



*Facultad
de
Ciencias*

**Un acercamiento numérico al
problema de los tres cuerpos**
A numerical approach to the Three Body Problem

Trabajo de Fin de Grado
para acceder al

GRADO EN FÍSICA

Autor: Diego Martín López

Director: Francisco Matorras Weinig

Julio - 2015

Abstract

The three-body problem is a nonlinear, non analytically solved without approximations problem which is key in the understanding on the motion of three free bodies bounded by gravitational attraction, such as the Sun-Earth-Moon system. The following work presents the development and results of a numerical program that solves the three-body problem as if none of the work around it were done before. We've started from the Newton's gravitational law and continued with the learnings got from the Physics degree. We've solved the ODE_2 for the acceleration of each of the bodies, so we got it's positions as a function of time, considering the bodies moved always in a Cartesian plane. The program was developed to numerically integrate these equations and to permit the analysis of different quantities related to the observed motion under different conditions. We've extensively tested the code for different examples, such as the Sun-Earth-Moon system and the Satellite-Earth-Moon system. We got that, while the integration step is $h < 10$ s and the bodies aren't closer than $R^* = 9 \times 10^6$ m, the program returns reasonable solutions for the Sun-Earth-Moon system and the Satellite-Earth-Moon system. We've done a fine-tuning to get the simulated system to behave in a similar way the real system does, but getting the orbits more circular in order to simplify the interpretations of the results. The initial conditions we had to set when the three bodies were free were so fine that, for example, changing the Earth's speed $\delta v_{\oplus} = 1$ m/s, made its period change $\delta T_{\oplus} = 1$ hour. When we set the initial conditions so the bodies had to move in circular motion, we got that Earth and Moon disturbed each other. The Earth got an eccentricity of $\epsilon_{\oplus} = 0.05$ and the Moon $\epsilon_M = 0.28$. We've study the Moon's stability when making little perturbations in its orbit, while the other conditions remained the same. We got that it needs to change its path $\delta\theta = 1.2^\circ$ in order to disengage from the Earth's orbit. We've make an attempt to study the stability of the system, and, during two hundred years, it hasn't seem to have any trend to change. Comparing the geostationary satellite with and without Moon, we got that, because of the Moon influence, it had a delay of around two hours every day. We made a study of the orbits for different distances to the Earth. We got that while the satellite remains near the Earth (not further than $R^{**} \approx 2 \times 10^8$ m), there are periodical perturbations in the orbit with respect to the circular case, but they don't seem to present any trend to change its orbit. Once the satellite surpasses this critical orbit the Moon traps it and its orbit gets chaotic. In this stage the satellite usually gets so close to the Moon that the error stack starts to be a problem. However, taking a small enough integration step, this error shouldn't mean any problem. When getting too close to the Moon, the satellite gets a boost in its speed due to momentum transfer. If the energy transferred results enough, the satellite escape from the system. Apart from the cases where the satellite and the Moon get too close to each other, where we may need an adaptive integration step, the program works well and allows us to make a detailed study of any case of the problem. Taking a look to the whole project, we have developed a reliable tool, within the approximations we have taken. The program has resulted useful for the study of these type of problems and, with some improvements to cover some identified weak issues, can be used for a more detailed studies of the three body problem itself or other similar problems.

Keywords: Numerical Calculus, Three Body Problem, Runge-Kutta, artificial satellites, classical gravitation.

Resumen

El problema de los tres cuerpos es un problema no lineal, no resoluble analíticamente sin aproximaciones, clave en la comprensión del movimiento de tres cuerpos que se atraen gravitacionalmente entre sí, como pasa en el sistema Sol-Tierra-Luna. El siguiente trabajo presenta el desarrollo y los resultados de un programa de cálculo numérico que resuelve el problema de los tres cuerpos como si ningún otro trabajo se hubiese desarrollado antes sobre él. Hemos partido de la ley de la gravitación universal de Newton y seguido con los conocimientos adquiridos en el grado en física. Haciendo uso de Matlab y a través de la implementación del método Runge Kutta de cuarto orden, hemos resuelto la EDO_2 para la aceleración de cada uno de estos cuerpos, de manera que obtuvimos sus posiciones como funciones del tiempo. Consideramos que los cuerpos siempre se movían en el plano cartesiano. El programa fue desarrollado para integrar numéricamente estas ecuaciones, y permitírnos analizar diferentes magnitudes relacionadas con el movimiento observado bajo diferentes condiciones. Realizamos un análisis exhaustivo del código con diferentes ejemplos, como el sistema Sol-Tierra-Luna y el sistema Satélite-Tierra-Luna. Obtuvimos que, mientras el paso de integración fuera $h < 10$ s y la distancia entre los cuerpos no fuese menor de $R^* = 9 \times 10^6$ m, el programa nos daba soluciones razonables para los sistemas Sol-Tierra-Luna y Satélite-Tierra-Luna. Hemos hecho un ajuste fino para que las simulaciones se comportasen de manera parecida a los sistemas reales, pero con las órbitas más circulares para poder simplificar las interpretaciones de los resultados. Las condiciones iniciales que tuvimos que introducir para los tres cuerpos libres eran tan sensibles que, por ejemplo, cambiando la velocidad de la Tierra $\delta v_{\oplus} = 1$ m/s cambiaba su período $\delta T = 1$ hora. Cuando inrodujimos las condiciones iniciales para que la Luna y la Tierra se movieran de manera circular d emanera separada, nos dimos cuenta que al simular el sistema completo éstas dejaban de moverse circularmente pues se perturbaban la una a la otra. La excentricidad de la Tierra era de $\epsilon_{\oplus} = 0.05$ y la de la Luna de $\epsilon_M = 0.28$. Hemos estudiado la estabilidad de la Luna tras hacer pequeñas perturbaciones en su órbita, mientras el resto de las condiciones las dejábamos igual. Encontramos que ésta necesitaba cambiar su trayectoria 1.2° para desacoplarse de la órbita terrestre. Hemos tratado de estudiar la estabilidad del sistema, y durante doscientos años no hemos encontrado ninguna tendencia del sistema que indique algún cambio en su trayectoria. Comparando el satélite geostacionario con y sin Luna, hemos visto que, debido a la influencia de ésta, el satélite sufría un retraso de dos horas cada día. Hicimos un estudio de las órbitas del satélite a distintas distancias de la Tierra, y mientras éste permaneciese no más lejos de $R^{**} = 2 \times 10^8$ m, encontramos perturbaciones periódicas que no aparecen en el caso completamente circular, pero no parecen mostrar ninguna tendencia a cambiar su órbita. Cuando el satélite sobrepasa la distancia límite, la Luna lo atrapa y empieza a comportarse de manera caótica. En este caso el satélite suele acercarse tanto a la Luna que el error acumulado empieza a ser problemático. Sin embargo, tomando un paso de integración suficientemente pequeño, este error no debería suponer un problema suficientemente grande como para cambiar la trayectoria del satélite. Cuando el satélite se acerca tanto a la Luna, éste recibe un incremento de velocidad debido a transferencia de momento. Si esta energía es suficientemente grande, el satélite puede incluso escapar del sistema. Aparte de los casos en los que el satélite se acerca demasiado a la Luna, donde debiéramos tomar un paso adaptativo, el programa funciona bien y nos permite hacer un estudio detallado de cualquier caso de este problema. Teniendo en cuenta el proyecto entero, hemos desarrollado una herramienta en la que confiar, dentro del límite de nuestras aproximaciones, y que ha resultado útil para el estudio de este tipo de problemas y, con algunas mejoras para suplir las deficiencias ya identificadas, puede ser empleado para estudios más detallados del problema de los tres cuerpos u otros problemas similares.

Palabras clave: Cálculo numérico, Problema de los tres cuerpos, Runge-Kutta, satélites artificiales, gravitación clásica.

Contents

1	Introduction	4
1.1	Approaches and sense of the project	4
2	Development and tests of the code	6
2.1	4 th order Runge Kutta	6
2.2	First and Second Order Differential Equations	7
2.3	The Two Body Problem	8
2.3.1	Circular orbits	9
2.3.2	Elliptical orbits	11
2.3.3	The Rocket Problem	12
3	The Three Body Problem and the Sun-Earth-Moon system	15
4	The three body problem and the Earth-Satellite-Moon system	23
5	Conclusions	28
6	References	30

1 Introduction

The Three-body problem consists in solving the positions of three free bodies that move due to gravitatory attraction. The equation we are interested in solving for each body is then:

$$\frac{d^2 \vec{x}_i}{dt^2} = G \left(\frac{m_j \vec{r}_{ij}}{d_{ij}^3} + \frac{m_k \vec{r}_{ik}}{d_{ik}^3} \right)$$

where x_i is the position of the body we are studying, m_j and m_k are the other two bodies masses, d_{ij} and d_{ik} are the distances between the first body and the other two and \vec{r}_{ij} and \vec{r}_{ik} the vectors that join the first body with the other two.

Due to the fact that these equations can't be solved by analytic methods without approximations, this problem has always been a very interesting topic in physics. It might appear as a subject with not very useful applications, but when it comes to study the celestial bodies, it is the first order tool in order to solve our questions on the matter.

The present work has been outlined as if none of the previous work around the Three-body problem had been made and focuses in a very practical line of work, so it starts from the Newton's law of universal gravitation and makes use of the 4th order Runge Kutta algorithm to make numerical simulations of the trajectories of three different bodies from some initial conditions. Our goal is to get to simulate the three body problem and get interesting results after simulating the Sun-Earth-Moon system and the path of an artificial satellite between the Earth and the Moon.

The inner motive of this work is learning how to plan and solve any numerical problem, and how to deal with a big project, from the planning to the conclusions.

We'll start showing the implementation of the 4th order Runge Kutta algorithm with a first and a second order differential equations and the tests we made to make sure our program worked so far. Then we increased the complexity of the code implementing the Two-body problem with which we simulated the Earth-Moon system. Once we had enough proves our program worked as well as we needed we implemented the Three-body problem. After some exercises with the Sun-Earth-Moon system we decided to remove some complexity to the code in order to make it faster and be able to get easier conclusions from the results, so we fixed the most massive body and continued studying the Sun-Earth-Moon system a bit more. The last exercise was to simulate an artificial satellite for different orbits between the Earth and the Moon. We'll finally finish the report with the conclusions we've got from this project.

1.1 Approaches and sense of the project

Before we start writing about the project, we must make sure the reader doesn't get the wrong idea about the meaning of this work. We don't claim we have done something new, neither something useful to research or implement in a serious working environment. We've just developed a tool in order to take some examples of numerically solved three body problem with some approximations and without using the most optimized resources, but the ones we considered appropriate and that granted us the success of the project on a reasonable time lapse.

We focused on the project from an academic perspective. We didn't really want to get a specific result but to learn as much as possible from the problems we were finding along the way. At the end we actually found some interesting results and we got to simulate the three body problem, always within the limits of our approximations.

In the first place we considered our bodies as points in a Cartesian Coordinate System plane. This means there are nor rotation or collisions. The orbits we've considered for the Sun-Earth-Moon system are set in a plane, while there's actually a small degree of inclination between the Moon-Earth orbit plane and the Sun-Earth orbit plane.

We also tried the motion of the bodies to be as circular as possible, so the results derived from the simulations were easier to understand. When we wanted the results to be even simpler, we fixed

the most massive body, as we'll explain later.

