Multi-satellite location system around the Earth with applications in space monitoring

Proposal of a mathematical model based on the numerical analysis of ordinary differential equations for the capture of satellite positions and velocities

Clemente Serrano

November 10th, 2018

1 Introduccion

The artificial satellites are born in the Cold War as a competitive resource during the battle of powers between the US and Russia. The bellicose environment of the time made the satellite a potential weapon for surveillance.

History tells that the first satellite, Sputnik I, was just sent to space by Russia on October 4, 1957. This was the milestone that marked the start of the aerospace race, moment from which to date, More than 20,000 objects have been recorded around the Earth, where more than 5,000 are of human production. [3]

Currently, aerospace development is a key element for the development of a country. Countries like Argentina and Brazil have invested millions of dollars in these technologies. In the case of Argentina, 270 million dollars were invested in 2014 and 1.3 million hours in the launching of the Arsat-1 satellite. This does not provide telecommunications services and data transmission to the same country. , but also to Chile, Uruguay and Paraguay [4,5]. The reason for this is that the satellites provide the basis for the operation of a large part (if not all) of the information technology, such as GPS, the internet, mega databases of business and government, cell phones, radio and television (among other devices). Another application whose base of information is provided by the weather is meteorology: through this, earthquakes, hurricanes, floods and other catastrophes can be detected early.

Chile does not have a civil space agency, so the satellites put in orbit as the

FASAT-ALFA, FASAT-Bravo and FASAT-Charlie have been built and sent to space by pa is foreigners who then sell the rights to Chile in order to have access to the benefits that this object entails. The FASAT Charlie, for example, was built by France and put into place by a Russian rocket from the country that armed it [6]. With this it is demonstrated that au remains much to develop in the space habitat. The independence and development of satellite technologies in Chile will be a key element for their political and economic growth. In addition, it is extremely useful to obtain environmental data, detect natural disasters and have access to greater marine surveillance.

During the next work, a satellite location system for spatial monitoring will be developed, based on the satellites ability to generate a global view of the terrestrial landscape. The idea is to generate a contribution in what could be one of the satellite applications for Chile in its development and aerospace independence.

Barely, what the system does is to predict what is the position and instantaneous speed of a satellite, so as to know at what moment it is fit to take a photograph. Now, why is this necessary? For the proper capture of an image, it is necessary to have at least four satellites with a certain range of visibility as compared to the objective, so if you have several satires at a time, you need to know be able to take a photograph (specifically, when four of them are over a certain range of spatial coordinates). The project envisages putting seven satellites in a row simultaneously.

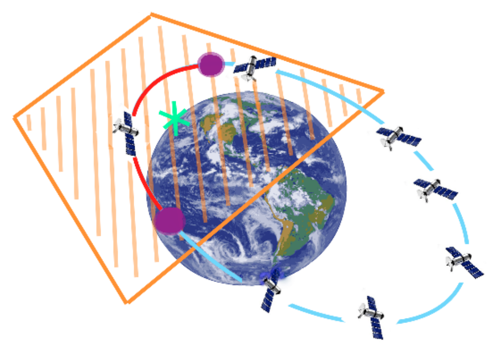

Figure 1: System of satellite distribution of the project (where the red trajectory is the curve of feasible positions and the orange tangent plane the spatial delimiter of the feasible curve).

The mathematical model that sustains the project is based on the numerical solution of the differential equations given by Newton's Law of Universal Gravitation, for initial conditions previously specified (positions and velocities in a particular time). To verify that the 3 satellites are in the right positions, they are drawn to a tangent plane to the Earth as an inertial frame of reference. The calculation engine will be the MATLAB software, which in addition to solving the differential equations numerically, simulate the process completely and indicate the positions and speeds of each satellite to make the predictions.

2 Background

During the development of the project, theories and methods were used, which must be mentioned and introduced. This helps to better understand the work done.

2.1 Tangent plane of the Earth-Satellites system

A plane is a surface formed by infinite points whose function is obtained from a point and a vector perpendicular to it.

Let π be a plane containing the point $P(x_1, y_1, z_1)$ and a non-zero vector perpendicular to it $n\langle a, b, c\rangle$. The plane contains the points $Q(x, y, z)$ where the vector $\overrightarrow{PQ}$ is orthogonal an, the equation of the plane is obtained from the product point of these vectors, considering that these are orthogonal You have the following:

$$n \cdot \overrightarrow{PQ} = 0$$

$$\langle a, b, c\rangle \cdot \langle x - x_1, y - y_1, z - z_1\rangle = 0$$

$$a(x - x_1) + b(y - y_1) + c(z - z_1) = 0$$

Finally, the general form of the equation of a plane in space can be obtained by regrouping terms, this is seen in the following way:

$$ax + by + cz + d = 0$$

Seen graphically, the obtained plane is the following one:

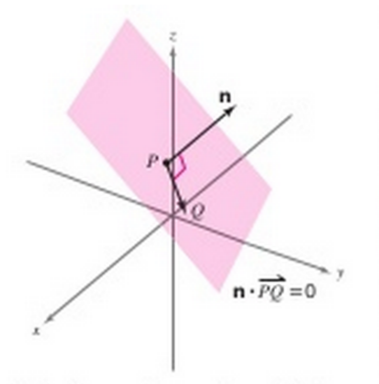

Figure 2: Plano formado por vector n ortogonal a $\overrightarrow{PQ}$

Let the vector function $r : \mathbb{R} \mapsto \mathbb{R}^3$ be defined as $r(t) = x(t)\mathbf{i} + y(t)\mathbf{j} + z(t)\mathbf{k}$ a smooth curve in an open interval I ($\dot{r}(t)$) is continuous and nonzero in I) that describes the trajectory of a particle or a moving object. Its unit vector tangent $T(t)$ in t can be described as follows:

$$T(t) = \frac{\dot{r}(t)}{\|\dot{r}(t)\|} \text{ while } \dot{r}(t) \neq 0$$

There are several vectors orthogonal to the tangent vector $T(t)$. One of these vectors is $\dot{T}(t)$, this seen in:

$$T(t) \cdot \dot{T}(t) = 0$$

Where $T(t)$, $\dot{T}(t)$ are different from 0. Normalizing $\dot{T}(t)$ the normal unit vector is obtained.

If $\dot{T}(t) \neq 0$, then the normal unit vector in t can be defined as:

$$N(t) = \frac{\dot{T}(t)}{\|\dot{T}(t)\|}$$

The tangent plane is that plane that contains the tangent vector. This plane is orthogonal to the normal vector of the curve at the point $P_0 = \mathbf{r}(t_0) = (r_1(t_0), r_2(t_0), r_3(t_0))$. The equation of the tangent plane is the following:

$$\langle (x, y, z) - (r_1(t_0), r_2(t_0), r_3(t_0)), N(t_0) \rangle = 0$$

Seen in another way:

$$\langle \mathbf{x} - \mathbf{r}(t_0), N(t_0) \rangle = 0$$

The tangent plane forms the Earth-Satellites system, in this plane is the tangent zone feasible, this is seen in the following image:

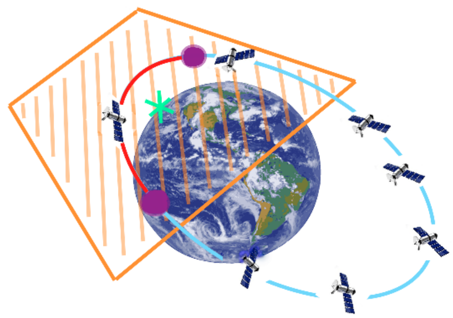

Figure 3: Sketch of the Earth-satellites system: orbits and feasible tangent zone.

