion.Therefore, it is possible to distinguish three different types of motion.

- Gyro motion
- Bounce motion
- Drift motion

The gyro motion is the rotation of the particle around the guiding center and acts similarly to the previously studied case. The bounce motion is a periodic motion in which the guiding center of the particle oscillates between two given latitudes called mirror points. Finally, the drift motion displaces the guiding center longitudinally in the equatorial direction. These motions are periodical and trap charge particles in the atmosphere forming belts of radiating charged particles called Van Allen radiation belts

2.3 Adiabatic Invariants

Algebraic expressions for the equations of motions are attainable by considering the adiabatic invariants of the system, quantities that remain constant over the periods of oscillations. Adiabatic

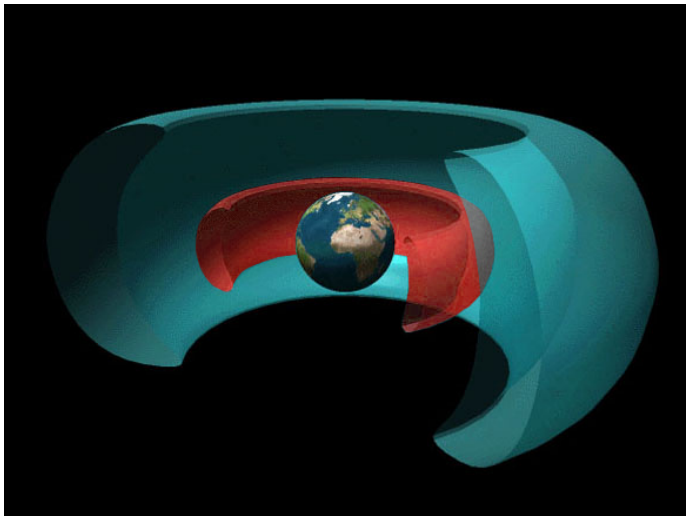


Figure 3: Earth's Van Allen radiation belts.

invariants appear when the time scales of motion are such that the parameters of an oscillation can be considered frozen in time with respect to the period of oscillation. In the studied case, the period of oscillation of the gyro motion is generally two orders of magnitude smaller than the one of bounce motion and the last one two orders of magnitude smaller than the period of the drift motion. This gives rise to three adiabatic invariants, one per mode of motion, that can be considered as frozen in time with respect the other modes. In the Hamiltonian formalism, these adiabatic invariants are equivalent to considering the action variable in the Hamiltonian formalism.

$$J = \oint (m\mathbf{v} + q\mathbf{A}) \cdot dl \tag{19}$$

where the integrand is the canonical momentum, dl the canonical position and A is the Euler potential of the magnetic field such as $\mathbf{B} = \nabla \times \mathbf{A}$

For the gyro motion, the canonical space of the system is the radius of motion of the gyro motion. Integrating using stokes theorem the following result is reached.

$$J_1 = \frac{m}{q}\mu \tag{20}$$

where μ is known as the first invariant.

$$\mu = \frac{mv_\perp^2}{2B} = \frac{W_\perp}{B} \tag{21}$$

Since B is not constant, as the particle gets closer to the cups of the field, poles where the field lines converge, B will increase demanding $W_\perp$ to increase analogously. However, $W_\perp$ can only take values up to W, $v_\perp = v$. When this maximum is reached is reached the particle stops. This is consistent with the sign of the drift gradient of the force that governs this motion, where the minus indicates that as direction of the force opposes the direction in which the field value increases, i.e the force decelerates the particle as it approaches the cups until the perpendicular kinetic energy cannot withstand the increase in the magnetic field and the particle stops.

$$\mathbf{F}_\nabla = -\mu \nabla B \tag{22}$$

The stopping point is referred as mirror point and it is reached when

$$B_{mirror} = \frac{W}{\mu} \tag{23}$$

The second invariant is associated with the bounce motion. if $r_g < \frac{B}{|\nabla B|}$,i.e. the magnetic field remains constant with respect the gyro motion, the motion of the particle can be approximated using the guiding center approximation. In the guiding center approximation the motion of the

particle is simplified as just the motion of the geometrical center around which the gyro motion revolves. In the absence of an Electrical field and thanks to the fact that the drift motion is "frozen in time" it is possible to approximate the action variable disregarding the equatorial displacement. This allows to simplify the bounce motion allowing us to simply constrain its limits between the upper and lower mirror point; $m1$ and $m2$. Since the motion takes place along a single field line, the area enclosed by it is zero which allows to cancel the second term of the expression.