The physics of the problem aren't as important, along the project, as the computational attack we've made to solve it. We've use the three body problem as an example to solve numerically, but, making the proper changes, we could use the program to solve any other.

2 Development and tests of the code

I chose to work with Matlab because it's a really friendly user tool, with lots of libraries and helpful resources along the net, which derives from the fact that it's widely used along the world. We also got to know, with the first trials we did, that Matlab was going to let us develop the tool we were looking for in a flexible way.

In the early stages of the project, we knew it wasn't possible, nor practical, for us to start with the three-body problem at the very beginning. We had to take little safe steps in order to develop a reliable tool.

That's the main reason why we started solving analytically easy to solve differential equations and the well known two-body problem. We'll start by showing the 4th order Runge Kutta method we used for the whole project and then introduce the two approaches we came with before we got into the three-body problem.

2.1 4th order Runge Kutta

The Runge Kutta methods are an important set of useful numerical algorithms in the resolution of ordinary differential equations. These methods were developed at the beginning of the 20th century by the German mathematicians C. Runge and M. W. Kutta.

We chose the 4th order Runge Kutta to solve our problem because it's conceptually simple in order to understand how to use it and powerful enough to let us get to a reasonable result at the end of the project.

Given $f'(t, x) = f(t, x)$, the algorithm divides the integral we usually would do analytically into four discrete steps, where the slope is approximated by four coefficients we've called x_1 , x_2 , x_3 and x_4 . At the end of each iteration, we get the weighted average of the four portions. Between the solution of an iteration and the next one, there's an interval of time given by h , which is called the integration step.

The mathematical shape of the algorithm is written as follows:

$$\begin{aligned}x_1 &= f(t_i, x_i) \\x_2 &= f(t_i + \frac{h}{2}, x_i + \frac{h}{2}x_1) \\x_3 &= f(t_i + \frac{h}{2}, x_i + \frac{h}{2}x_2) \\x_4 &= f(t_i + h, x_i + hx_3) \\x_{i+1} &= x_i + \frac{h}{6}(x_1 + 2x_2 + 2x_3 + x_4)\end{aligned}\tag{1}$$

where $i + 1$ is the next step we are calculating, and i is the step we've already got. This iterative process begins from a known initial condition $f(0, x)$.

If we want to solve a second order differential equation, $f''(t, x) = f(t, x)$, then the algorithm gets a bit more complicated. In this case we have to divide the ODE_2 into two ODE_1 so we can proceed in a similar way as we did with the previous case. The order in which we write the line codes is important, because of the interdependence of coefficients on each step.

$$\begin{aligned}v_1 &= f(t_i, x_i) \\x_1 &= v_i \\v_2 &= f(t_i + \frac{h}{2}, x_i + \frac{h}{2}x_1) \\x_2 &= v_i + \frac{h}{2}v_1\end{aligned}\tag{2}$$

$$\begin{aligned}
v_3 &= f(t_i + \frac{h}{2}, x_i + \frac{h}{2}x_2) \\
x_3 &= v_i + \frac{h}{2}v_2 \\
v_4 &= f(t_i + h, x_i + hx_3) \\
x_4 &= v_i + hv_3 \\
x_{i+1} &= x_i + \frac{h}{6}(x_1 + 2x_2 + 2x_3 + x_4) \\
v_{i+1} &= v_i + \frac{h}{6}(v_1 + 2v_2 + 2v_3 + v_4)
\end{aligned}$$

where $x_1, x_2, x_3, x_4, v_1, v_2, v_3$ and v_4 are the coefficients we have to calculate in order to get the next step of the integral.

Because the time inside the simulation doesn't usually correspond to the time the simulation is running, we have to make a distinction between these two concepts. For example, a twenty minutes running time simulation, may correspond to a simulated time of one year. This means that while we have been waiting for twenty minutes until the simulation finished, the bodies we have simulated has been looping around each other for one virtual year.

Along the project we'll see how important it is to choose the right h . The idea for the method is to choose the smallest positive h possible, but the smaller h is, the slower is the method. Because of our limited time, we had to find a h which would give us a good precision-time relation.

2.2 First and Second Order Differential Equations

The first step we took was to solve a first order Ordinary Differential Equation (ODE):

$$\begin{aligned}
\dot{x} &= -kx \\
x &= e^{-kt}
\end{aligned}$$

where k is a constant, and x is a function of time t .

The first prove of the success of the implemented Runge-Kutta to Matlab was the comparison between the program numerical approach and the analytical solution. When plotting both solutions together we found out that both overlapped, as we could expect. Then we plotted the difference δ_z in absolute value for different h .

As we see in Fig.(1)) the error function shape describes the one of an exponential function: at the beginning the exponential function decays very fast, so the error function has a much bigger peak the less steps in time you get (less steps in time means bigger h). An interesting result here is that for a ten times smaller step, the code needs ten times more time to finish the simulation, but the result is several orders of magnitude more precise (four in this case, as we see in Fig.(1)).

The second test for the code was a second order ODE,

$$\begin{aligned}
\ddot{x} &= -kx \\
x &= \sin(kt) + \cos(kt)
\end{aligned}$$

with which we proceeded the same way as with the ODE_1 shown above. Once we got the numerical solution to overlap the analytical we got the error plots for different h .

In this other example, the analytical function is periodical, so the mean error should be independent of time. However, we see in Fig.(2) that it increases with time. This comes from the fact that in numerical analysis the error is cumulative, so for long simulated time examples we'll see not very precise results for the last points if we don't set a h small enough.

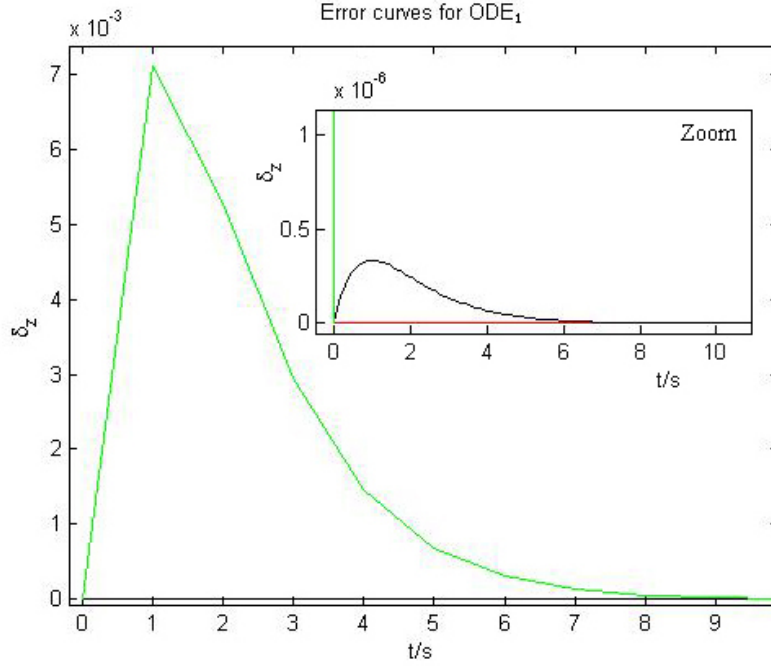


Figure 1: Error functions of the ODE_1 trial for different integration steps, from $h = 1$ s to $h = 10^{-4}$ s each step ten times smaller. The curves for $h < 0.1$ s are too small to be seen in the shown graph. $h = 1$ s is plotted in green and $h = 0.1$ s in black.

We've seen that the error we get for $h = 0.1$ s is already quite small, at least for the first minutes of simulated time. When we take h even smaller the error becomes practically negligible. We've also seen that the error is cumulative, so simulations with long simulated time will have big errors at the end of them if we are not cautious of taking a small enough h .

2.3 The Two Body Problem

The Two-body problem consists in solving the positions of two free bodies attracting each other with their respective gravitational force.

This problem meant a big approach to the real topic we were after. In this case the ODE to be solved was the one for the acceleration for each body due to the gravitational attraction to each other. Because we had two bodies, we needed two ODE_2 :

$$\begin{aligned}\ddot{x}_1 &= G \frac{m_2 r}{|r|^3} \\ \ddot{x}_2 &= G \frac{m_1 r}{|r|^3}\end{aligned}$$

where G is the gravitational constant, m_1 and m_2 are the masses of the bodies and r the distance between each other.

The way we implemented these ODEs to Matlab, was in the Cartesian coordinate system. It might appear better to introduce them with a polar coordinate system, but it was conceptually easier to program it in the Cartesian system and we didn't want to program a very optimized and difficult code, but just a simple tool with which we could study afterwards some examples of the Three-body problem.

The main difficulty on the implementation was the tedious work of writing two equations for each

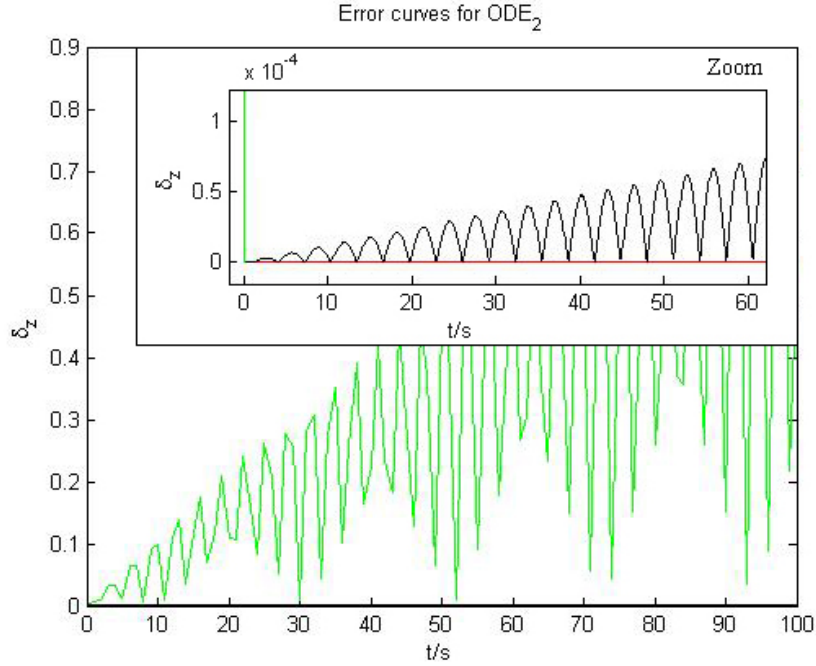


Figure 2: Error functions of the ODE_2 trial for different h , from $h = 1$ s to $h = 10^{-4}$ s. The curves for $h < 0.1$ are too small to be seen in the shown graph. $h = 1$ s is plotted in green and $h = 0.1$ s in black.

body (because it's an ODE_2) and coordinate four times (one for each integral part). This resulted into thirty two different equations on the program code.

Because we didn't need every step value for any of these magnitudes, we only saved one over a hundred or a thousand steps in order to plot later the orbit paths or the distances between the bodies. This way, a lot of simulation time was saved, because Matlab doesn't like to work with too large vectors.

In order to have a point of reference with a daily known case, we used the masses and distances of the Earth-Moon system. In this case we compared our results with the expected movement of the Moon, approximating the Moon orbit around the Earth as a circular planar orbit, and letting both bodies free. With this new code we made two tests focused in the problems we might had when we expanded it to the three-body case. In the first place we checked which h was needed so that the Moon circular orbits closed without much error. Afterwards we put the Moon in elliptical orbits and checked for the error of the closed orbit depending on the initial velocity angle.