2.2 The equation of the ellipse and its relation to the satellite movement

2.2.1 The ellipse

An ellipse can be defined as the set of all the points in the plane, whose sum of distances to two fixed points, which are called foci, is constant. The line that joins the foci intersects the ellipse in two points, which are called vertices. The rope that joins the vertices is the major axis and its middle point is the center of the ellipse. On the other hand, we also have the minor axis of the ellipse, which is a notebook that passes through the center and that is perpendicular to the major axis.

The equation of the ellipse, with center (h, k) and axle lengths greater $2a$ and minor axis $2b$, where $a > b$, is as follows:

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1$$

And the equation of the ellipse, with center (h, k) with lengths of the major axis $2b$ and minor axis $2a$, where $b > a$, is the following:

$$\frac{(x-h)^2}{b^2} + \frac{(y-k)^2}{a^2} = 1$$

The spotlights of the ellipse are located on the major axis, at c units from the center, with:

$$c^2 = a^2 - b^2$$

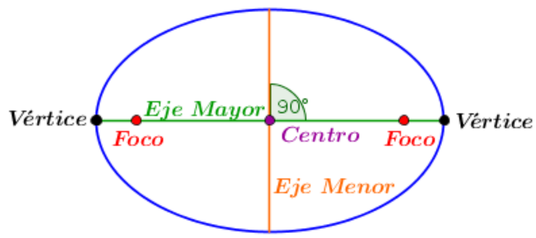

Figure 4: The ellipse and its elements.

One of the applications of the ellipse, which relates it to this project, is the ability it has to describe the movements of the planets and in this case, the satellites.

The satellites that orbit around the Earth describe elliptical trajectories. The force responsible for this movement is the gravity attraction of the Earth. In its trajectory, the satellite crosses two points of the Earth: the Perigee (here the satellite is as close as possible to the Earth) and the Apogee (the satellite is at the maximum possible distance from the Earth).

The eccentricity (degree of deviation of the conic section with respect to a circumference) of the orbit of the Earth is very small, this being almost a circular orbit (it is less than 0.2). It is said that the more circular the orbit is, the more perfect it is. On the other hand, the inclination indicates how much an orbit is inclined. An inclination of 0 degrees is when an orbit is just in the plane of the Equator, and 90 degrees when the orbit passes through the North and South poles.

2.3 La mecanica de Newton

The postulates proposed by Lagrange and Hamilton to address the problem of n bodies take as a formative basis the so-called *Law of Universal Gravitation* by Isaac Newton. Because of this, both Hamiltonian and Lagrangian methods allow reaching the same results as Newtonians. The fact that more complex mathematics are used and more global physical concepts are addressed, allows us to affirm that the mechanics of Lagrange and Hamilton are a more elegant reformulation of the law of universal gravitation.

From Newton's second law,

> *'Every force applied to a body generates an acceleration, which is directly proportional to the force and inversely proportional to the mass of the body'.*

the law of universal gravitation proposes that there is a force of action at a distance F_g which is generated by the mere fact of two or more masses m_1, m_2, $\ldots m_n$ are in the same space. Specifically, for the case of two bodies, the modulus of this force is inversely proportional to the square of the distance r between the two bodies and directly proportional to the product of their masses. Mathematically, this means:

$$|\mathbf{F}| \propto \frac{m_1 m_2}{r^2}$$

Now, if this expression is multiplied by a constant, the relation of proportion becomes an equality. For this then:

$$|\mathbf{F_g}| = G\frac{m_1 m_2}{r^2}$$

Denoted as G, this constant is called *the universal gravitational constant* and is defined as:

$$G = 6.67384 \cdot 10^{-17}\ \frac{\mathrm{N} \cdot \mathrm{km}^2}{\mathrm{kg}^2}$$

But knowing the module of the gravitational force often does not provide all the necessary information to analyze a physical process. In $\mathbb{R}^3$, this magnitude can be represented in a vectorial way by simply amplifying by the module $|\mathbf{F_g}|$ a unit vector $\hat{r}$ in the direction in which the force of attraction of one body is applied to the other (remember that the line of force in this type of case is directed from one center of mass to the other). Therefore, considering that $\hat{r} = \mathbf{r}(x, y, z)/\|\mathbf{r}\|$ then:

$$\mathbf{F}_g = \frac{G m_1 m_2}{r^2} \cdot \hat{r} \tag{1}$$

See figure 5. The masses m_1 (red) and m_2 (blue) are located according to the Cartesian coordinates (x_1, y_1, z_1) and (x_2, y_2, z_2) respectively in the space. The mere fact of both being present, or in other words of having mass, generates a mutual attraction represented as the force $\mathbf{F}$ (bidirectional gray line). Now, Why is it bi-directional? For due to the principle of action and reaction (Newton's Third Law) it acts with the same intensity for both the mass m_1 with respect to m_2 and for the mass m_2 with respect to m_1 only with contrary senses.

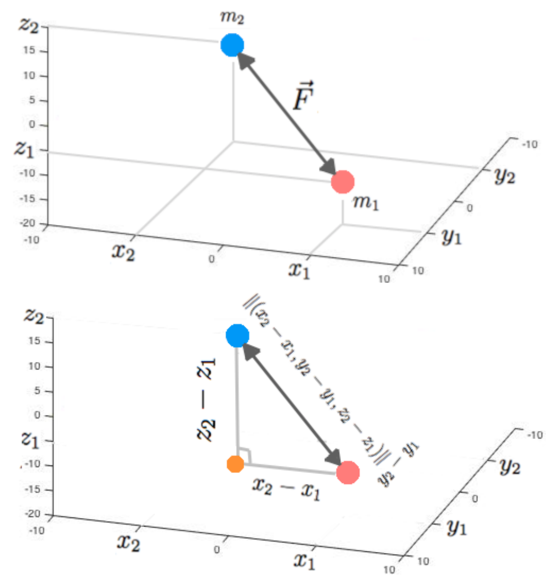

Figure 5: The problem of the two bodies from the Newtonian approach.

The unit vector $\hat{r}$ of the last defined equation can be defined through the analysis of the positions of the bodies with respect to each other. First, the unit position vector for m_2 with respect to m_1 will be simply defined as the difference between the coordinates of m_2 and those of m_1 on the module of that vector, that is:

$$\hat{r}_{m_2 \to m_1} = \frac{(x_2 - x_1, y_2 - y_1, z_2 - z_1)}{\|(x_2 - x_1, y_2 - y_1, z_2 - z_1)\|}$$

In the same way, the unit vector representative of the position of m_1 with respect to m_2 would be then:

$$\hat{r}_{m_1 \to m_2} = \frac{(x_1 - x_2, y_1 - y_2, z_1 - z_2)}{\|(x_1 - x_2, y_1 - y_2, z_1 - z_2)\|}$$