$$J_2 = \oint m\mathbf{v} \cdot d\delta + \oint \mathbf{A} \cdot dl = 2\int_{m_1}^{m_2} mv_{\parallel} ds + \int\int \nabla \times \mathbf{A} d\delta = 2\int_{m_1}^{m_2} mv_{\parallel} ds \tag{24}$$

Which using the result for the first invariant can be expressed as

$$J_2 = 2\int_{m_1}^{m_2} mv\sqrt{1 - \frac{B(s)}{B_m}}ds = 2mvI \tag{25}$$

Finally for the third invariant we repeat the same operation but integrating along the whole drift shell. This is same as taking the flux through the shell if the electromagnetic field's frequency is smaller than the drift frequency.

$$J_3 = \Phi = \oint mv_d r d\phi \tag{26}$$

Expressing J_3 in terms of magnetic moment.

$$\Phi = B\pi r_g^2 = B\pi \frac{m^2 v_{\perp}^2}{q^2 B^2} = \frac{2\pi m}{q^2}\mu = constant \tag{27}$$

Since Φ must remain constant we can see that the gyro radius must decrease as the particle moves to the cups of the field when the field lines converge, i.e. the closer to the poles the smaller the gyro radius. This implies that the maximum value of the gyro radius is reached at the equator of the planet.

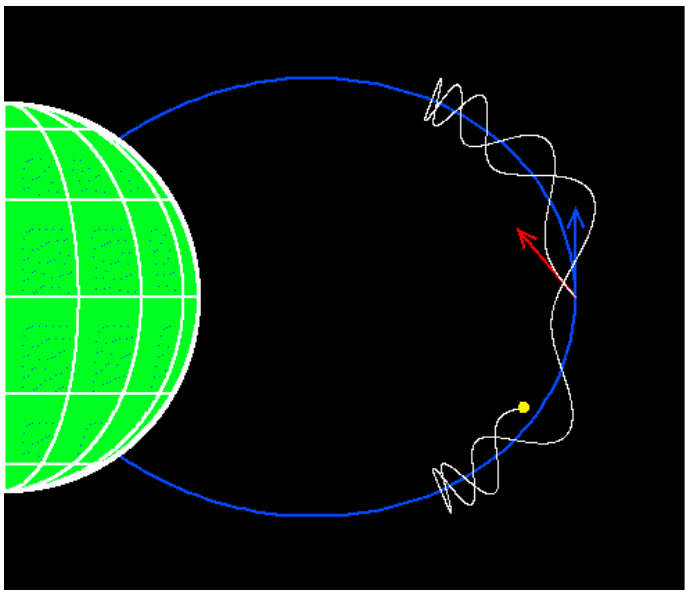


Figure 4: Trajectory of a particle in a simple magnetic dipole field generated by a planet.

2.3.1 The loss cone

Another expression for the magnetic dipole field can be obtained in terms of the latitude *theta* and the L-factor defined as $L = r_{eq}/R_E$ where $r = r_{eq}cos^2(\theta)$.

$$\mathbf{B} = \frac{B_E}{L^3}\frac{(1 + 3sin^2(\theta))^{1/2}}{cos^6(\theta)} \tag{28}$$

where $B_E = \frac{\mu_0 M_E}{4\pi R_E^3} = 3,.11 \times 10^{-5} T$

This expression is useful to find a value for the so called equatorial lost cone. Recovering the definition of pitch angle (9) and the first invariant μ it is clearly appreciable that the mirror point of particles with a small pitch angle will be at a higher latitude than those with greater pitch angles.This is due to the fact that for two particles with the same W, the one lower pitch angle particle will have a high $v_parallel$ component meaning that it will take longer until all its velocity aligns perpendicular to the magnetic field.This can lead to particles with mirror points inside the planet body. Particles with such pitch angle are said to be inside the equatorial loss cone since thy crash into the planet and no longer oscillate.