2.3.1 Circular orbits

While going through the mathematical problem we may got that the orbit the body goes over is closed, but when studying the problem from a numerical perspective it's fair to find some error. Depending on the ratio time-quality we'll allow more or less error in the simulation.

Making the Moon go around the Earth in circular orbits was quite easy using the uniform circular motion. We just had to give the Moon the following velocity:

$$v = \sqrt{G \frac{M_{\oplus}}{d}}$$

where G is the Gravitational constant, M_{\oplus} is the mass of the Earth and d is the distance between the two bodies.

In order to study the error on the circular orbit, we had to pay attention on the position of the Moon on the moment before it crossed the horizontal position after the first loop (x_1, y_1) and the moment when it had already crossed it (x_2, y_2) . We can approximate the error on the loop δx as the distance to the origin from the point of the line that links (x_1, y_1) and (x_2, y_2) that crosses the horizontal. This is better explained with the scheme of Fig.(3). It's easy, once we've got (x_1, y_1) and (x_2, y_2) , to derive that,

$$n = \frac{y_1 x_2 - y_2 x_1}{x_2 - x_1}$$

$$m = \frac{y_1 - n}{x_1}$$

and

$$\delta x = \frac{-n}{m}$$

where m and n are the coefficients of the linear function that includes the points (x_1, y_1) and (x_2, y_2) .

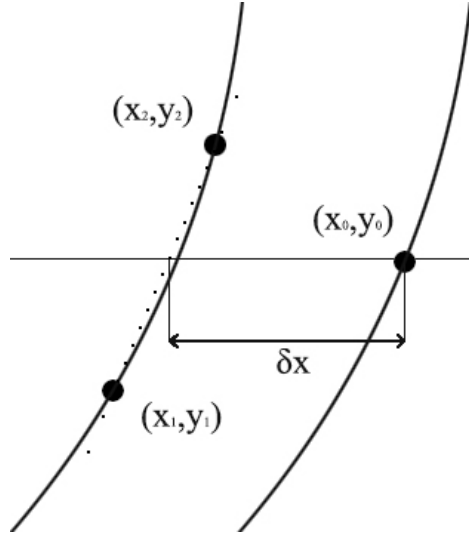


Figure 3: Scheme of the error δx taken in the two bodies circular orbits example. (x_0, y_0) is the initial point of the orbit, (x_1, y_1) is the point before the body crosses the $y = 0$ line, and (x_2, y_2) is the point one step after (x_1, y_1) .

	$h = 10^{-2} \text{ days}$	$h = 10^{-3} \text{ days}$	$h = 10^{-4} \text{ days}$	$h = 10^{-5} \text{ days}$	$h = 10^{-6} \text{ days}$
$\delta x / \text{m}$	-250.78	-1.014	-0.0122	-3.31×10^{-4}	-1.226×10^{-4}

Table 1: Errors for the closed orbits in the two bodies circular orbits example.

As we see in the values from Table (1), and as we may expect from the results shown in the previous sections, we get a more precise answer for a smaller integration step. Once more we see that a ten times smaller step sometimes result in a more than ten times more precise result. It's important not to forget that a ten times smaller step simulation is ten times slower, so we couldn't take the smallest h to get the most precise answer, but a reasonable h so we get a reasonable relation of precision and running time. For the two body problem we used $h = 10^{-4}$ days, which costs around four seconds of waiting for thirty days of simulated time. It's important to remark that $\delta x = 10^{-2}$ m isn't really a big error if we take into account the fact that the orbits we were working with were around 6×10^6 m wide.

The integration step from now on has been taken in day units just because it's more convenient to write down in the code that the Moon body is going to be orbiting for thirty times the integration

step in order to make one loop than multiplying it every time by sixty squared times twenty four. At the end, when taking $h \approx 10^{-4}$ days we are taking around ten seconds each step. It's not that important to take one unit or other, but to know with which unit you are working with and how much error are you taking at the end of the simulation.

As a conclusion from this exercise, we got that $h = 10^{-4}$ days is a good enough integration step for our precision-running time ratio, and it's the one we took for the following simulations.

2.3.2 Elliptical orbits

Once we finished studying the solutions of the code for the circular case, we wanted to make an automatic sweep code, so we could avoid as much manual analysis as possible in future simulations. For this, we prepared the moon with the same initial conditions as for the circular case, and then changed the initial velocity angle, from 90° to different angles. We wanted it to stop after one loop, so we imposed the following condition:

$$\text{if } y_2(i) < 0 \ \&\& \ y_2(i+1) > 0 : \text{ stop}$$

so the second body (the Moon) stopped once it had crossed the horizontal line upwards.

The result of the simulation for different initial angles is presented in Fig.(4), with the path of each of the orbits. As we expected, the bigger the angle with respect to the vertical, the more elliptical the orbit is.

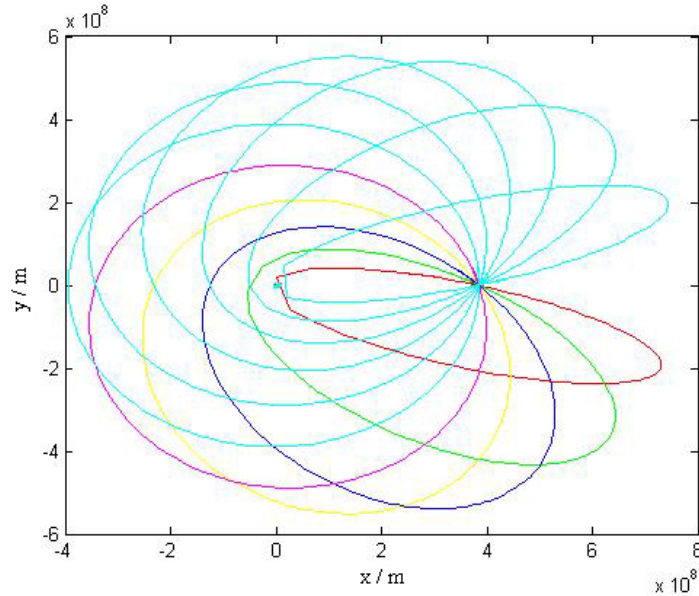


Figure 4: Orbits of the Moon around the Earth for different initial angles.

We've avoid to shoot the Moon at 0° and 180° , just because at those angles the Moon crashes with the Earth and no orbit is described.

Next thing we did was to check for the error on the closed orbits for the different angles. For that we draw the difference δ_x between the end of each loop and the initial position of the body for the first three loops.

As we see in Fig.(5), and as we saw with the ODE_2 in the previous section, the error is cumulative. We get another learning, for close enough orbits (we see it for 15°) the error is bigger. This gives us a clue about how well works the program when two of the bodies are too close. When searching for the minimum distance reached between the Moon and the Earth when the initial angle is 15° , we got $R_{min} = 8.705 \times 10^6$ m, so in further analysis will have into account the fact that for around this

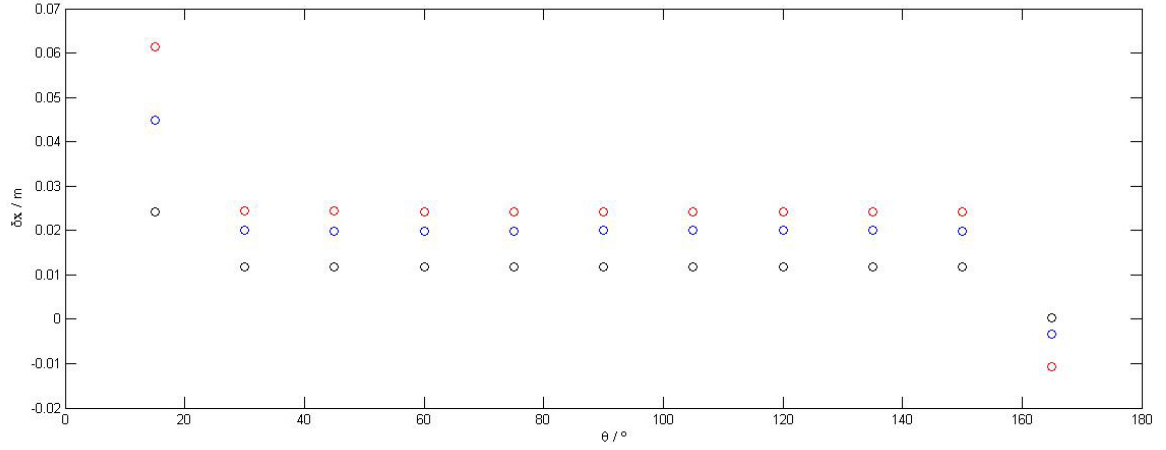


Figure 5: δx for the first three loops of the Moon around the Earth for different initial angles, from 15° to 165° , in 15° steps. Black points are from the first loop, blue ones for the second and the reds for the third. It's been used $h = 10^{-4}$ for this simulation.

distance or less, the program starts to be quite less precise.

The reason the program doesn't work so well when the bodies are too close comes from the fact that the integrals are proportional to $\frac{1}{d^2}$, so when d is small the integral in each step gets too big. This could be solved with an adaptive step program that we haven't implemented, so we could decrease h when the bodies came too close to each other.

We should point out that this distance between the bodies we've got as problematic, while seeming quite big when thinking that the bodies are just points, it's actually quite near the Earth radius value ($R_\oplus \approx 6 \times 10^6$ m). So the Moon represented in our program starts to move not as perfect as it should when it's almost touching the surface of the Earth. It's quite difficult to make ourselves a good idea of the magnitudes we are dealing with, but this far, the results we've presented for the code, while not being really optimized, are good enough for our purpose.

We can say that the program works right for the Two-body problem, which is quite nearer on complexity to the Three-body problem.

2.3.3 The Rocket Problem

In order to make another automatic program, we searched for the initial speed a rocket would need to reach the Moon from the surface of the Earth. The two bodies represented were the Earth, set in the centre of the system, and the rocket, which started in the Earth surface distance to the centre, as we see in the scheme from Fig.(6). The break condition for the program this time was reaching the Earth-Moon distance with the second body, and the program changed the rocket initial velocity and started over once the rocket started to fall down the Earth.

We made a sweep on the velocities of the rocket until it could get to the desired height. In order to get to $R_M = 3.85 \times 10^8$ m we knew the rocket needed a smaller speed than its escape velocity, with which it would never fall into the Earth again.

It's trivial to derive that the escape velocity for the rocket is:

$$v_e = \sqrt{\frac{2GM_T}{R_T}} = 3.056 \times 10^9 \text{ m/day} \quad (3)$$

so the velocity we were searching for was smaller than this result.

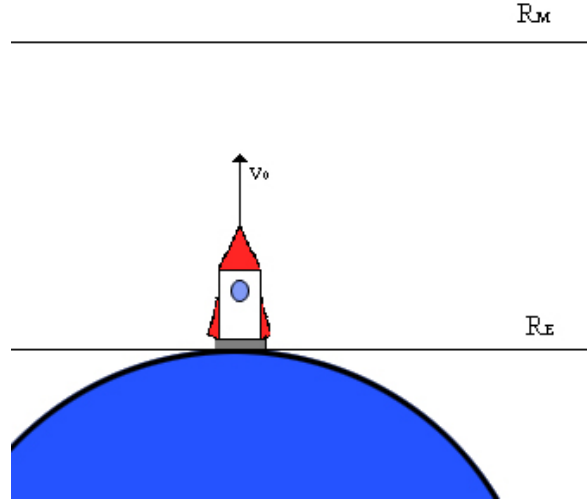


Figure 6: Scheme of the rocket problem. R_E is the Earth radius, R_M is the distance between the Earth and the Moon, and v_0 is the initial speed of the rocket.