Considering the above, then the equation () can be reproposed for each of the masses with respect to the other in the following way:

i. For the mass m_1 with respect to m_2:

$$\mathbf{F}_{g_{m_1 \to m_2}} = \frac{Gm_1m_2}{\|(x_1 - x_2, y_1 - y_2, z_1 - z_2)\|^2} \left(\frac{(x_1 - x_2, y_1 - y_2, z_1 - z_2)}{\|(x_1 - x_2, y_1 - y_2, z_1 - z_2)\|} \right)$$

$$= \frac{Gm_1m_2\Big(x_1 - x_2, y_1 - y_2, z_1 - z_2\Big)}{\|(x_1 - x_2, y_1 - y_2, z_1 - z_2)\|^3}$$

ii. For the mass m_2 with respect to m_1:

$$\mathbf{F}_{g_{m_2 \to m_1}} = \frac{Gm_1m_2}{\|(x_2 - x_1, y_2 - y_1, z_2 - z_1)\|^2} \left(\frac{(x_2 - x_1, y_2 - y_1, z_2 - z_1)}{\|(x_2 - x_1, y_2 - y_1, z_2 - z_1)\|} \right)$$

$$= \frac{Gm_1m_2\Big(x_2 - x_1, y_2 - y_1, z_2 - z_1\Big)}{\|(x_2 - x_1, y_2 - y_1, z_2 - z_1)\|^3}$$

The properties of vectorial algebra allow to decompose the equations described above for each component of the position, that is, the gravitational force on the axis x, on the axis y and on the axis z. To simplify the index notation, it will be considered that $\mathbf{F}_{m_{12}x}$, $\mathbf{F}_{m_{12}y}$ and $\mathbf{F}_{m_{12}z}$ are the force components for the mass m_1 relative to m_2 and that $\mathbf{F}_{m_{21}x}$, $\mathbf{F}_{m_{21}y}$ and $\mathbf{F}_{m_{21}z}$ are the force components for the mass m_2 as a function of the mass m_1. Considering the previous:

$$\mathbf{F}_{m_{21}x} = \frac{Gm_1m_2\,(x_2 - x_1)}{\|(x_2 - x_1, y_2 - y_1, z_2 - z_1)\|^3} \qquad \mathbf{F}_{m_{21}y} = \frac{Gm_1m_2\,(y_2 - y_1)}{\|(x_2 - x_1, y_2 - y_1, z_2 - z_1)\|^3}$$

$$\mathbf{F}_{m_{21}z} = \frac{Gm_1m_2\,(z_2 - z_1)}{\|(x_2 - x_1, y_2 - y_1, z_2 - z_1)\|^3} \qquad \mathbf{F}_{m_{12}x} = \frac{Gm_1m_2\,(x_1 - x_2)}{\|(x_1 - x_2, y_1 - y_2, z_1 - z_2)\|^3}$$

$$\mathbf{F}_{m_{12}y} = \frac{Gm_1m_2\,(y_1 - y_2)}{\|(x_1 - x_2, y_1 - y_2, z_1 - z_2)\|^3} \qquad \mathbf{F}_{m_{12}z} = \frac{Gm_1m_2\,(z_1 - z_2)}{\|(x_1 - x_2, y_1 - y_2, z_1 - z_2)\|^3}$$

Following is a fundamental question which allows us to associate this part of Newton's mechanics with the theme of this work: How to determine at any time of the dynamics of the system the position and speed of one body with respect to the other? How to parametrize these equations as a function of time? The answers are simple: **must take the problem to its differential form and then integrate these equations to obtain primitives as a function of time t**.

The key to all this is in *Newton's Second Law*.

$$\mathbf{F} = m\mathbf{a} \quad \text{such that} \quad \mathbf{a} = \text{acceleration}$$

then the vectorial equations of the gravitational force can be proposed in the following way (consider a position vector $\mathbf{r}(t)$ generalized which refers to the body I or II according to the mass that multiplies its second derivative respect weather):

$$m_1 \frac{d\mathbf{r}}{dt^2} = \frac{Gm_1m_2\Big(x_1 - x_2, y_1 - y_2, z_1 - z_2\Big)}{\|(x_1 - x_2, y_1 - y_2, z_1 - z_2)\|^3}$$
$$m_2 \frac{d\mathbf{r}}{dt^2} = \frac{Gm_1m_2\Big(x_2 - x_1, y_2 - y_1, z_2 - z_1\Big)}{\|(x_2 - x_1, y_2 - y_1, z_2 - z_1)\|^3}$$

If again these equations are decomposed according to the directional axis considered, then the following ODE's system is obtained:

$$
\begin{cases}
m_1 \frac{d^2\mathbf{x_1}}{dt^2} = \frac{Gm_1m_2\left(x_1-x_2\right)}{\|(x_1-x_2,y_1-y_2,z_1-z_2)\|^3} \\
m_1 \frac{d^2\mathbf{y_1}}{dt^2} = \frac{Gm_1m_2\left(y_1-y_2\right)}{\|(x_1-x_2,y_1-y_2,z_1-z_2)\|^3} \\
m_1 \frac{d^2\mathbf{z_1}}{dt^2} = \frac{Gm_1m_2\left(z_1-z_2\right)}{\|(x_1-x_2,y_1-y_2,z_1-z_2)\|^3} \\
m_2 \frac{d^2\mathbf{x_2}}{dt^2} = \frac{Gm_1m_2\left(x_2-x_1\right)}{\|(x_2-x_1,y_2-y_1,z_2-z_1)\|^3} \\
m_2 \frac{d^2\mathbf{y_2}}{dt^2} = \frac{Gm_1m_2\left(y_2-y_1\right)}{\|(x_2-x_1,y_2-y_1,z_2-z_1)\|^3} \\
m_2 \frac{d^2\mathbf{z_2}}{dt^2} = \frac{Gm_1m_2\left(z_2-z_1\right)}{\|(x_2-x_1,y_2-y_1,z_2-z_1)\|^3}
\end{cases}
\tag{2}
$$

As previously mentioned, through these equations, given any time t any in the interaction of the two bodies can be determined:

i. Through the first integration of the speed of either of the two bodies.

ii. After the second integration, the position of either of the two bodies.

3 Development of the solution

As explained in previous chapters, the location system must fulfill a series of functions, within which are the capacity to determine as accurately as possible the trajectory that the satellites follow around the Earth, must have a mechanism that allows to indicate if the satellite is in the correct position to capture the image and must be able to calculate the time in which the satellite is in this condition.

Now, the first thing to do is simulate the orbitals of the trajectories followed by the satellites and integrate the functions of the orbitals. This was done numerically with the support of MATLAB.

The programmed algorithm consisted of several parts, which intertwined with one of the main functions find availability times to capture the images and other elements that determine the satellite movement.

3.0.1 Integrating the differential equations of satellite movement

To integrate the differential equations of satellite movement (defined in the theoretical framework, section 2.3) a mathematical treatment must be done to be able to adapt them to the context of this project. Based on the variables and equations obtained in the previous step, finally an algorithm will be programmed in MATLAB capable of obtaining all the necessary data so that the system that is modeled is effective and precise.

It must be remembered that the system model will be constructed for cases in which external forces do not influence.

From here on, the following nomenclature will be considered in replacement of the one previously considered:

i. m_S: mass of the satellite measured in Kg, amount previously named m_2.

ii. m_T: mass of the Earth measured in Kg, magnitude in replacement of the constant m_1.

iii. $\mathbf{r}_S$: satellite position vector measured in Km.

iv. $\mathbf{r}_T$: Earth position vector measured in Km.

v. $\mathbf{v}_S$: satellite speed vector, measured in Km/h

vi. $\mathbf{v}_T$: Earth speed vector, measured in Km/h

The transformation will consist in reducing the order of the problem of the bodies going from two to one. This means that the masses m_T and m_S will be considered as one. Below is how it is that mathematics allows this.