An expression for such loss cone can be achieved using the following identity which holds as long as the magnetic moment is conserved.

$$\frac{\sin^2 \alpha}{\sin^2 \alpha_{eq}} = \frac{B}{B_e q} \tag{29}$$

Where $B_e q = B_E / L^3$. When evaluated at the mirror point $\alpha_m = \pi/2$.

$$\sin^2 \alpha_{eq} = \frac{B_e q}{B} = \frac{\cos^6 \theta_m}{(1 + 3\sin^2 \theta_m)^{1/2}} \tag{30}$$

Similarly the latitude at which the field line meets the surface of the planet can be defined as

$$\sin^2 \alpha_l = \frac{\cos^6 \theta_E}{(1 + 3\sin^2 \theta_E)^{1/2}} \tag{31}$$

which by making the following substitution $\cos^2 \theta_E = L^{-1}$ it can be turned into

$$\sin^2 \alpha_l = (4L^6 + 3L^5)^{-1/2} \tag{32}$$

This shows that the loss cone angle only depends on the L factor of the orbit, in other words of the radius of the field line along it is drifting at the equator. This quantity can turn to be surprisingly small. For example for the geostationary orbit of Earth $6.6Re$, $\alpha_l = 3degrees$

2.4 Magnetodisc structures

The magnetic dipole approximation is a faithful representation for Earth's magnetic field for short ranges in which the Solar wind doesn't play a major role. However, when modeling planets like Saturn and Jupiter modifications must be made. Jupiter and Saturn, aside from the incoming plasma from the solar wind, receive a significant flow of plasma from their moons which eject great quantities of material to the atmosphere. This is caused by the high volcanic of their respective satellites. This material in orbit can become easily become ionized acting as a second source of plasma. The fast rotation of these planets leads to the creation of plasma toruses confined in the equatorial region due to the action of centrifugal forces. This results in the formation of equatorial plasma disc that adds an additional magnetic field stretching the dipole magnetic lines around the equator. This distortion of the magnetic dipole is commonly referred as magnetodisc.

Something also worth mentioning about Jupiter and Saturn magnetic fields is that, unlike Earth's, their magnetic poles are aligned with the geographical poles.

2.4.1 Caudal's formalism

In his 1986 paper Caudal provided a basis to model magnetodisc structure under a set of conditions that simplify the problem.

- The dipole axis is aligned with the spin axis
- The magnetic field and the plasma distribution are axis-symmetric
- The plasma pressure is isotropic and the temperature is constant along field lines.

- Gravitational forces are negligible.

Based in these assumptions, the force balance equation for the magnetodisc results in

$$\mathbf{j} \times \mathbf{B} = \nabla \mathbf{P} - nw^2 \rho \mathbf{e}_\rho \tag{33}$$

where the left hand side of the equation is the pressure due to the Lorentz force $\frac{\mathbf{F}}{A} = \frac{\mathbf{I} \times \mathbf{B}}{A} = \mathbf{j} \times \mathbf{B}$, $\nabla \mathbf{P}$ represents the plasma pressure and $nw^2 \rho \mathbf{e}_\rho$ is the pressure due to the centrifugal forces.

Applying a Euler potentials to express B and further assumptions a numerical solution to this equation can be reached. For more detail ...

3 App Instruction set

3.1 How to launch the app

Mat-lab R2017a and Mat-lab R2017b bring Mat-lab Apps as a default feature in their toolbox. Apps act as an updated version of the GUI system previously offered by Mat-Lab up until 2012's versions. Apps allow for a more friendly interaction with the code, facilitating the user interaction with it in cases such as inputing initial parameters or alike.

The first step to launch a mat-lab app is to download the .mlap file in the correct directory inside the machine, next the necessary auxiliary files. Once the files are properly allocated, the App can launched simply by double clicking in its file from the main mat-lab window (see figure 1)