The theoretical distance we use to compare with the simulation comes from the conservation of the energy, so the energy when the rocket is in the Earth's surface must be the same as the energy when the rocket is somewhere along the way to the Moon height and just starts falling down.

$$\begin{aligned} E_1 &= \frac{-Gm_i m_{ii}}{y_0} + \frac{m_{ii} v_0^2}{2} \\ &= \frac{-Gm_i m_{ii}}{y} \\ &= E_2 \end{aligned}$$

where G is the gravitational constant, m_i is the Earth's mass, m_{ii} is the rocket mass, v_0 is the initial velocity of the rocket, y_0 is the Earth radius (where the rocket starts from) and y is the position of the rocket when it is starting to fall. Solving this for y we get:

$$y = \frac{Gm_i}{\frac{Gm_i}{y_0} - \frac{v_0^2}{2}} \quad (4)$$

so the error δh we plot in Fig.(7) is just y minus the integrated height in the simulation.

As we see in Fig.(7), the error pattern we get is quite fluctuating and decreases until it suddenly grows up at the end. The simulation finishes with the initial velocity with which the rocket reaches the Moon. The velocity we got is $v_f = 9.7 \times 10^8$ m/day, which is smaller than the escape velocity, as we expected.

The figure starts with a big error probably because the rocket starts around the conflict area where we may say the two bodies are too close, so the program isn't as precise as we may like.

As the rocket gets away from the Earth the error decreases, because the cumulative error is less important than the one that appears when the bodies are too close to each other. At the end of the graph the error suddenly grows up probably because the initial velocity of the rocket is each loop closer to the escape velocity, and, taking a look to Eq.(3), when that happens Eq.(4) gets closer to infinity. This is similar to the case when the two bodies are nearby, when the program start to get numbers too big it starts getting bigger errors.

The last important point from Fig.(7) are the fluctuations. They probably come from the truncation Matlab does. For some speed the value it gets after the truncation is very similar to the theoretical value, but for the very next speed the truncation makes the final result to have quite a discrepancy with respect to the theoretical one.

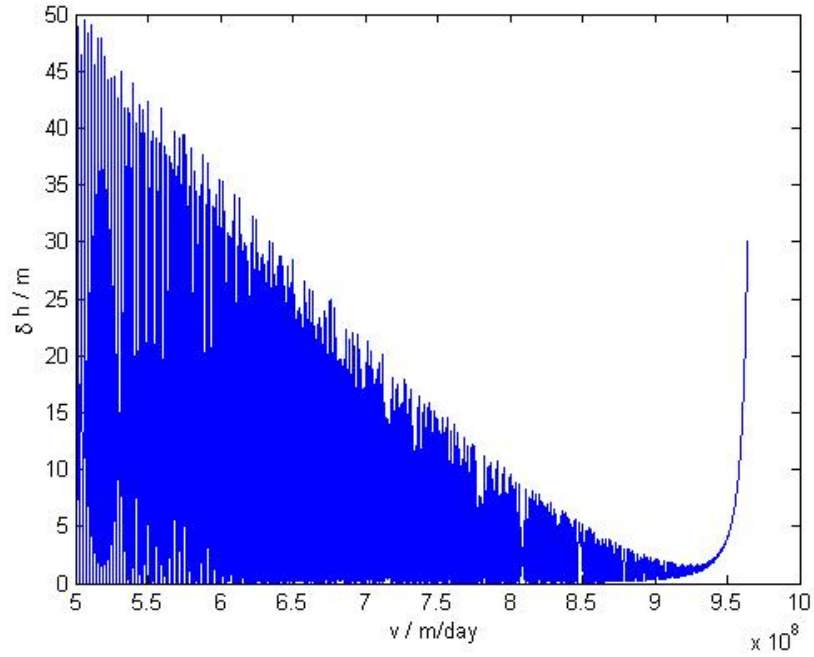


Figure 7: Difference between theory and simulation δh for the height of a rocket shot from the Earth's surface with certain velocity v . For this simulation it's been used $h = 10^{-4}$.

We also simulated the rocket problem with $h = 10^{-5}$ days and we got that the fluctuations got a hundred times smaller, so apparently the integration step has a stronger influence over the program than the truncations, and because we couldn't control these last ones in an easy way, we continued using h as an indicator of the precision on the simulations.

With the rocket problem we got a code which was quite independent (it made long sweeps changing the initial conditions of the system on its own) and with apparently good results.

3 The Three Body Problem and the Sun-Earth-Moon system

Once we got to this point we already knew we had a program that was able to simulate a two-body system with quite enough precision - when the bodies weren't too close - during around one year of simulated time. We were finally ready to take the step of extending the code to the third body. However, before we got to the Satellite-Earth-Moon system, we got into a much better known system, the Sun-Earth-Moon system, which allowed us to interpret better the results of the simulations.

Before we get into the code checks, it's important to specify the way the three bodies started to move from in the program. Taking into account that we were going to simulate planar orbits, we set the Sun in the centre of coordinates, located the Earth at the Sun-Earth distance horizontally from the Sun ($(x_{Earth}, y_{Earth}) = (d_{S-E}, 0)$) and the Moon at the Earth-Moon distance from the Earth ($(x_{Moon}, y_{Moon}) = (d_{S-E} + d_{E-M}, 0)$). From there, the three bodies were "shot" vertically (at 90°) with their initial velocity, where the initial velocity of the Sun was such that the centre of mass was at rest, $v_\odot = \frac{m_2 v_\oplus + m_3 (v_\oplus + v_L)}{m_1}$. The Moon's and Earth's velocity were chosen so they made circular orbits when they were in couples (Sun-Earth and Earth-Moon). This configuration will be called the "Moon-eclipse formation" from now on, and it's represented in Fig.(8).

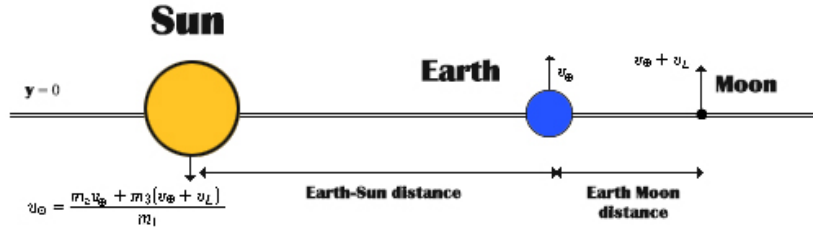


Figure 8: Moon eclipse formation scheme. v_\odot, v_\oplus and v_L are the initial velocities for the Sun, the Earth and the Moon, respectively, and m_1, m_2 and m_3 are their masses, in the same order.

The first thing we realized was that the orbit of the Earth around the Sun without Moon ($m_{Moon} = 0$ kg), starting with the initial conditions we got from the bibliography, didn't finish after three hundred and sixty five days, it needed four more days to finish it. In order to fix it we tried using a smaller integration step, as we see in Fig.(9), but it didn't solve it, which took us to think that it wasn't a cumulative error due to a large simulation time. The problem were the initial conditions. Searching for the initial conditions that made the Earth finish the orbit after one year, we got that the Earth initial velocity had to be very precise, and we used $v_\oplus = 2.9784 \times 10^4$ m/s. While with no Moon the Earth's orbit finished at 365 days and 12 minutes, the orbit with the three bodies finished at 365 days and 12 hours. Changing the velocity $\delta v = 1$ m/s meant a variation of $\delta T = 1$ hour in the period of the Earth. Actually, because the bodies weren't constraint to make circular orbits, in this case, making the speed higher didn't make the period smaller, because the orbit got more elliptical. This meant a higher speed resulted in a higher period. The reason because the Earth's period decreased when including the Moon into the system is related to the angular momentum. When there's no Moon, the total angular momentum, considering the Sun as fixed, and the system moving in circular motion, for simplicity, is as follows:

$$L_1 = m_\oplus r_\oplus v_\oplus$$

where m_\oplus is the mass of the Earth, r_\oplus is the distance from the Earth to the Sun and v_\oplus is the Earth's tangent velocity. When including the Moon we have to add another term,

$$L_2 = m_\oplus r_\oplus v'_\oplus + m_M r_M v_M$$

where m_M is the Moon's mass, r_M is the distance from the Moon to the Sun and v_M is the Moon's tangent velocity. The Earth's mass and distance to the Sun don't change, but its speed does, so, because of the angular momentum conservation, $L_1 = L_2$, the new Earth's velocity becomes,

$$v'_\oplus = v_\oplus - \frac{m_M}{m_\oplus} \frac{r_M}{r_\oplus} v_M \quad (5)$$

so the new Earth's velocity is always smaller than the previous one, which means a delay in the Earth's period when considering a circular motion.

So, a small variation in these initial conditions meant a big difference in the Earth's period, and without a very careful look, we could even get an a priori surprising result. This first conclusion made us be much more careful with the initial conditions we set before starting any simulations.

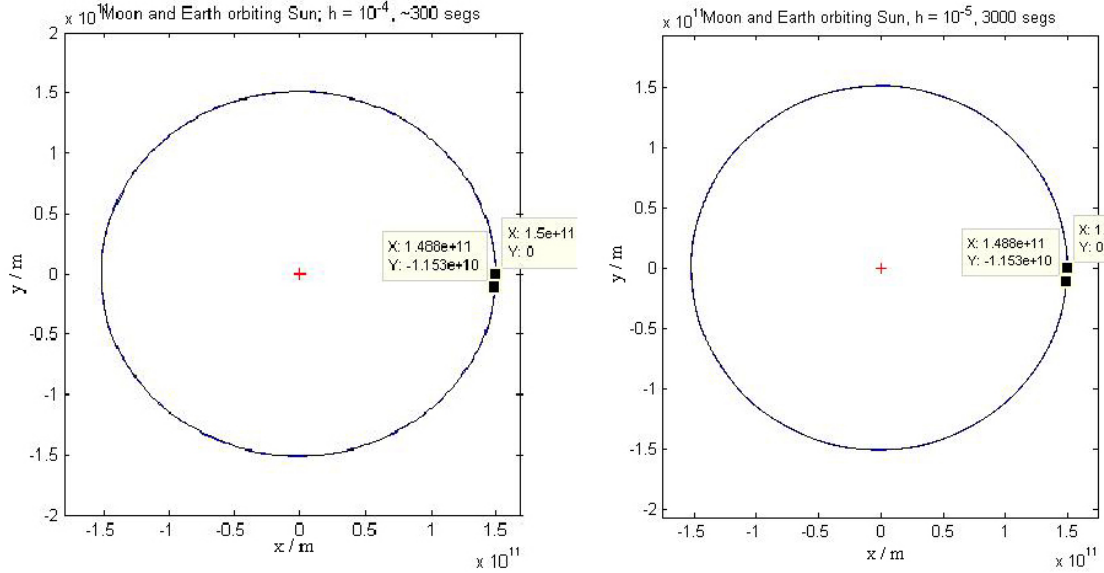


Figure 9: Orbit of the Earth around the Sun. In the left figure the step used is $h = 10^{-4}$ days and in the right figure, $h = 10^{-5}$ days. However, in both orbits the Earth finishes in the very same spot.