To depart, we invite you to review the vectorial ODEs of the bodies with respect to each other. It can be noticed that if it is multiplied by (-1), both equations arrive at the same:

$$
\begin{aligned}
m_T\frac{d^2\mathbf{r}_T}{dt^2} &= \frac{Gm_Tm_S\Big(x_T-x_S, y_T-y_S, z_T-z_S\Big)}{\|(x_T-x_S, y_T-y_S, z_T-z_S)\|^3} \\
-m_S\frac{d^2\mathbf{r}_S}{dt^2} &= \frac{Gm_Tm_S\Big(x_T-x_S, y_T-y_S, z_T-z_S\Big)}{\|(x_T-x_S, y_T-y_S, z_T-z_S)\|^3}
\end{aligned} \tag{3}
$$

This means that:

$$
\begin{aligned}
m_T\frac{d^2\mathbf{r}_T}{dt^2} &= -m_S\frac{d^2\mathbf{r}_S}{dt^2} \\
m_T\frac{d^2\mathbf{r}_T}{dt^2} + m_S\frac{d^2\mathbf{r}_S}{dt^2} &= 0
\end{aligned}
$$

Note that if the equation (.1) is multiplied by m_S and the (.2) by m_T to then subtract m_S(.1) with m_T(.2) a link can be established between them which says that:

$$
\begin{aligned}
m_Sm_T\frac{d^2\mathbf{r}_T}{dt^2} - m_Tm_S\frac{d^2\mathbf{r}_S}{dt^2} = &-\frac{Gm_T^2m_S\Big(x_T-x_S, y_T-y_S, z_T-z_S\Big)}{\|(x_T-x_S, y_T-y_S, z_T-z_S)\|^3} \\
&-\frac{Gm_Tm_S^2\Big(x_T-x_S, y_T-y_S, z_T-z_S\Big)}{\|(x_T-x_S, y_T-y_S, z_T-z_S)\|^3}
\end{aligned}
$$

Which means that,

$$
m_Sm_T\frac{d^2}{dt^2}\left(\mathbf{r}_S-\mathbf{r}_T\right) = -\frac{Gm_Tm_S\left(m_S+m_T\right)\Big(x_T-x_S, y_T-y_S, z_T-z_S\Big)}{\|(x_T-x_S, y_T-y_S, z_T-z_S)\|^3}
$$

Going back, product of the *The Third Law of Newton* or principle of action and reaction the force of attraction that m_T exerts over m_S is exactly the same in magnitude that the one exerted m_S over m_T but with the opposite sense $\mathbf{F}_{m_T\to m_S} = -\mathbf{F}_{m_S\to m_T}$, which means that:

$$
m_T\frac{d^2\mathbf{r}_T}{dt^2} = \frac{Gm_Tm_S\Big(x_T-x_S, y_T-y_S, z_T-z_S\Big)}{\|(x_T-x_S, y_T-y_S, z_T-z_S)\|^3}
$$

In the previous section it was considered that both bodies were in movement, however, for purposes of work only one is in motion: the satellite. This is a not least statement, since the system of equations changes completely. Why?

From here on, it will be considered that the mass of the Earth is m_T and that of the satellite m_S, therefore: $m_1 = m_T$ and $m_2 = m_S$. If one considers that the Earth is a static body for the system that is modeled, then it has no velocity, which implies that neither acceleration. Mathematically this is expressed as follows:

$$\begin{aligned}
\mathbf{r}_T &= \text{cte} && \text{la posicion de la Tierra no varia} \\
\frac{d\mathbf{r}_T}{dt} &= 0 && \text{la derivada de una constante es cero} \\
\frac{d^2\mathbf{r}_T}{dt^2} &= 0 && \text{la derivada de cero tambien es cero}
\end{aligned}$$

In parallel, for simplicity of calculations it will be considered that the spatial location of the planet Earth will be at the origin, that is to say that $x_T = y_T = z_T = 0$.

What is the impact of this on the mathematics proposed in the theoretical foundation? For all those differential equations that implied the position vector for the body m_2 (or otherwise m_T) are no longer necessary to consider and that all those Cartesian components related to the body m_2 are null. This means that the system () is modified to the following:

$$\begin{cases}
m_S m_T \frac{d^2}{dt^2}\left(\mathbf{x}_s - 0\right) = -G\frac{m_T m_S (m_T + m_S)}{\|x_s, y_s, z_s\|}\mathbf{x}_s \\
m_S m_T \frac{d^2}{dt^2}\left(\mathbf{y}_s - 0\right) = -G\frac{m_T m_S (m_T + m_S)}{\|x_s, y_s, z_s\|}\mathbf{y}_s \\
m_S m_T \frac{d^2}{dt^2}\left(\mathbf{z}_s - 0\right) = -G\frac{m_T m_S (m_T + m_S)}{\|x_s, y_s, z_s\|}\mathbf{z}_s
\end{cases}$$

Or simplified,

$$\begin{cases}
\frac{d^2\mathbf{x}_s}{dt^2} = -G\frac{m_T + m_S}{\|x_s, y_s, z_s\|}\mathbf{x}_s \\
\frac{d^2\mathbf{y}_s}{dt^2} = -G\frac{m_T + m_S}{\|x_s, y_s, z_s\|}\mathbf{y}_s \\
\frac{d^2\mathbf{z}_s}{dt^2} = -G\frac{m_T + m_S}{\|x_s, y_s, z_s\|}\mathbf{z}_s
\end{cases} \tag{4}$$

The undisturbed system integrator script contemplates the following:

A first step of the modeling process is to propose to the program what will be the problem that must be solved, that is, the differential equations to be integrated.

The main tool that will be used is the command *ode*45(). As such, this system integrates first order differential equations through the Runge-Kutta method of order four, therefore it approximates the solutions.

So what is done if the equations of the system are of second order? Well simply apply the method of order reduction, assigned new variables like u, v and w in the following way:

$$\begin{cases} \frac{d\mathbf{x}_s}{dt} = u \\ \frac{du}{dt} = -G\frac{m_T+m_S}{\|x_s,y_s,z_s\|}\mathbf{x}_s \\ \frac{d\mathbf{y}_s}{dt} = v \\ \frac{dv}{dt} = -G\frac{m_T+m_S}{\|x_s,y_s,z_s\|}\mathbf{y}_s \\ \frac{d\mathbf{z}_s}{dt} = w \\ \frac{dw}{dt} = -G\frac{m_T+m_S}{\|x_s,y_s,z_s\|}\mathbf{z}_s \end{cases} \tag{5}$$

As you can see, the system has been extended from 3×1 to 6×1.

Below is the script that structures this system:

```
function dr= movtierrasat(t,r)
    dr=zeros(6,1);
    G=6.67384*10^(-17);
    ms=1000;
    mt=5972*10^24;

    mu=(-G*(mt+ms))./((r(1).^2+r(3).^2+r(5).^2)^(3/2));

    dr(1)=r(2);
    dr(2)=mu.*r(1);
    dr(3)=r(4);
    dr(4)=mu.*r(3);
    dr(5)=r(6);
```

```
    dr(6)=mu.*r(5);
end
```

The way in which this works is the following:

i. First of all, this script is defined as an elementary one, that is, as a function whose name is *movtierrasat*.

Note that it receives two parameters: t and r, which represent the integration time and the initial position and velocity values in $\mathbb{R}^{\not\models}$ respectively.

ii. After that, we define the matrix (initially zero) of the differential equations together with a series of constants important as the mass of the Earth mt, the mass of the satellite ms and the universal gravity constant G.

There is a very important factor, constant in the equations, which is the product between the constant G and the sum of the masses. To reduce the complexity of the equations, this constant has been individualized in a single term like mu.

iii. Finally, the equations are defined in such a way that:

a) r(1), r(3) and r(5) represent the positions in x, *and* and z respectively.
b) r(2), r(4) and r(6) refer to the auxiliary variables u, v and w respectively.
c) dr(1), dr(2), dr(3) dr(4), dr(5) y dr(6) mean the first derivatives of the variables x, u, y, v, z and w respectively.