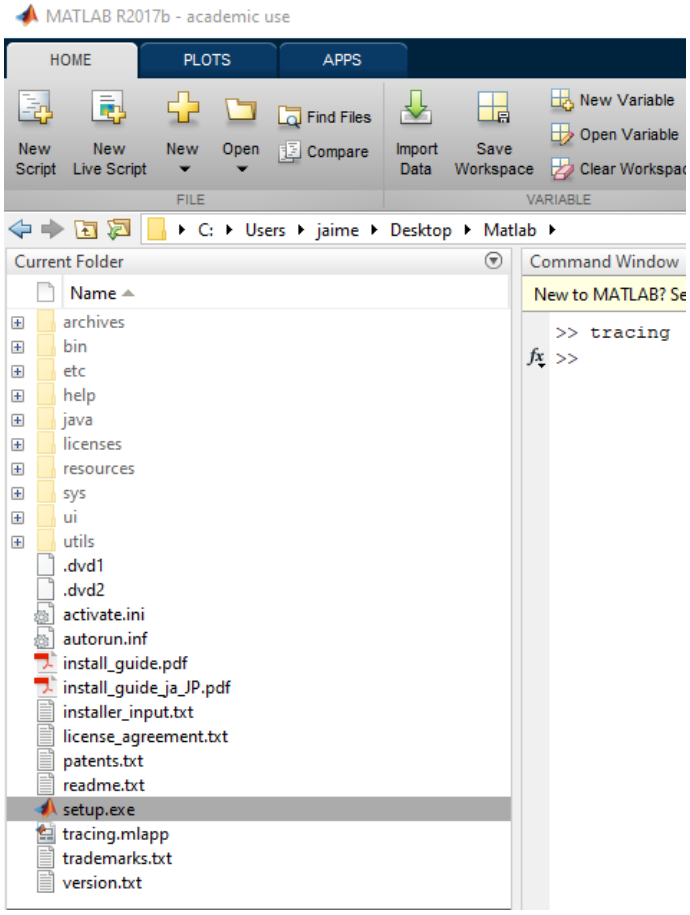


Figure 5: By double clicking in the highlighted file inside the file console the user can launch the App's GUI shown in figure 1.

3.2 How to interact with the app

If the App has been successfully launched, a Graphics User Interface (GUI) will show in your screen. The GUI of this App consists of 8 interactive gadgets. According to their functionality the gadgets can be classify in three groups. (Use figure 6 as reference)

3.2.1 Initial conditions

The app present two variables that can be changed by the user

- The initial kinetic energy of the proton (Gadget 2)
- The pitch angle (Gadget 3)

These are controlled by the gadgets 2 and 3 respectively. Gadget 2 is manipulated with the upwards and downwards arrows next to text box. This shifts the value of the energy of the proton by 10 mega electron volts in each click. It is recommended to start with 10 Mev for Earth, and 100 Mev for Saturn and Jupiter. Gadget 3 is a knob that controls the pitch angle expressed in degrees. We recommend to start with values between 15 and 30 degrees for all planets. Under 5 degrees loss cone phenomenon will appear.

3.2.2 Simulation Configuration

There are user decisions regarding the configuration of the simulation.

- The length of the simulation (Gadget 1)
- The graphics display (Gadget 4)
- The planet it will take place (Gadgets 5,6 and 7)

The length of the simulation is controlled gadget 1. Similarly to gadget 2, gadget 1 shifts the duration of the simulation by 1 bounce motion period. We recommend starting with 1 or 2 units of duration to first test the computation time of the machine. Gadget 4 controls the graphics the simulation will retrieve according to number of options chosen. There are 4 options offering a 3D animation comparison between the two algorithms, a 2D static comparison of the trajectories, an energy conservation comparison and a latitude motion comparison. Finally, gadgets 5, 6 and 7 launch the simulation in the selected planet when pressing one of the three buttons.

3.2.3 Authorship

The last gadget, 8, displays a window containing text information of the rights of use of the code and the authorship.

3.3 Understading the graphical output of the app

The app can produce a maximum of 4 graphical displays that compare the results of the Boris method and the MATLAB solver.

3.3.1 3D Animation

3D Animation creates a new window in which a 3d animation of the trajectory of the particle takes place. At the beginning, graph appears just with the planet in green tones and the guiding lines of the trajectory in yellow. As times elapses, the two calculated trajectories unravel in the same panel, having the Boris method in red and the MATLAB solver in blue.

In this animation the motion of the combined gyro, bounce and drift motion modes can be appreciated, as well as the deceleration of the particle towards the poles and the change of the gyro radius, faithfully representing its maximum at the equator. Once the two trajectories have been completed the legend will display and the User will need to press space in the command line of MATLAB's main window to proceed.

It is also worth noticing that the app will print a confirmation of the input parameters by the user in the command line.