Looking into the eccentricity of the orbits of the Earth orbit around the Sun and the Moon orbit around the Earth, we got that the orbits described in our program were more circular than the ones of the real system. The data from the bibliography are $R_{max\oplus} = 1.52 \times 10^{12}$ m for the Earth apogee and $R_{min\oplus} = 1.47 \times 10^{12}$ m for its perigee. In the Moon case, $R_{maxM} = 4.06 \times 10^8$ m and $R_{minM} = 3.56 \times 10^8$ m. Using the following definition for eccentricity,

$$\epsilon = \sqrt{1 - \left(\frac{R_{min}}{R_{max}}\right)^2}$$

we got the theoretical values, $\epsilon_{\oplus} = 0.2543$ and $\epsilon_M = 0.4808$. With our simulations, however, using the velocity we talked about before, we get $\epsilon_{\oplus} = 0.0360$ when there's no Moon, and $\epsilon_{\oplus} = 0.0478$ and $\epsilon_M = 0.2776$ with the Sun-Earth-Moon system complete.

This more circular system we've got might be caused because we have search for the initial conditions, and we've set them in the Moon eclipse formation, as if the bodies were going to move circularly. The higher eccentricity when including the Moon in the system comes from the perturbation that the Moon causes in the Earth's orbit.

Although we got really circular orbits, the Moon made twelve and a half loops around the Earth in one year - remember that the Moon period is almost a month - which is very close to the real system behaviour, where it does almost thirteen and a half loops. As we see in Fig.(10), the Earth without Moon has a smooth path around the Sun, but when the Moon is included we see some perturbations on its path. Those perturbations and the path of the Moon reflects the loops of the Moon around the Earth.

As we see in Fig.(11), we simulated the three body motion for over two hundred years of simulated time, and we didn't find any variation trend in their paths. Nor the Earth or the Moon changed drastically their distance from one another and the Sun, as we expected, because the real system appears to have been working for at least that amount of time.

In the figure, while the Moon-Earth distance is much smaller than the Moon-Sun or the Earth-Sun

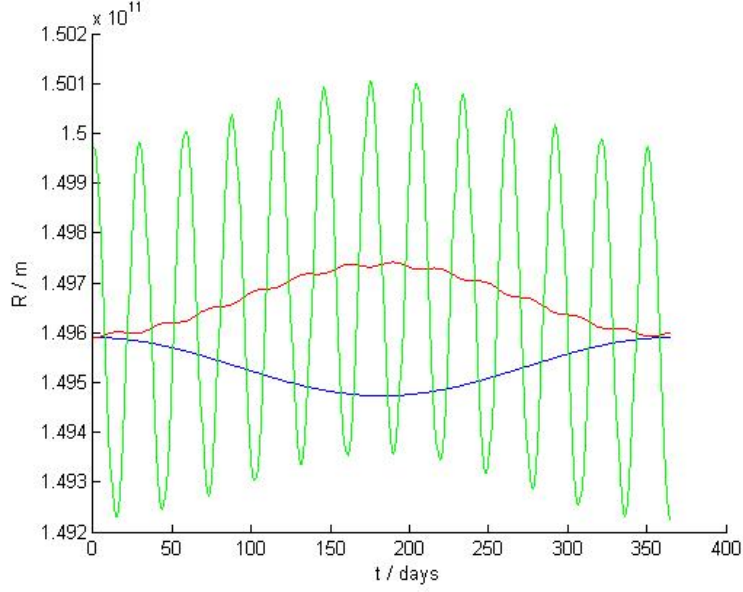


Figure 10: Earth distance to Sun without Moon (blue), with Moon (red) and Moon distance to Sun (green) along one year time.

distances, it appears bigger because of the normalization and the fact that the eccentricity of the Moon orbit around the Earth is much bigger than the eccentricity of its path (or the Earth path, which is very similar) around the Sun, as we saw before.

The peaks we see for the Earth-Moon distance that repeats every eleven years is a period we haven't pay attention to, due to the fact that there doesn't seem to be any similar correspondence with the real Sun-Earth-Moon system, but it's just a result of the approximations we have made. It may be related to some sort of resonance effect with the Sun-Earth or Sun-Moon orbits.

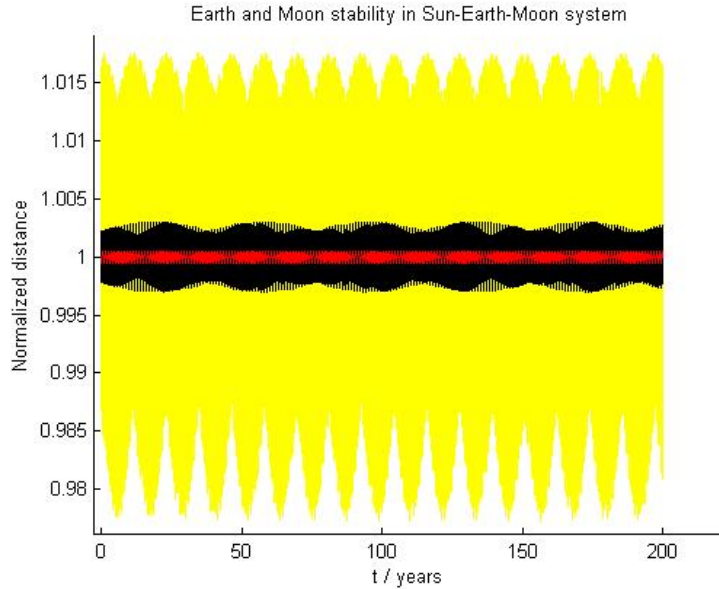


Figure 11: Moon-Sun (black), Earth-Sun (red) and Moon-Earth (yellow) normalized distances to their own mean value. After two hundred years of inside simulation time, no perturbation on the distance trend of the bodies is found.

In order to have a deeper look on Moon's stability, we tried, with the same velocity and distance

to the Earth and the Sun, to make it start from a different point than the one presented in the Moon eclipse formation, as we see in Fig.(12). As expected, the Moon orbit changed its path, even disengaging from the Earth's orbit. In this case, it seems the resultant orbit is stable, but usually, taking initial conditions a bit distorted, result in a Moon that even gets away from an orbit around the Sun.

It's important to remark that, in the three body problem, although we might have got a simulation which represents quite good the conditions of the real Sun-Earth-Moon system, within the limits of our approximations, we shouldn't forget that a little change in the initial conditions we took for it, makes the whole system to finish in a very difficult to predict result.

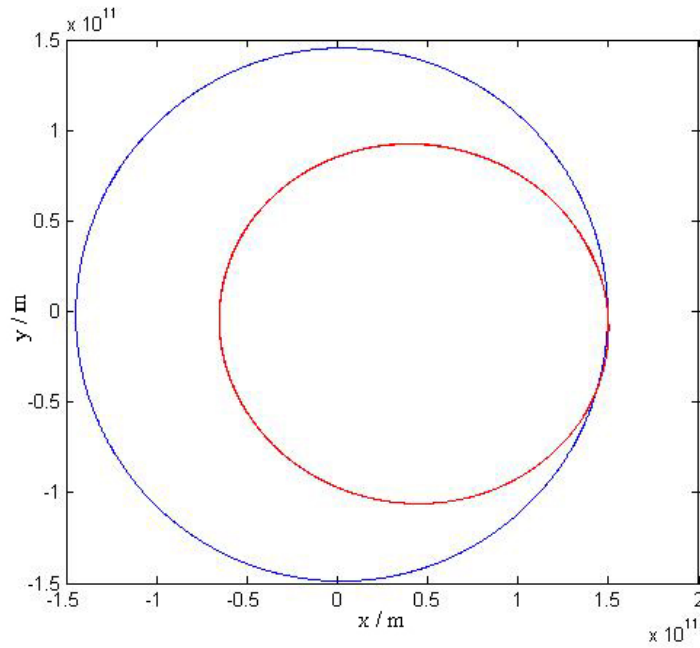


Figure 12: Moon (red) and Earth (blue) orbits around the Sun. The Moon didn't start from the Moon-eclipse formation, but from the same distance as the Earth from the Sun.

So far, the program worked well simulating the three body problem with the Sun-Earth-Moon system, always within the range of our approximations. However, in order to make the problem a bit simpler and get results a bit easier to interpret, we decided to fix the most massive body, the Sun in this case. This way, the code got simpler, because we removed an equation, and it got faster. The mass of the Sun is so big compared with those of the Earth and the Moon that the effect over it from them is almost negligible. This is the reason because this approximation is quite fair, although the results of the program change a bit, getting simpler to understand because there's one less body moving around.

We started by studying the perturbations that the Moon caused to the Earth again (now with the Sun fixed) in order to get a comparison with the case when the Sun is free. Then we made a comparison between the Earth and Sun forces over the Moon and finally we looked for the angle of disengagement for the Moon orbit around the Earth in the Moon-eclipse formation.

As we've seen in Fig.(10) the Moon described twelve and a half loops around the Earth and the Earth orbit got a bit distorted due to the Moon effect. In Fig.(13) we see the same effect, with the difference that when there's no Moon, because the Sun doesn't move, the Earth describes a perfect circle around it when we use the initial conditions for the uniform circular motion, as we expected.

Another thing we were curious about was the influence of the most massive bodies on the third

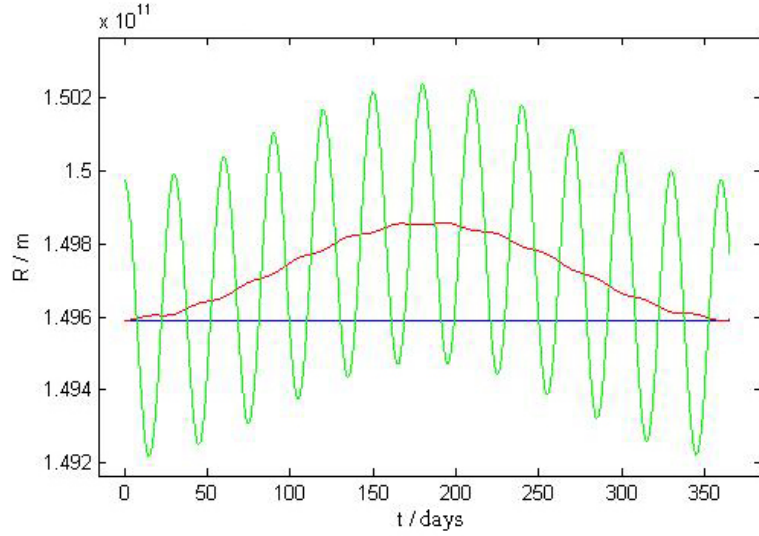


Figure 13: Comparison of the effect of the Moon on the distance between Earth and Sun. The blue line is the distance between Earth and Sun without Moon (Moon mass equals zero), and the red curve is the distance when there's Moon. The green curve is the distance between Moon and Sun, that explains the perturbations on the Earth orbit and reflects the twelve months of the year with its peaks (the Moon period around the Earth is almost a month).

one. As we could see in the orbit trace of the Moon when drawing the three bodies, it didn't seem to make orbits around the Earth, but around the Sun, as we see in Fig.(14).