Next, a table will be presented showing the positions and speeds of the 7 orbiting satellites in the 3 coordinate axes:

Satellite	**x** (km)	**y** (km)	**z** (km)	$\mathbf{v}_x$ (km/h)	$\mathbf{v}_y$ (km/h)	$\mathbf{v}_z$ (km/h)
1	8667.7	-4226.2	3219.8	2965.8	4883.6	3415.6
2	-8309	-4231.2	4592.800	2993.8	-4313.6	-3876.6
3	-7567.7	-4351.2	3921.2	2565.8	-4330.9	4695.6
4	-8719.7	-3032.4	-3921.100	-3065.8	-4542.900	3915.600
5	3667.7	4426.2	8219.800	-3265.8	4483.600	-3415.600
6	-4367.7	-8426.2	-4319.800	-3265.8	-3483.600	4415.600
7	8598.7	4310.9	5519.800	1765.8	-4683.600	3615.600

Table 1: Positions and speeds of the seven satellites.

Now, the sentence that executes the resolution of the previously raised system of differential equations is the following, considering that satellite 1 will be used as an example, since for the 7 satellites the procedure is the same only that the initial conditions change:

```
[t1, r1]=ode45('movtierrasat',[0 200],
[8667.7, 2965.8, -4226.2, 4883.600, 3219.800, 3415.600]);
```

First, we use the MATLAB command called ode45, which integrates each system of differential equations, in a given time and with the following initial conditions: The initial position in x corresponds to $8667,7\ ,km$, in y corresponds to $-4226.2\ km$, and in z is $3219.8\ km$, with respect to the velocities one has to $\mathbf{v_x}$ is $2965.8\ km/h$, $\mathbf{v_y}$ is $4883.6\ km/h$ and lastly $\mathbf{v_z}$ is $3415,6\ km/h$. The order in which they are located in the algorithm is the following:

$$[x,\ \mathbf{v_x},\ y,\ \mathbf{v_y},\ z,\ \mathbf{v_z}]$$

Next, the images representing the satellite trajectories will be presented:

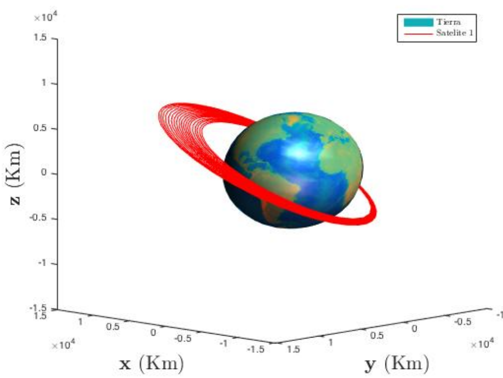

Figure 6: Satellite 1 trajectory

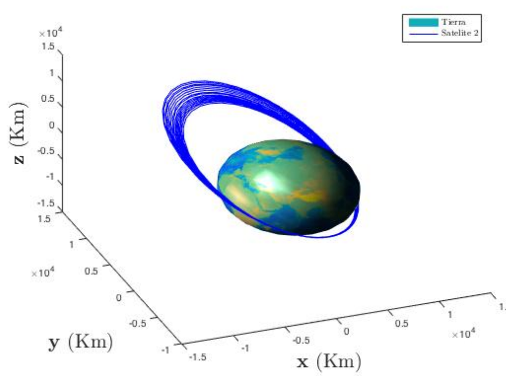

Figure 7: Satellite 2 trajectory

The first graph shows the behavior of the satellite around the Earth, where the trajectory is not affected by external forces.

The second graph shows the orbit of the second satellite, which, seen previously, does not have an external force that makes it vary.

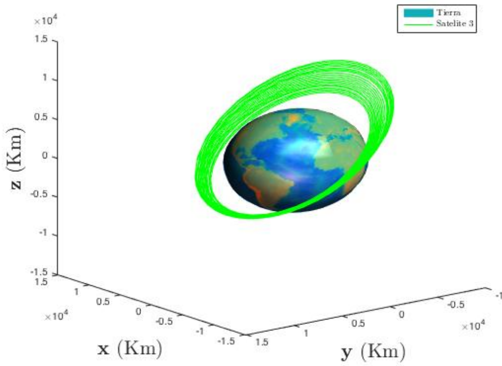

Figure 8: Satellite 3 trajectory

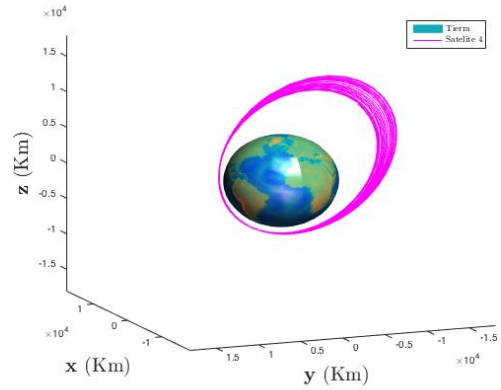

Figure 9: Satellite 4 trajectory

Then there are the graphs of satellites 3 and 4 and said above, these orbits do not have external forces that alter the orbit.

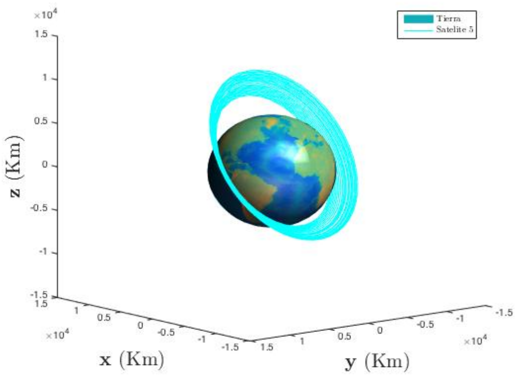

Figure 10: Satellite5 trajectory

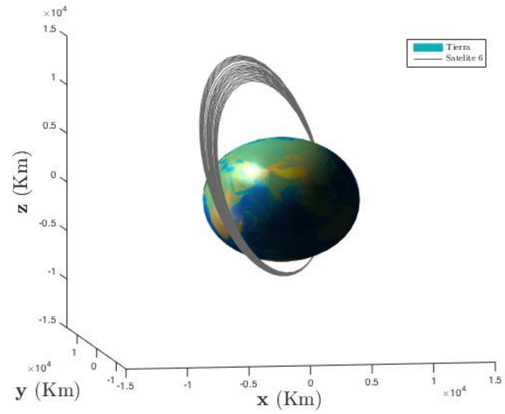

Figure 11: Satellite 6 trajectory

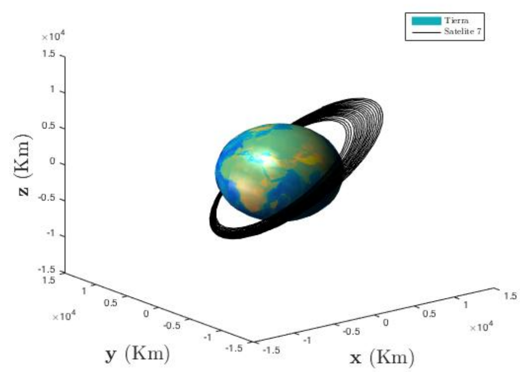

Figure 12: Satellite 7 trajectory

Finally there are the graphs of the orbits of satellites 5,6 and 7, where they are also without external forces action.