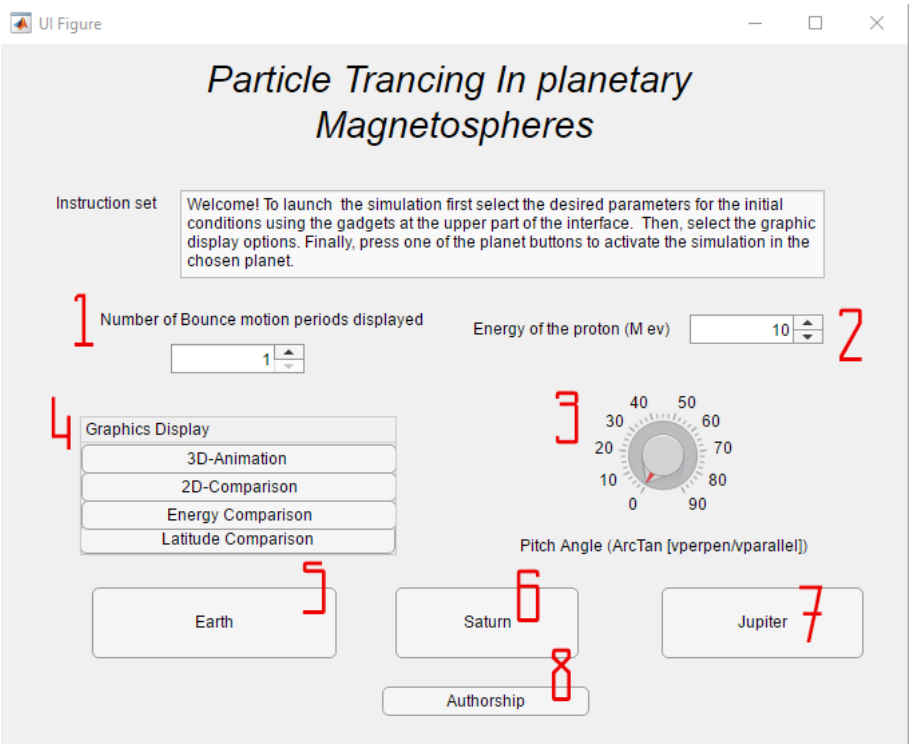
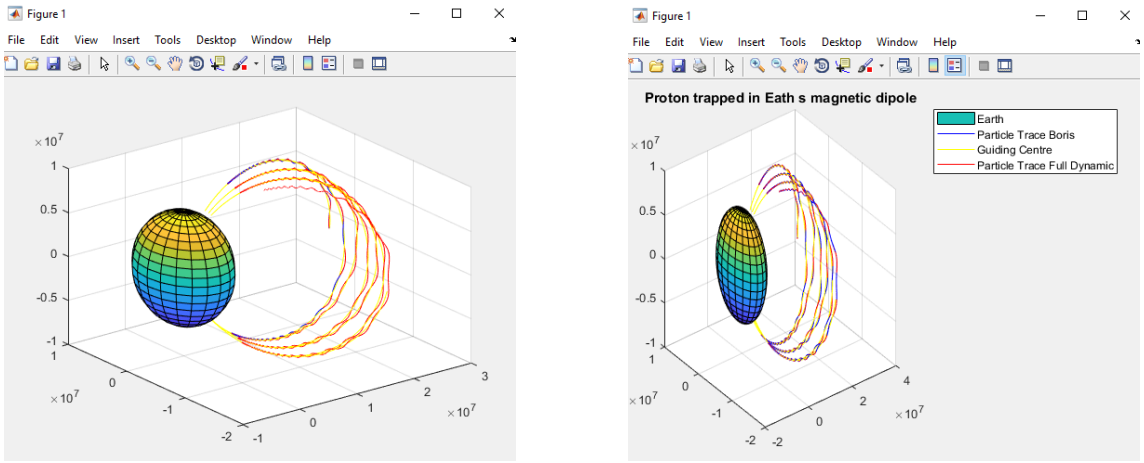


Figure 6: Early version of the GUI.