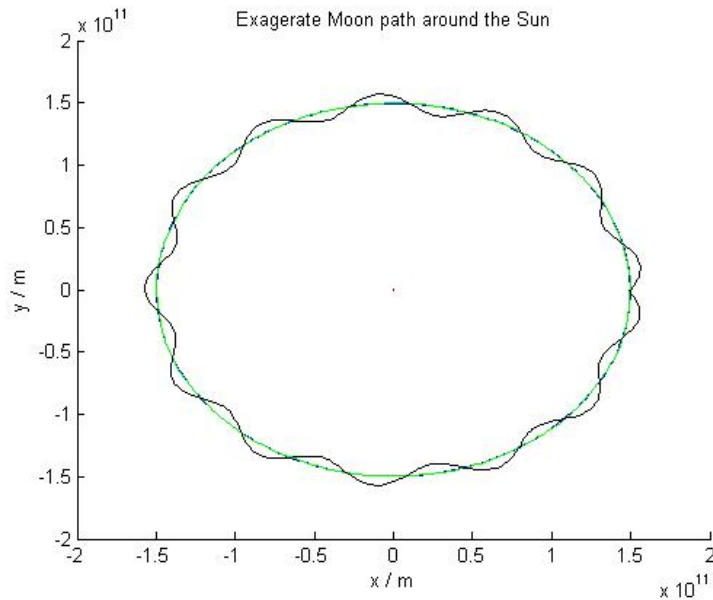


Figure 14: Moon orbit (green) with the same orbit exaggerated (in black) so we can see the 'zig-zag' that the Moon makes around the Earth (blue path under the green path of the Moon). In the centre there's the fixed Sun (in red).

This isn't a revolutionary concept, but comes a bit against the most daily intuition. It's true that if we observe the Moon from the Earth perspective, it moves around it. But when we come to analyse which body the Moon motion was more influenced by, we got that both the Sun and the Earth had a similar contribution, but the one of the Sun was stronger (twice as big actually), as we see in Fig. (15).

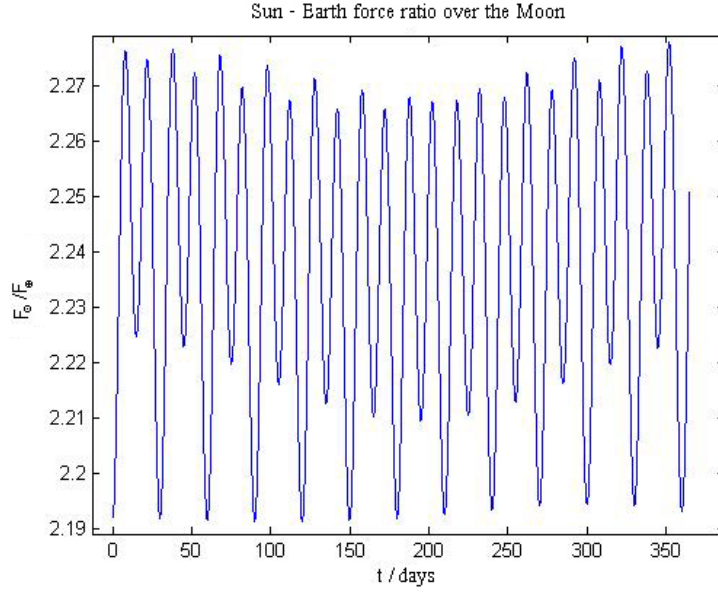


Figure 15: Ratio of forces of the Earth and Sun against the Moon. The Sun attracts the Moon twice as much as the Earth does.

That's because from the Sun perspective, the Moon doesn't seem to orbit the Earth, but moves around it with some perturbations due to the proximity of the Earth. This is of course a frame of reference problem, because as we've seen in Fig.(9) from the Earth's perspective the Moon makes circles around it. Because the Moon is almost in a constant gravitatory field (its relative difference in distance to the Sun and Earth is quite small) from its own point of view the Earth and the Sun make almost perfect circles around itself, as we see in Fig.(16).

In order to explain the peaks on Fig.(15) we have to take a look on the distance graph comparison on Fig.(17), just because,

$$\frac{F_{Sun}}{F_{\oplus}} \propto \frac{d_{\oplus}^2}{d_{Sun}^2} \quad (6)$$

Comparing Fig.(15) and Fig.(17), we see the pronounced lower peaks correspond when the Moon get a minimum in both the distances to the Earth and the Sun. The other not so pronounced minimums in Fig.(15) come to be when a maximum in the Moon-Earth distance coincides with a minimum from the Moon-Sun distance.

When we studied the Moon stability before, when the Sun was free, we saw it was quite easy to change its path when changing its initial conditions. Once we fixed the Sun in the centre, knowing we could get a not so chaotic result, we tried to figured out how much do we have to change this conditions so the Moon didn't orbit the Earth any more.

As we see in Fig.(18), the more we get the Moon away from the vertical at the beginning of its path, the less it orbits regularly around the Earth. It starts changing its period, as we see in green. Then it makes wider orbits, represented in magenta, until we get to 1.2° , the red curve, when it gets lost around the Sun and gets away from the Earth's path.

As an example of a case where the Moon could actually change its course 1.2° , we've proposed a meteorite collision, as schemed in Fig.(19).

When making an easy analysis of the momentum in a perfect inelastic collision, we get that the mass and velocity of a meteorite that may crash into the Moon to make it change its path 1.2° has to be $\frac{m_{moon}}{m_{meteorite}} = 1.5 \times 10^3$ and $\frac{v_{meteorite}}{v_{moon}} = 31.8$ (the angle of the meteorite with respect to the vertical

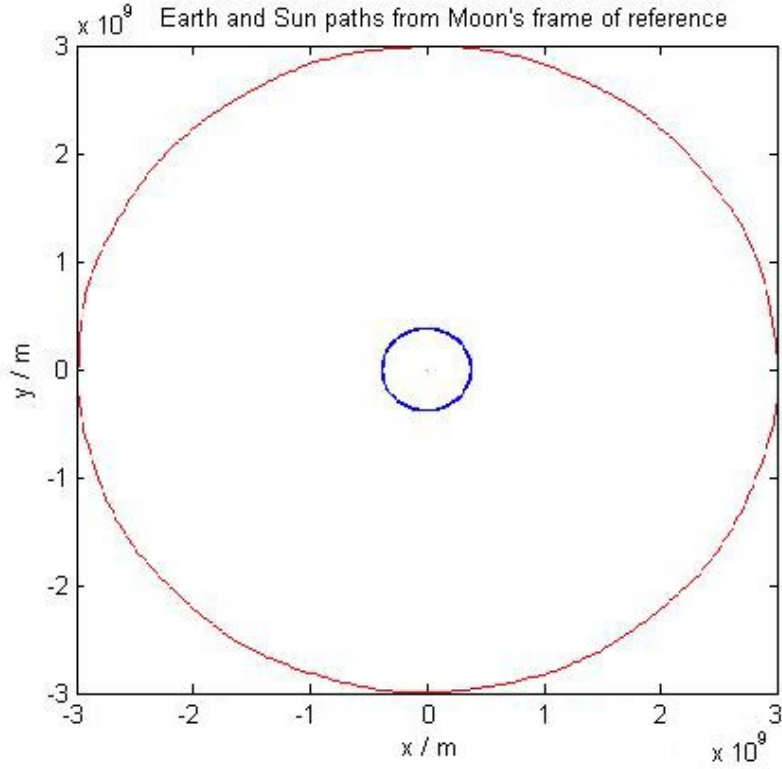


Figure 16: Earth (blue) and Sun (red) orbits from the Moon's frame of reference. Sun orbit has been plotted fifty times smaller so it can be properly compared with the Earth's orbit.

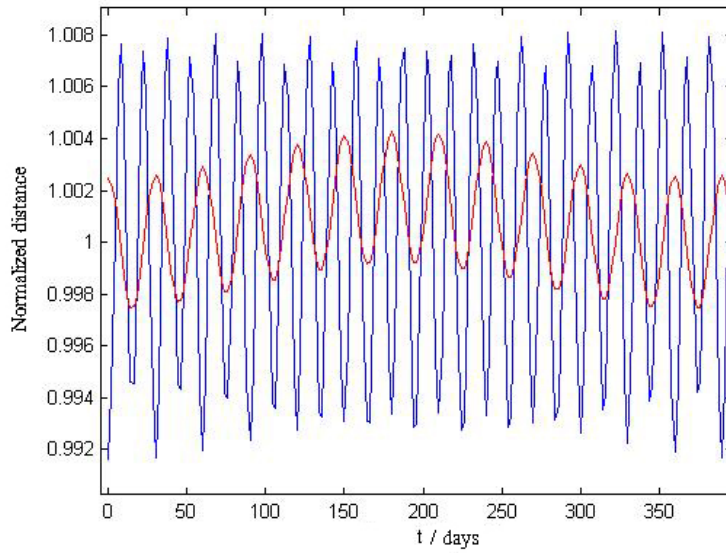


Figure 17: Comparison of the distance between Earth and Moon (blue) and Sun and Moon (red). Both distance curves have been normalized to their own mean, so they can be compared. The red and blue peaks phase describes quite well the peaks progression on Fig.(15).

was taken as $\theta = (90 - 1.2)^\circ$. So this result means that a quite fast rock a thousand times "smaller" than the Moon, would be enough to take the Moon out of its path around the Earth.

We shouldn't forget that this is just a calculation over a rough approximation, but that gives us

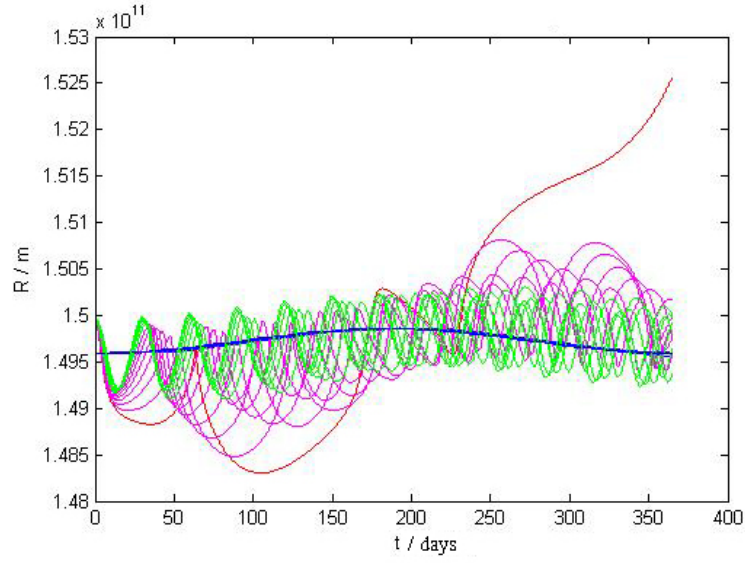


Figure 18: Distance between the Sun and the Earth (on blue) and the Sun and the Moon (on different colours) for different initial angles for the Moon velocity. The green curves are for angles between 90° and 89.4° , the magenta for angles between 89.4° and 88.9° , and the red curve is for an angle of 88.8° .

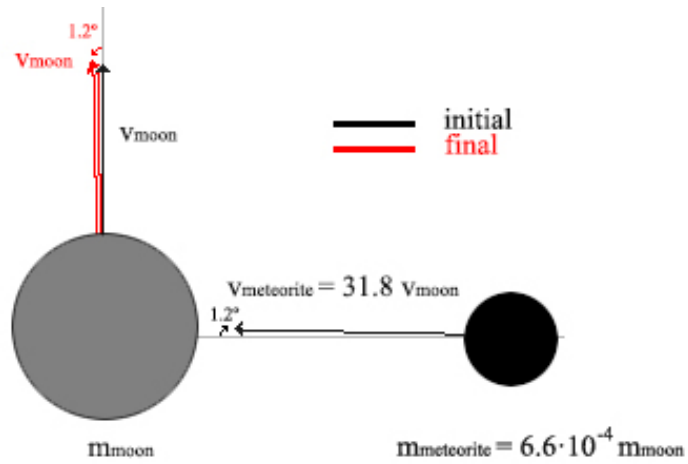


Figure 19: Scheme for the exercise of searching velocity and mass of a meteorite that would change the Moon path for 1.2° .

a very well idea of the complexity on the motion of the three-body problem.