As you can see in these trajectories, all these are different, because thanks to that the satellites do not collide, as each one has an independent trajectory, then they will never collide what can be clearly seen in the simulation proposed in MATLAB.

To simulate these trajectories we use the ode45 command of MATLAB that represents a numerical method to solve ODEs, which is an approximation, therefore the trajectory of the satellite could present a small variation since this is not exact.

Below you can see the 7 orbits of the satellites, simulated by MATLAB.

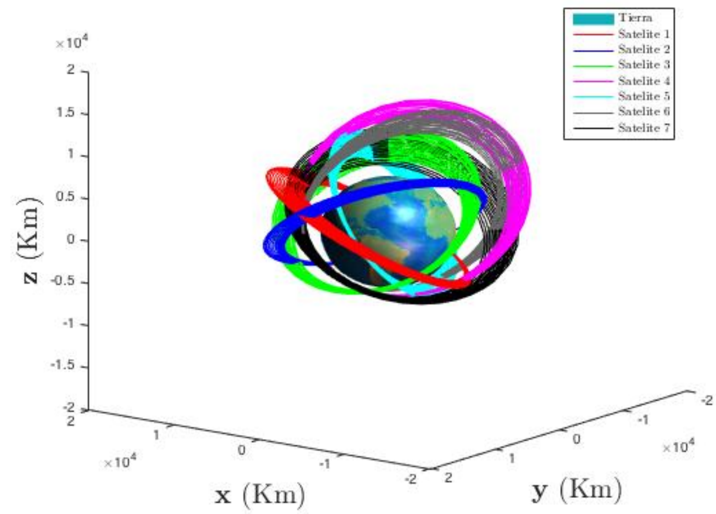

Figure 13: Trajectories of the 7 satellites

In them it is possible to be appreciated that the trajectories do not interrupt to each other, reason why the risk of collision is practically null.

3.0.2 Calculation of the terrestrial sphere and the plane of tangency with the identification of satellites on it

In this second step, after having resolved the differential equations, it is necessary to mention how the equation of the Earth and its Plane Tangent to it was obtained. Finally, the methodology will be mentioned to evaluate whether or not the satellites are on the tangent plane. This is of great importance for the military project so the explanation of this will be found later on.

It will start by defining the Earth as the total reference system of the process. For simplicity of calculations it has been decided to centre this solid surface at the origin of the system.

Considering that the radius of the planet is 6371 km, then its equation would be the following:

$$x^2 + y^2 + z^2 = (6371)^2 \tag{6}$$

Then we will present the MATLAB script that illustrates the development carried out in order to obtain graphically the Earth designed with real dimensions.

```
load('topo.mat','topo','topomap1');
[xe,ye,ze]=sphere();
r=6371;
cla reset
props.AmbientStrength = 0.1;
props.DiffuseStrength = 1;
props.SpecularColorReflectance = .5;
props.SpecularExponent = 20;
props.SpecularStrength = 1;
props.FaceColor= 'texture';
props.EdgeColor = 'none';
props.FaceLighting = 'phong';
props.Cdata = topo;
surface(r*(xe),r*(ye),r*(ze),props);
light('position',[-1 0 1]);
light('position',[-1.5 0.5 -0.5], 'color', [.6 .2 .2]);
```

The script is developed according to the following,

- In the first line of code what is done precisely is to download the package that is included in MATLAB that allows to give the physical aspect of the Earth to a sphere that must be defined with certain desired parameters (for this project are the real dimensions of the Land).
- After this, define the sphere that has coordinates xe, ye, and ze (on the x, y, z axes).
- The radius of the earth must be defined since it will be used to graph the Earth later and since it is being designed with real measurements, 6371 km.
- The following lines are typical of the package downloaded from Earth from MATLAB-

- Now, with the command *surface* what is done is plotting effectively the already designed Earth. This command allows to graph surfaces in three dimensions, so it receives as parameters the components of the sphere *xe, ye and ze*. These values must be multiplied by the radius of the Earth, in order to obtain the real dimensions of it as expected.
- Finally, the light commands are extra commands that come with the downloaded package for the design of the Earth that make the earth have shadow on one side noting that it is a volume (a body, which in this case is the Earth). which has been represented graphically.

Below are the graphs of the Earth, where you can see the measurements of the earth at its coordinates (x, y) and (z, x), being able to observe that it effectively meets the radius that is approximately 6371 km and effectively taking its sphere shape.

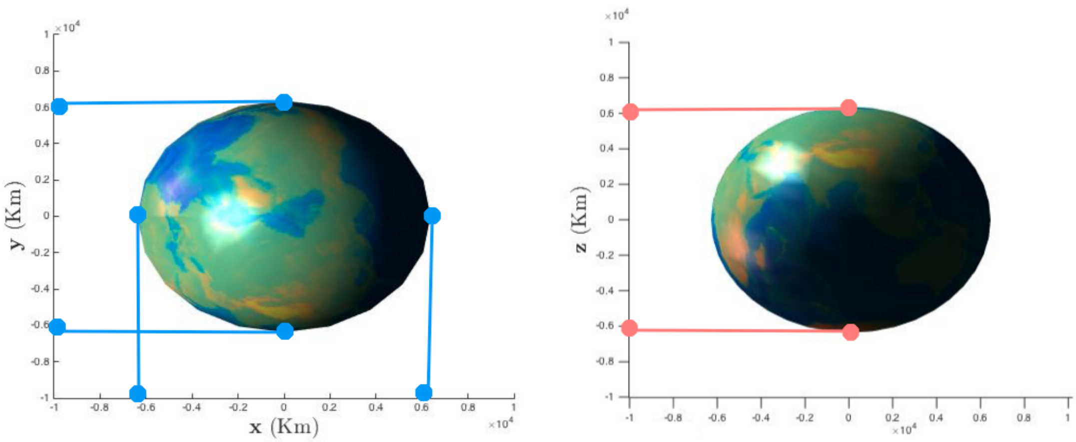

Figure 14: Dimension of the Earth.

Now, how to make a plane tangent to this sphere?

To split a point is chosen in the $\mathbb{R}^2$ domain of the spherical surface. Taking the $(-1000, 1000)$, then the point of tangency is defined as:

$$
\begin{aligned}
(-1000)^2 + (1000)^2 + z^2 &= (6371)^2 \\
z^2 &= (6371)^2 - (-1000)^2 - (1000)^2 \\
z &= \sqrt{(6371)^2 - (-1000)^2 - (1000)^2}
\end{aligned}
$$

This ends up defining that coordinate as $z = \sqrt{38589641}$.

Through this it can be proposed that then there is an ortonogal vector to the sphere at that point which is born from its center. Considering that the Earth is centered at the origin, then said orthogonal vector would be defined as follows:

$$
\begin{aligned}
\mathbf{N} &= (-1000, 1000, \sqrt{38589641}) - (0, 0, 0) \\
\mathbf{N} &= (-1000, 1000, \sqrt{38589641})
\end{aligned}
$$

If you consider that the equation of every plane is

$$Ax + By + Cz = D$$

where A, B and C are factors of the orthogonal vector, then the equation of the plane would be as follows:

$$-1000x + 1000y + \sqrt{38589641}z = D$$

And how to get D? Nothing else should be reevaluated points in the function of the plane.