(a) In course

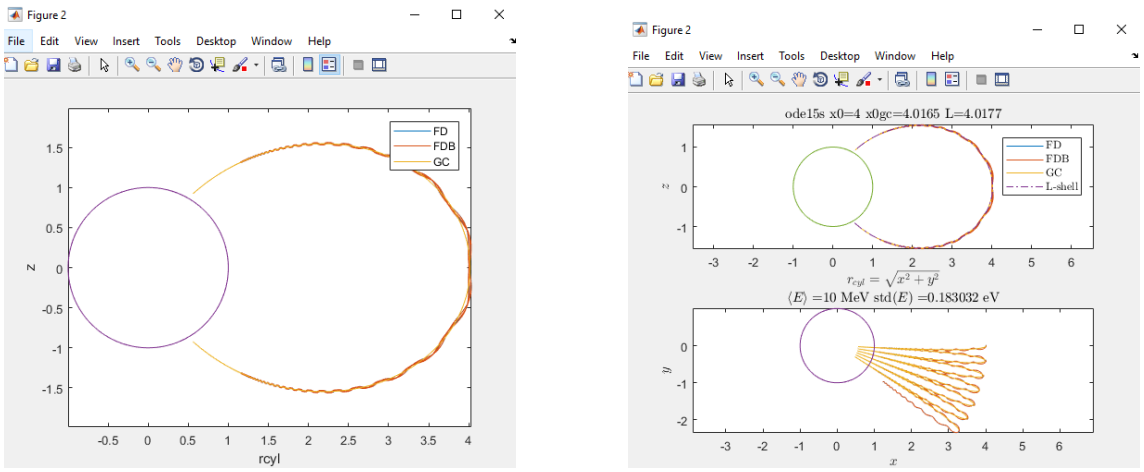
(b) Finished

Figure 7: 3D Animation panel

3.3.2 2D Comparison

2D Comparison will display two different panels showing the trajectory in an static manner, representing the planet as purple circle. The first panel, shows a profile perspective of the motion in which the bounce motion can be easily appreciated. The second panel contained two figures. The upper one, similar to the one in the first panel, displays a profile perspective with an added comparison to the L shell of the initial position which allows to see the stretching the field lines due to the magneto-disc structures.The lower figure, shows a zenith perspective of the motion in which the drift motion can be easily appreciated. After each panel, the user will need to enter space into the command line of MATLAB's main window to proceed.

It is also worth noticing that the app will print the exact value of L shell in the command line.



(a) In course

(b) Finished

Figure 8: 2D Trajectory comparison

3.3.3 Energy Comparison

Energy comparison displays a single panel with two figures. The upper one, compares the conservation of energy of the motion for both solver methods. This is due by comparing the difference between the instantaneous and the average energy at each time step of the simulation. The lower one, shows a comparison of the how faithfully the invariability of the μ factor is represented by comparing the value of μ for each time step against the one given by the initial parameters.
After each panel, the user will need to enter space into the command line of MATLAB's main window to proceed.
It is also worth noticing that the App will print the exact value of the mean energy for both solvers in the command line.

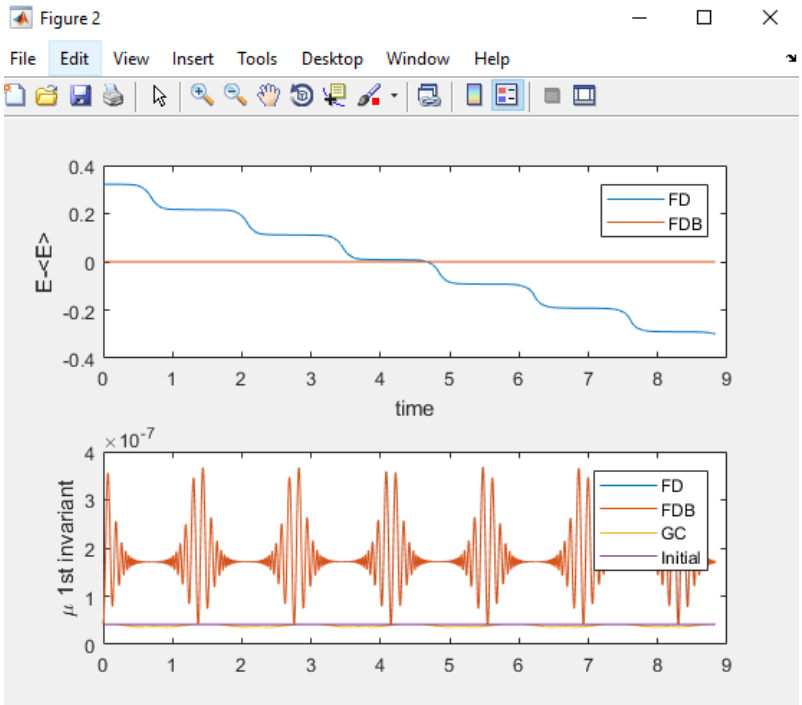


Figure 9: Energy comparison

3.3.4 Latitude Comparison

Finally, latitude comparison displays the trajectory of particle as by showing the change in latitude; upper figure in the panel, and longitude; lower figure in the panel. This allows to a easily appreciate the periodical nature of the motion.
After each panel, the user will need to enter space into the command line of MATLAB's main window to proceed.