4 The three body problem and the Earth-Satellite-Moon system

After all the tests on the program since we started by implementing the Runge-Kutta algorithm, we were quite sure that it worked properly for the approximation we chose and while the bodies didn't get too close to each other. The last thing we wanted to pay attention to was the Satellite-Earth-Moon system.

While when studying the Sun-Earth-Moon system there wasn't much realism in changing the initial conditions for each body from the real case perspective, when working with a body that represents an artificial satellite it has much more sense to change its conditions, depending on which orbit we want it to loop around. We focused on the influence of the Moon on the satellite, and how its path changes when it gets too close to our natural satellite.

We kept the most massive body fixed approximation. In this case, it's not such a good approximation, keeping the Earth motionless, as it was with the Sun. However, because the satellite mass was negligible in this case, our system was equivalent to the Earth-Moon centre of mass system, working with the reduced masses.

Before we started sweeping the space between the Earth and the Moon with the satellite, we wanted to take a look to the geostationary case, so we made sure we had enough control over the system, and we could claim we were able to predict what happened in the easiest case.

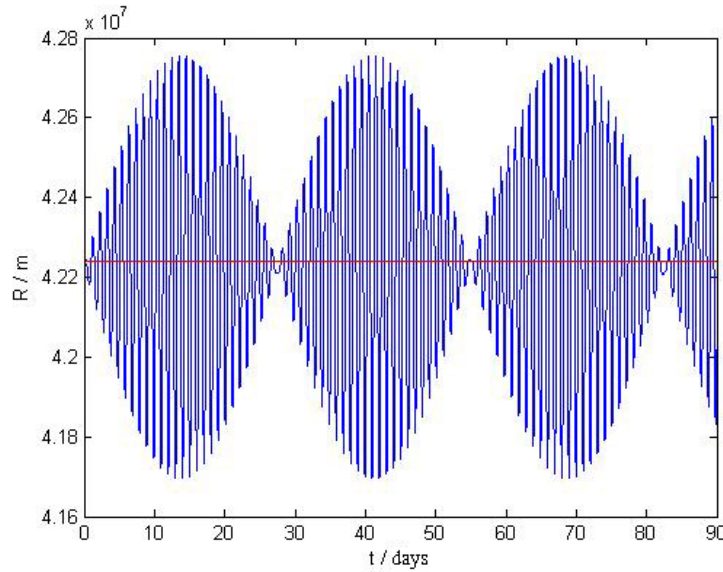


Figure 20: Distance between the satellite and the Earth along three months. The red line is the distance from the Satellite to the Earth when there's no Moon, while the blue line is the distance between satellite and Earth when there's Moon.

The configuration chosen for the bodies to start from for the examples on the Earth-Satellite-Moon system was the Moon eclipse formation, presented in Fig.(8), where the previous Sun, Earth and Moon, were now represented by the Earth, the satellite and the Moon, respectively. As we see in Fig.(20), the satellite, while orbiting the Earth in the geostationary orbit, was perturbed by the Moon, so it didn't make perfect circles around the Earth, and had a twenty eight days perturbation period (coinciding with the Moon period around the Earth) that made it have a delay of two days every thirty days.

This delay in the period of the satellite comes from the same reason we saw before for the Earth. As we saw in Eq.(5), when introducing the third body in the system, the one looping around the massive one gets delayed because of the conservation of the angular momentum.

In this case, the satellite needs an average of one hour and forty three minutes more than a day to finish a loop. As we see in Fig.(20), this daily period changes depending on the Moon's period. When the Moon is in the middle of the period, the satellite needs some more time to finish its loop (its distance to the Earth increases), while when the Moon finishes its period the satellite has a one day period.

Taking a look into stability, we've simulated over ten years of inside simulation time. The satellite remained in a stable orbit, but as we'll see in later examples, we can't guarantee the stability of the satellite for a long period of time. However, because the Moon is far enough from it in this example, it's likely to remain stable in the geostationary orbit.

The next exercise was to search for the farthest point from the Earth where the Satellite was still stable for around a month before the Moon perturbed enough its motion.

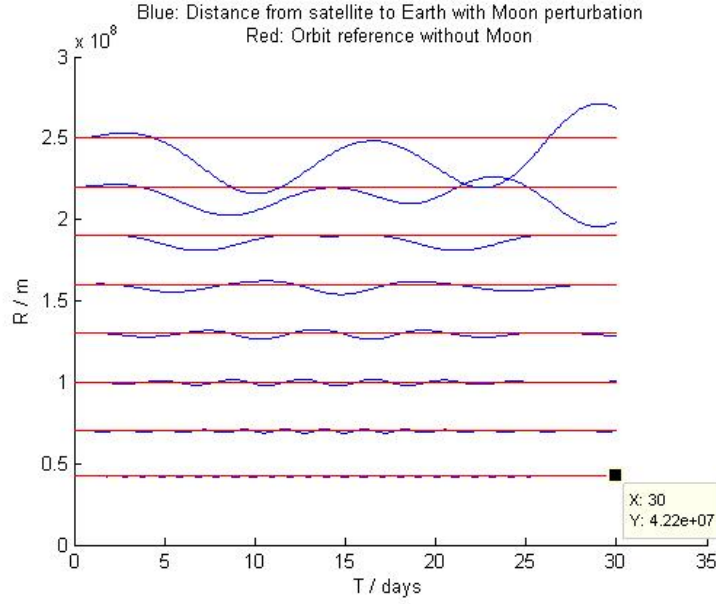


Figure 21: Distance between the satellite and the Earth for different orbits (blue) and the reference orbit (red) if they were perfectly circular because there was no Moon. The first orbit plotted (and marked with its (X,Y) values at the end) is the geostationary.

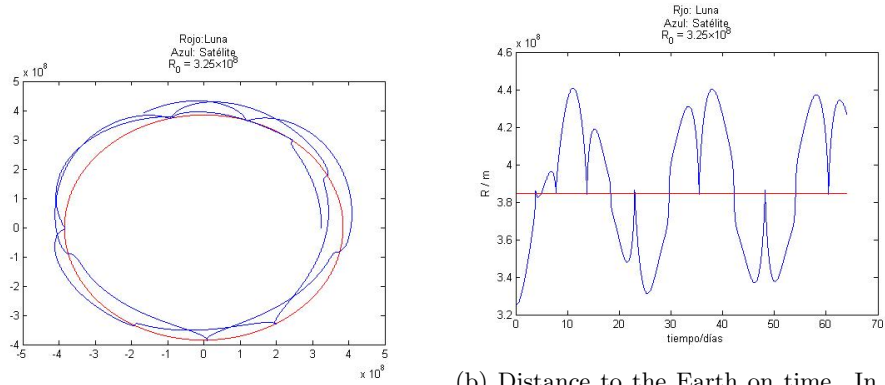
As we see in Fig.(21), the farthest from the Earth we shoot the satellite, the more inhomogeneous the orbit is, until around $R \approx 2 \times 10^8$ meters, when the orbit gets chaotic. In Table (2) we can see quantitatively the discrepancies on the orbit paths.

$R_0 \times 10^7 / m$	4.22	7	10	13	16	19	22	25
$\frac{\delta R_{max}}{R_0} \times 10^5$	5.38	12.23	23.11	37.12	60.19	90.95	243.82	341.72

Table 2: Normalized discrepancies of the orbits presented on Fig.(21). δR_{max} is the biggest discrepancy between the blue and the red lines for each orbit.

As we've already mentioned, we can't ensure the stability of any of these orbits, not even the nearest to the Earth. We can just say they are stable for at least thirty days when the orbit is less than $R \approx 2 \times 10^8$ meters from the Earth.

The orbits starting from a distance over two hundred million meters away from the Earth, illustrate perfectly the main problem on the three-body problem, the unpredictability of the motion of the bodies along time. We've chosen the example of the orbit starting at $R_0 = 3.25 \times 10^8$ meters as an example.

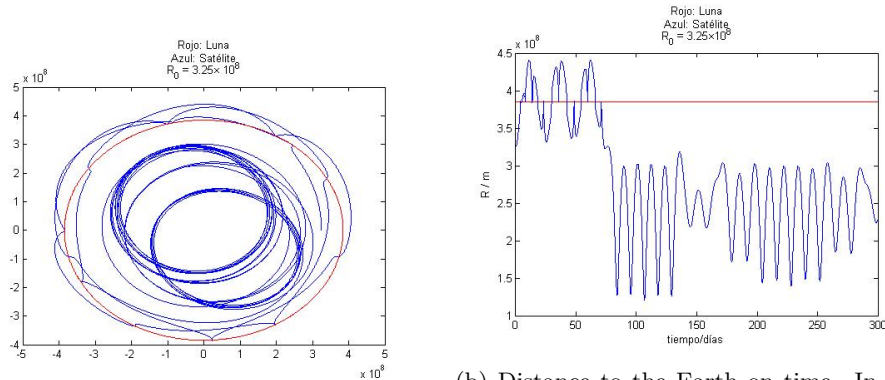


(a) In red, the moon path, while in blue, the satellite path is presented.

(b) Distance to the Earth on time. In red the Moon distance, while in blue, the satellite distance.

Figure 22: Distance from the satellite and Moon to Earth starting the satellite from a distance of $R_0 = 3.25 \times 10^8$ meters.

As we see in Fig.(22), it seems the satellite has entered a bounded orbit somewhere around the Moon. But, when exploring the orbit for some more time, as we can see in Fig.(23), the satellite suddenly drops into another orbit and starts looping around, between the Earth and the Moon without a trivial order.



(a) In red, the moon path, while in blue, the satellite path is presented.

(b) Distance to the Earth on time. In red the Moon distance, while in blue, the satellite distance.

Figure 23: Distance from the satellite and Moon to Earth starting the satellite from a distance of $R_0 = 3.25 \times 10^8$ meters.

Because it seemed that during the first seventy days the satellite orbited the Moon in some strange way we plotted the distance from the satellite to the Moon, in order to see what would be like watching the satellite from the Moon.

As we see in Fig.(24b), the orbits around the Moon weren't circular at all and the satellite actually got really close to the Moon position. This may make us think that the orbit shape is highly influenced by the program error, as we see in previous sections. However, for this simulation we've used a $h = 10^{-5}$ days (which is equivalent to $h = 0.864$ s), and making use of Table (1), although these values were for the two body problem case, we can make ourselves an idea that the error gathered step by step isn't that important, in principle, as to change the whole orbit path.