The end of the tangent plane is to define the place from where the bombing will be made to the Earth, this place will be the North Pole of it, to define the plane the following MATLAB algorithm will be used.

```
syms X Y
[xp,yp]= meshgrid(-9000:1000:9000);
zp=6371^2/sqrt(38589641)-1000.*xp
./sqrt(38589641)-1000.*yp./sqrt(38589641);
hold on
mesh(xp,yp,zp);
```

This script translates into the following:

- First, the following symbolic variables X *and* Y are created. The reason for this is because MATLAB by default performs numerical calculations, so if you want to perform operations in which you have variables instead of numbers, MATLAB exclamaria that there is an error by variables not defined before. To solve this problem, what is called symbolic calculation is used where we work with symbolic expressions where their fundamental elements are constants and symbolic variables.

- Then, as you want to graph in 3D, the meshgrid function allows you to define a grid, or a data grid (which can be seen as a set of points that are part of the domain of the function) which has independent variables of xp, yp.

- Then, using the knowledge of the equation of the tangent plane (which is demonstrated in the section of Theoretical Framework of the present informative article) the equation of the tangent plane number is created.

- Once the equation of the plane has been defined, as previously mentioned, we want to work with symbolic expressions, so we use the function *inline* which, in simple terms, linearizes the equation making it depend on the symbolic variables created in the first line of code, that is, that as arguments has the symbolic variables X *and* Y.

- Finally, the plane on the Earth is plotted using the function *Mesh*

With these two algorithms it is possible to create the Earth with the Tangent Plane which is located in the north pole, where it will be the place of bombing.

Below are the graphs of the earth with the tangent plane from two different angles, in order to better see the location of this.

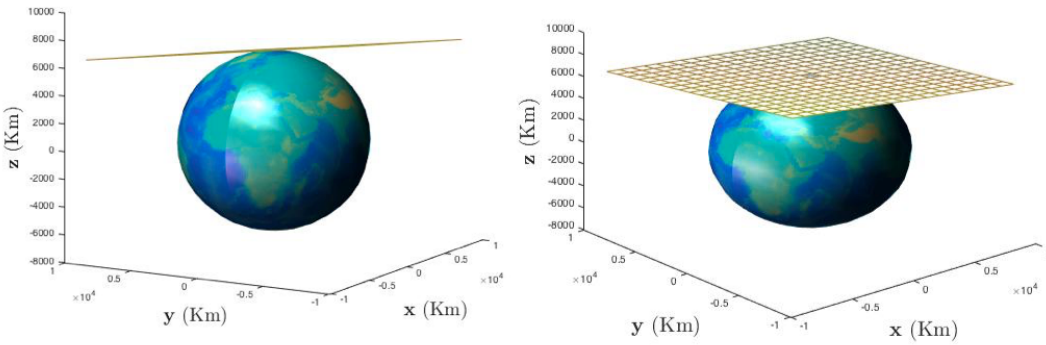

Figure 15: Earth and Tangent Plane

In these two images you can see that the Earth is with its Tangent Plane which has been located on its surface at the North Pole. The location of this plane, is to indicate the physical place where the satellites must pass to take the images. Later the real importance of the location of the tangent plane on the surface of the Earth could be noticed.

The script that calculates the plane is presented below:

```
syms X Y

ZPlano=inline('6371^2/sqrt(38589641)
-1000*X/sqrt(38589641)-1000*Y/sqrt(38589641)');

if ZPlano(r1(i,1),r1(i,3))<r1(i,5)
            ...
    else
            ...
    end
```

In the first line the symbolic variables X and Y are defined. Then, Zplano is defined as a linearized function of the equation of the tangent plane. Then the different conditions are created in which if the image of z in ZPlano evaluated in the x coordinate of the satellite (in this case satellite 1) and in the y coordinate of the same satellite, is less than the z coordinate, then over The previously defined point corresponding to the satellite is assigned certain characteristics to identify that the point is effectively fulfilling the condition. An example of this is to change the color of the modeled satellite, that is, change color when it is on the plane and return to its normal color when it is not on it in order to better visualize it and mainly, clearly identify them . Finally, the process is repeated for the rest of the satellites since they all work in the same way, only that it is evaluated for each particular satellite.

Having all these clarifications, we will proceed to the general algorithm that relates all the programs that have been working in this section.

3.0.3 The simulation

Next, the complete program of the simulation will be presented. Note that this script is the complete simulation of a single satellite by way of example, so it must be applied 7 times (it is not necessary to show all since the procedure followed is the same).

```
load('topo.mat','topo','topomap1');
[xe,ye,ze]=sphere();
r=6371;
cla reset
props.AmbientStrength = 0.1;
props.DiffuseStrength = 1;
props.SpecularColorReflectance = .5;
props.SpecularExponent = 20;
props.SpecularStrength = 1;
props.FaceColor= 'texture';
props.EdgeColor = 'none';
props.FaceLighting = 'phong';
props.Cdata = topo;
surface(r*(xe),r*(ye),r*(ze),props);
light('position',[-1 0 1]);
light('position',[-1.5 0.5 -0.5], 'color', [.6 .2 .2]);

syms X Y
[xp,yp]= meshgrid(-9000:1000:9000);
zp=6371^2/sqrt(38589641)
-1000.*xp./sqrt(38589641)-1000.*yp./sqrt(38589641);
ZPlano=inline('6371^2/sqrt(38589641)
-1000*X/sqrt(38589641)-1000*Y/sqrt(38589641)');
hold on
mesh(xp,yp,zp);

hold on

[t1, r1]=ode45('movtierrasat',[0 200], [8667.7, 2965.8, -4226.2, 4883.600, 3219.800, 3415.600]);
```

```
hLine = line('XData',r1(1), 'YData',r1(3),'ZData',r1(5),'Marker','o', 'MarkerSize',6, 'LineWidth',2);

% plot3(r1(:,1), r1(:,3), r1(:,5))

hold on

tiemposdisp1=[];

i = 1;

while true
    if ZPlano(r1(i,1),r1(i,3))<r1(i,5)
            set(hLine,'Color','r')
            tiemposdisp1(i)=t1(i);
    else
            set(hLine,'Color','b')
    end


    set(hLine, 'XData',r1(i,1), 'YData',r1(i,3),'ZData',r1(i,5))

    drawnow
    pause(0.1)
    axis([-20000 20000 -20000 20000 -20000 20000])

    i = rem(i+1,600)+1;
    if ~ishandle(hLine), break; end
end
```

Basically, what the program does is the following:

- First, there are the scripts to plot the earth along with the tangent plane seen above.
- Then, a MATLAB command called ode45 is used, which integrates each system of differential equations, in a given time and with initial conditions for each particular satellite. For this case it is being used for the first satellite.
- After each satellite is plotted as a point, with a width and size defined by the commands' MarkerSize' and 'LineWidth'.
- The *plot3* command is used to plot the trajectory of each satellite, but this
- Then an infinite loop is created, which will make the satellites orbit the earth indefinitely.
  - In the first line, a while loop counter is defined.

- Then an infinite loop is made with While True, until the graphics window closes, this allows the movement of the satellite to last what the user determines, that is, it can see the movement until it decides to close the window. interrupt the process.
- The different conditions are then created in which if the z-image of the symbolic function ZPlano evaluated in the x-coordinate of the satellite and in the y-coordinate of the same satellite, is smaller than the z coordinate of the satellite, then over the previously defined point corresponding to satellite hLine, is assigned the color red and if this condition is not fulfilled it is assigned the blue color.

  Then the process is repeated for the rest of the satellites.
- Finally these satellites are plotted, but with the previously defined conditions, in this way once a satellite passes over the tangent plane, it will turn red, thus knowing when 4 satellites are in the field of the plane and can take a photo of the section.

Below is the earth with the 7 satellites orbiting. It can be seen how four of them are on the tangent plane, having a red color instead of blue since they meet the indicated condition.