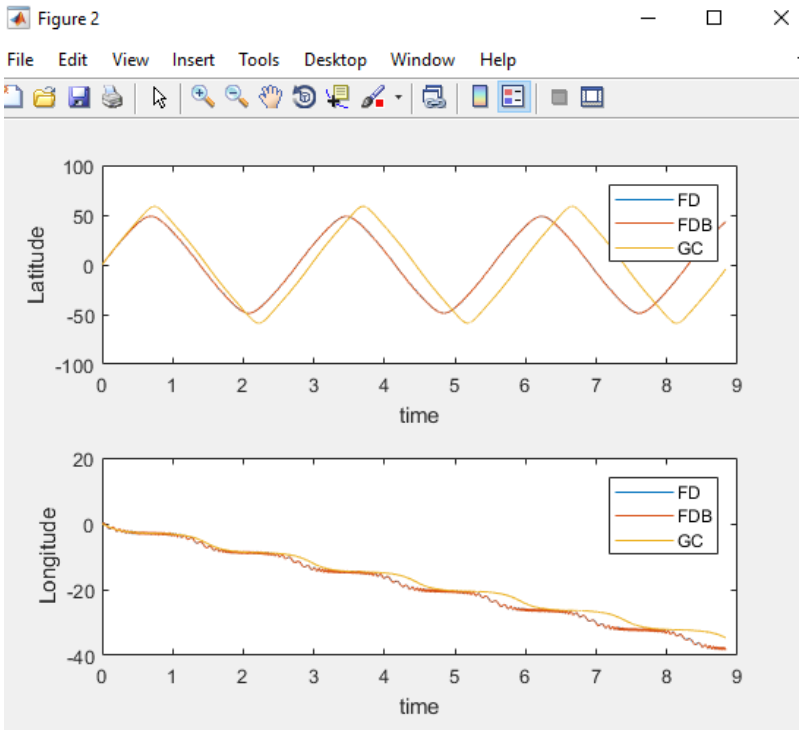


Figure 10: Latitude and Longitude comparison

3.4 How to change the app's code

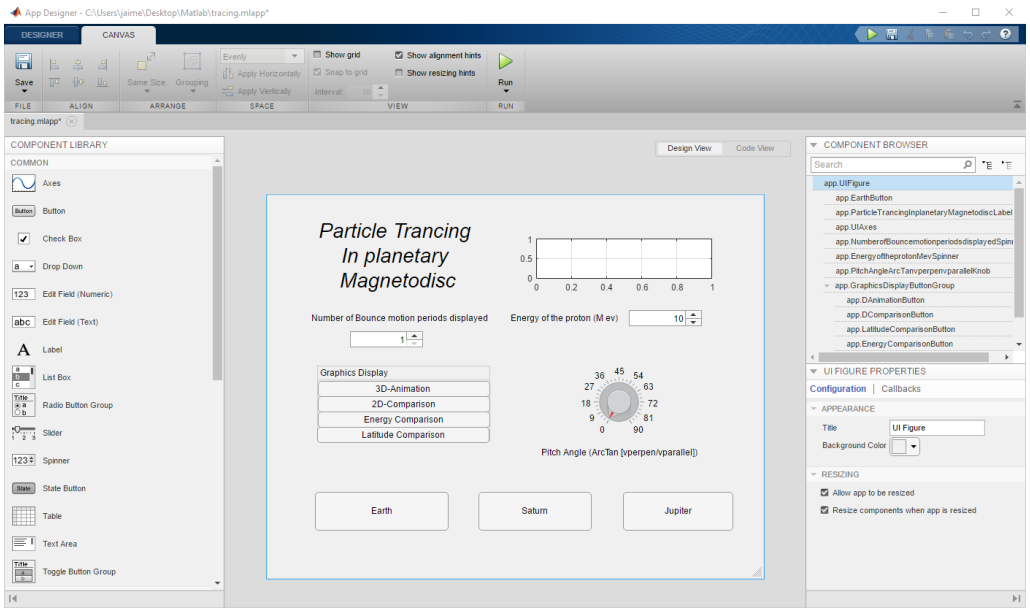


Figure 11: Desingner view of the app

To change the code of the App it is necessary to launch the App designer mat-lab window. To do so, the user must click in the open tool at the top left tool bar of th main mat-lab window and select from it the .mlapp app file (See figure 7). This will lead the user to the App designer window. From here the user can alter the provided code in two ways. First, the design view allows to the user to introduce new gadgets, modify the appearance of the existing ones, as well as, adding call backs.

Call backs associate the values inputted into the gadgets with functions whose output can be used later on. To better understand how call backs work it is beneficial to switch to the Code View by clicking on its button at the top right corner of the preview window. The code view allows to see the actions made on the design view in the traditional programming style, i.e. showing code lines instead visuals. From this view, we can see that each call back has created its own function with an associated output named as app.'name'.value. By explicitly calling these outputs inside other call back's functions, it is possible to allow for interaction between the different gadgets previously implemented.