Every time the satellite gets really close around the Moon in Fig.(24), we see in Fig.(22) that its path suddenly changes, represented by a peak. During these peaks we suspected that something like a transference in momentum between the Moon and the satellite shall happen. As we see in Fig.(25),

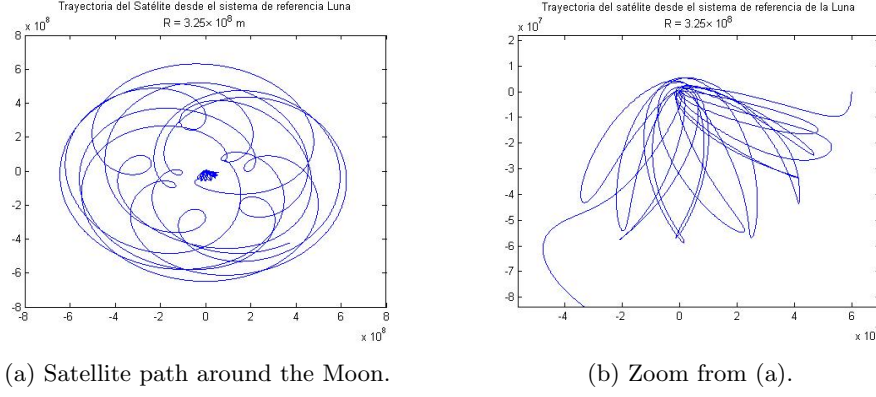


Figure 24: Distance from the satellite to the Moon starting the satellite from a distance of $R_0 = 3.25 \times 10^8$ meters to the Earth.

corresponding with each peak from the seventy three first days, the satellite gets a boost in its speed. This boost comes from the Moon, which, being much more massive than the satellite, transfers part of its momentum to the satellite, when this one gets close enough to it.

Later in the simulation, the satellite disengages from the weird orbits around the Moon we see in Fig.(24b) and starts to go around somewhere around the centre of masses of the whole system. The peaks in the speed we see in this part of the simulation aren't as pronounced as the ones from before, and just come from the gravitational pull the satellite suffers from the centre of masses.

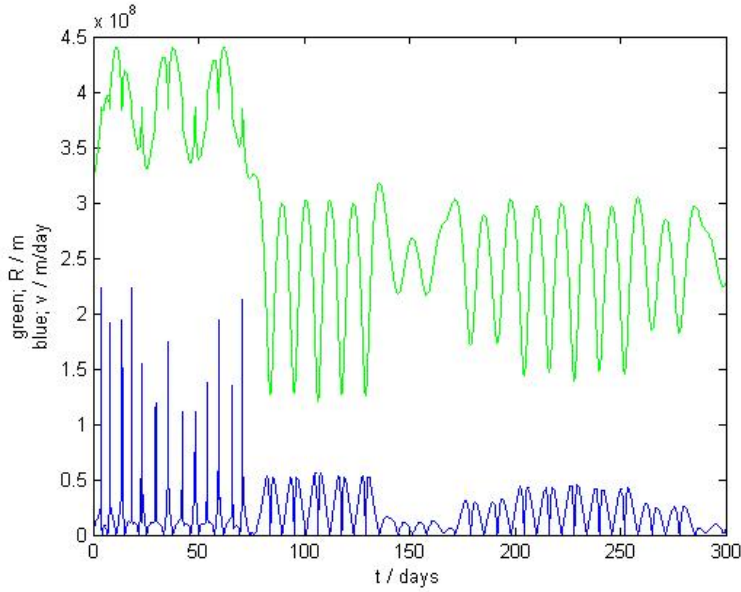


Figure 25: Distance between the satellite and the Earth for an initial orbit of $R = 3.25 \times 10^8$ m (green) and its corresponding speed in absolute value (blue) along time.

The example for the $R = 3.25 \times 10^8$ m orbit, at least for the time we have plotted, described a bounded orbit. However, most of the times we looked randomly for an orbit, we found that, after a transference of momentum because the satellite got really close to the Moon, it just escaped away from the system. Sometimes, the satellite just got enough energy from the Moon as to escape the first time it passed near it, as in Fig.(26), and other times it orbited some time around the system and suddenly got away, as in Fig.(27).

From these other two examples we see that, apart from the unpredictability of the motion of the satellite once it gets close enough to the Moon, we can also say its bound to the system is usually uncertain in these cases.

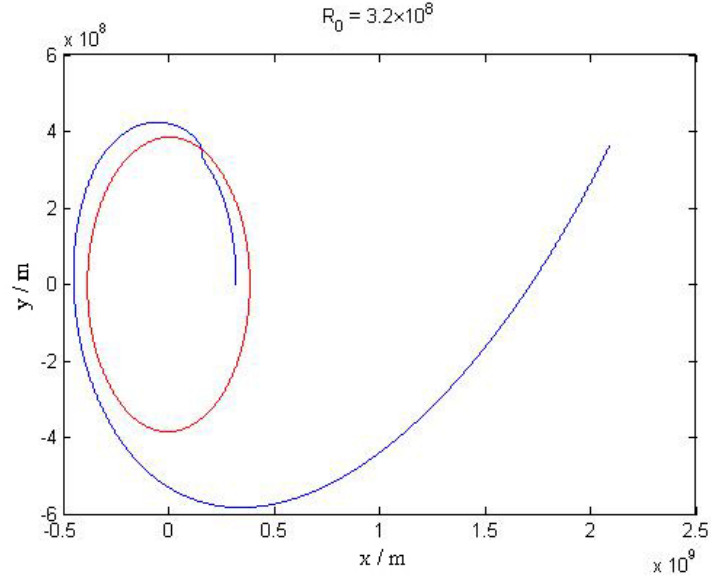


Figure 26: Orbit path of the satellite beginning from $R = 3.2 \times 10^8$ m (blue) and Moon's path around the Earth (red).

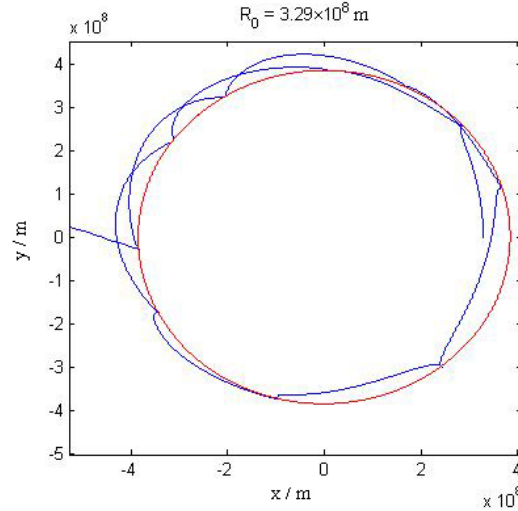


Figure 27: Orbit path of the satellite beginning from $R = 3.29 \times 10^8$ m (blue) and Moon's path around the Earth (red).

From this last section about the three-body problem we get as conclusions that it seems that the effect of the Moon on the satellite when they aren't too close (the critical distance from the Earth was around $R \approx 2 \times 10^8$ m) is small, apart from the delay that comes from the angular momentum conservation. The program returns reasonable solutions for the geostationary orbit as well. Once the satellite gets captured by the Moon, its motion gets chaotic, so it doesn't seem viable to throw a free satellite that near to the Moon.

The tool we have developed, using a small enough h to avoid too much error stack, in spite of its simplicity, has resulted very useful in order to study these kind of problems.

5 Conclusions

It's important to remark that we've develop a program to solve the three-body problem as an example. The physics of the problem aren't as important as the numerical approach we've used for its resolution. We've got several conclusions, achievements and learnings through the development of the project, which have already been commented in the previous sections and that will be gathered in this one.

1. We have implemented several codes in Matlab to solve ODE systems, using the fourth order Runge Kutta algorithm, so we could study the three-body problem. We took daily known examples to check the results we got. The codes were made in increasing complexity, starting with the two-body problem of the Sun-Earth system. We continued working with the three-body problem taking the Sun-Earth-Moon system and the Earth-Moon system with an artificial satellite. We got reasonable results within the limits of our approximations. We can say the implementation was successful, considering our results agreed with our expectations.
2. We have used two different approaches for the Sun-Earth-Moon system, letting the three bodies free and fixing the Sun, achieving the following results:
 - 2.1. We've done a fine-tuning to get the simulated system to behave in a similar way the real system does, but getting the orbits more circular in order to simplify the interpretations of the results.
 - 2.2. The initial conditions we had to set when the three bodies were free were so fine that, for example, changing the Earth's speed $\delta v_{\oplus} = 1$ m/s, made its period change $\delta T_{\oplus} = 1$ hour. When we set the initial conditions so the bodies had to move in circular motion, we got that Earth and Moon disturbed each other. The Earth got an eccentricity of $\epsilon_{\oplus} = 0.05$ and the Moon $\epsilon_M = 0.28$.
 - 2.3. We've make an attempt to study the stability of the system, and, during two hundred years, it hasn't seem to have any trend to change.
 - 2.4. We've study the Moon's stability when making little perturbations in its orbit, while the other conditions remained the same. We got that the Moon needs to change its path $\delta\theta = 1.2^\circ$ in order to disengage from the Earth's orbit.
3. We have simulated an artificial satellite orbit around the Earth and the Moon, with the following results:
 - 3.1. Comparing the geostationary satellite with and without Moon, we got that, because of the Moon influence, it had a delay of around two hours every day.
 - 3.2. We made a study of the orbits for different distances to the Earth. We got that while the satellite remains near the Earth (not further than $R^{**} \approx 2 \times 10^8$ m), there are periodical perturbations in the orbit with respect to the circular case, but they don't seem to present any trend to change its orbit.
 - 3.3. When the satellite gets trapped by the Moon its motion gets chaotic. In this stage the satellite usually gets so close to the Moon that the error stack starts to be a problem. However, taking a small enough integration step, this error shouldn't mean any problem. When getting to close to the Moon, the satellite gets a boost in its speed due to momentum transfer. If the energy transferred results enough, the satellite escape from the system.
 - 3.4. Apart from the cases where the satellite and the Moon get too close to each other, where we may need an adaptive integration step, the program works well and allows us to make a detailed study of any case of the problem.
4. Taking everything into account, we've proved that we've developed a useful tool with reasonable results to solve the problems we have dealt with. In order to use it in a scientific environment, however, there are several things that must be improved:
 - 4.1. When two bodies come too close together ($R^* \approx 9 \times 10^6$), because of the nature of the gravitational force ($\propto \frac{1}{r^2}$), the code stacks more error than usual for the rest of the simulation. This effect can be minimized implementing an adaptive algorithm, so the integration step decreases when the two bodies get too close.

- 4.2. The code would run faster if the number of equations got decreased by treating the problem theoretically.
- 4.3. In order to make the code even faster, another computing program would be useful. It could also be prepared to be sent to more powerful computers and minimize the simulation time.
- 5. The study of this problem has been a very interesting topic. It also has been useful to refresh some computing concepts and learn a whole new. I have had to refresh how to solve the second order ordinary differential equations and some concepts from mechanics too. In addition, it has been the first big program I have developed, and the step by step process has been a really enriching experience.

6 References

Because of the simplicity with which we have wanted to develop this work, we haven't needed more bibliography than that for the implementation of the 4th order Runge Kutta and the initial conditions of the bodies for the Sun-Earth-Moon system.

The information presented in Section (2.1) was taken from:
[https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods#The_Runge.E2.80.93Kutta_method](https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods#The_Runge%E2.80.93Kutta_method)
(30/06/2015)

The initial conditions for distance, mass and initial speed for the Sun, Earth and Moon with respect to the Sun were taken from:

<https://en.wikipedia.org/wiki/Sun> (30/06/2015)
<https://en.wikipedia.org/wiki/Earth> (30/06/2015)
<https://en.wikipedia.org/wiki/Moon> (30/06/2015)

Matlab questions and problems were solved in:
<http://es.mathworks.com/help/matlab/> (30/06/2015)

For further reference contact the author: diegomartnlpez@gmail.com