In this code, the main bodies of code that perform the simulation have been written as functions that initialize when the respective planet button is pressed. The input of the user is taken by assigning the value of the initial conditions to the respective gadget call back app.'name'.value.

However, to gain a better understanding of how mat-lab apps work it is recommendable to follow the official introductory tutorials [3] since the explanation provided doesn't cover the majority of the depth of mat-lab app designer.

3.5 Understading the code behind the app

In this section we will use the code for the Earth simulation to understand the structure of the code used and allow the users to do their own changes.

Accessing the App's code through the code view mode of the MATLAB app designer window and scrolling down to the callback of the Eath's button, the user will find that the underlying code is thought as a general function that contains several inner functions that produce different needed steps when computing the simulation. The code of the general function can be broken down into three main bodies.

- Initial Parameters
- Tracing
- Plotting

3.5.1 Initial Parameters

The code starts by setting the MATLAB solver ODE15s as the solver that will be used for the built in method. By changing the commented line of code the user can alter this decision. Following from this, basic physical constants and user input are defined. The first function the user encounters in the code is the function "init" which builds initial conditions for the trajectory from the previous physical constants and user input. These initial conditions are.

- Xo - The inital position and velocity for the particle
- Xogc - The inital position and velocity for the guiding center
- mu - The first invariant
- tc - Gyro period
- tb - Bounce period
- td - Drift period
- lm - Mirror latitude

Following MATLAB's syntax the code for the function is defined a few lines after. In this code we can see that calculate this quantities the auxiliary functions 'dipoleMagneticField3D', 'mirrorlat', and 'periods' were used. To find the code for this functions the user must scroll down past the plotting section of the code.

3.5.2 Tracing

Tracing and Plotting occupies most of the lines of code of the general function. Its functions are all contained under the function 'trace' which is called just under the call to the 'init' function before definition of the 'init' function.

To find the definition of the 'trace' function the user must look for the end of the definition of the 'init' function. In it we can see that the function starts by defining the planet surface for the 2D comparison figures and follows by solving the trajectories for the particle and the guiding center using the built in methods. This is done by defining the respective differential equations as auxiliary functions that are then plugged into MATLAB's 'solverh' function. These auxiliary functions; 'dynamic3d' and 'gcdynamic3d', can be traced to the bottom of the code of the general function.

After this, the Boris algorithm is used to calculate the particle's trajectory, making use of the auxiliary functions 'BorisInit'; which provides the zeroth step, and a for loop that updates each point of the Xb grid using 'BorisIter'.

3.5.3 Plotting

Inside the same trace function one can find the lines of code responsible for generating the graphical output of the app.

All the plotting elements are contained in their respective if loops to trigger only when the user input is adequate and can be easily identified by looking for the respective gadget callback.

Normally, each plot code line is preceded by a few lines of code that calculate the particular parameters needed for the plot based on the out of 'init' and the already obtained parameters previously defined in the same 'trace'.

3.5.4 Auxiliary functions

For a detailed description of each auxiliary function mentioned please scroll down to the bottom of the general function.

3.5.5 Jupiter and Saturn modifications

Jupiter's and Saturn's code slightly differs from the one of the Earth. These changes mainly regard the auxiliary functions used to calculate the Magnetic field at each time step and can be easily identified inside their respective general call back function by comparison with the previously broken down example.

Nonetheless, this functions also rely on the presence of files aside from the file that launches the GUI. These files contain the data collected by several space missions that allowed to properly characterize the magnetosphere of these planets and allowed to the draw the boundary conditions of in the caudal formalism.

4 Acknowledgements

I would like to thank professor Patrick Guio for his vital guidance during the development of the project and for kindly providing the code that underlies it.

5 References

[1] N. Achilleos, P. Guio, C. S. Arridge (2010) ' A model of force balance in Saturn's magnetodisc'

[2] Ozturk, M. K. (2012), ' Trajectories of charged particles trapped in Earth's magnetic ' Monthly Notices of the Royal Astronomical Society, Volume 401, Issue 4, 1 February 2010, Pages 2349–2371

field', American Journal of Physics 80, 420–428.

[3] MATLAB Apps official web site: https://uk.mathworks.com/discovery/matlab-apps.html

[4] Encyclopedia of the Solar System, ISBN: 9780124158450. Chapters: 6, 7, 8 and 12.