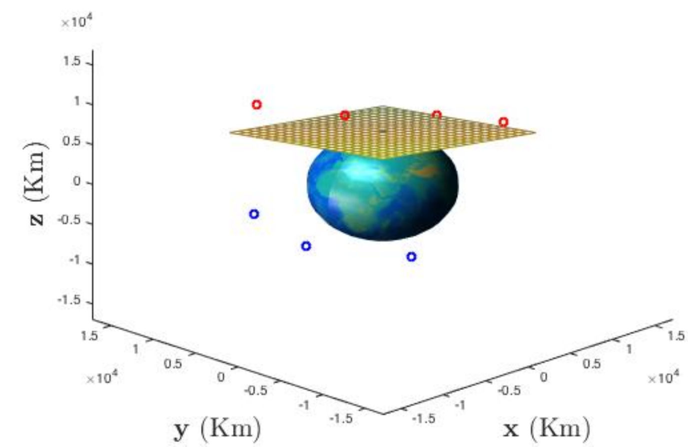

Figure 16: Orbiting satellites.

What does this image represent? Well, the Earth to which its tangent plane has been placed and four satellites positioned on it (four satellites capable of capturing the image).

3.0.4 Availability analysis

The availability analysis is the final and fundamental function of the location system that was designed and developed during the project. The algorithm used for this is:

```
syms X Y
[xp,yp]= meshgrid(-9000:1000:9000);
zp=6371^2/sqrt(38589641)-1000.*xp./sqrt(38589641)
-1000.*yp./sqrt(38589641);
ZPlano=inline('6371^2/sqrt(38589641)
-1000*X/sqrt(38589641)-1000*Y/sqrt(38589641)');

hold on
mesh(xp,yp,zp);
hold on

[t1, r1]=ode45('movtierrasat',[0 200], [8667.7, 2965.8,
-4226.2, 4883.600, 3219.800, 3415.600]);
[t2, r2]=ode45('movtierrasat',[0 200], [-8309, 2993.8,
-4231.2, -4313.600, 4592.800, -3876.600]);
[t3, r3]=ode45('movtierrasat',[0 200], [-7567.7, 2565.8,
-4351.2, -4330.900, 3921.200, 4695.600]);
[t4, r4]=ode45('movtierrasat',[0 200], [-8719.7, -3065.8,
-3032.4, -4542.900, -3921.100, 3915.600]);
[t5, r5]=ode45('movtierrasat',[0 200], [3667.7, -3265.8,
4426.2, 4483.600, 8219.800, -3415.600]);
[t6, r6]=ode45('movtierrasat',[0 200], [-4367.7, -3265.8,
-8426.2, -3483.600, -4319.800, 4415.600]);
[t7, r7]=ode45('movtierrasat',[0 200], [8598.7, 1765.8,
4310.9, -4683.600, 5519.800, 3615.600]);

hLine1 = line('XData',r2(1), 'YData',r2(3),'ZData',r2(5),'Marker','o', 'MarkerSize',6,
'LineWidth',2);
hLine = line('XData',r1(1), 'YData',r1(3),'ZData',r1(5),'Marker','o', 'MarkerSize',6,
'LineWidth',2);
hLine3 = line('XData',r3(1), 'YData',r3(3),'ZData',r3(5),'Marker','o', 'MarkerSize',6,
'LineWidth',2);
hLine4 = line('XData',r4(1), 'YData',r4(3),'ZData',r4(5),'Marker','o', 'MarkerSize',6,
'LineWidth',2);
hLine5 = line('XData',r5(1), 'YData',r5(3),'ZData',r5(5),'Marker','o', 'MarkerSize',6,
'LineWidth',2);
hLine6 = line('XData',r6(1), 'YData',r6(3),'ZData',r6(5),'Marker','o', 'MarkerSize',6,
'LineWidth',2);
hLine7 = line('XData',r7(1), 'YData',r7(3),'ZData',r7(5),'Marker','o', 'MarkerSize',6,
'LineWidth',2);

hold on
```

```
tiemposdisp1=[];
tiemposdisp2=[];
tiemposdisp3=[];
tiemposdisp4=[];
tiemposdisp5=[];
tiemposdisp6=[];
tiemposdisp7=[];

i = 1;
while true
    if ZPlano(r1(i,1),r1(i,3))<r1(i,5)
            set(hLine,'Color','r')
            tiemposdisp1(i)=1;
    else
            set(hLine,'Color','b')
            tiemposdisp1(i)=0;
    end

    if ZPlano(r2(i,1),r2(i,3))<r2(i,5)
        set(hLine1,'Color','r')
        tiemposdisp2(i)=2;
    else
        set(hLine1,'Color','b')
        tiemposdisp2(i)=0;
    end

    if ZPlano(r3(i,1),r3(i,3))<r3(i,5)
        set(hLine3,'Color','r')
        tiemposdisp3(i)=3;
    else
        set(hLine3,'Color','b')
        tiemposdisp3(i)=0;
    end

    if ZPlano(r4(i,1),r4(i,3))<r4(i,5)
        set(hLine4,'Color','r')
        tiemposdisp4(i)=4;
    else
        set(hLine4,'Color','b')
        tiemposdisp4(i)=0;
    end

    if ZPlano(r5(i,1),r5(i,3))<r5(i,5)
        set(hLine5,'Color','r')
        tiemposdisp5(i)=5;
    else
        set(hLine5,'Color','b')
        tiemposdisp5(i)=0;
    end

    if ZPlano(r6(i,1),r6(i,3))<r6(i,5)
        set(hLine6,'Color','r')
        tiemposdisp6(i)=6;
    else
        set(hLine6,'Color','b')
        tiemposdisp6(i)=0;
    end
```

```
    if ZPlano(r7(i,1),r7(i,3))<r7(i,5)
        set(hLine7,'Color','r')
        tiemposdisp7(i)=7;
    else
        set(hLine7,'Color','b')
        tiemposdisp7(i)=0;
    end


    set(hLine1, 'XData',r2(i,1), 'YData',r2(i,3),'ZData',r2(i,5))
    set(hLine, 'XData',r1(i,1), 'YData',r1(i,3),'ZData',r1(i,5))
    set(hLine3, 'XData',r3(i,1), 'YData',r3(i,3),'ZData',r3(i,5))
    set(hLine4, 'XData',r4(i,1), 'YData',r4(i,3),'ZData',r4(i,5))
    set(hLine5, 'XData',r5(i,1), 'YData',r5(i,3),'ZData',r5(i,5))
    set(hLine6, 'XData',r6(i,1), 'YData',r6(i,3),'ZData',r6(i,5))
    set(hLine7, 'XData',r7(i,1), 'YData',r7(i,3),'ZData',r7(i,5))
    drawnow
    pause(0.1)
    axis([-20000 20000 -20000 20000 -20000 20000])

    i = rem(i+1,600);
    if ~ishandle(hLine), break; end
end

for j=1:i-1
    if tiemposdisp1(j)==1
        plot(j,tiemposdisp1,'Color','k','Marker','.');
        hold on
    end
    if tiemposdisp2(j)==2
        plot(j,tiemposdisp2,'Color','k','Marker','.');
        hold on
    end

    if tiemposdisp3(j)==3
        plot(j,tiemposdisp3,'Color','k','Marker','.');
        hold on
    end
    if tiemposdisp4(j)==4
        plot(j,tiemposdisp4,'Color','k','Marker','.');
        hold on
    end
    if tiemposdisp5(j)==5
        plot(j,tiemposdisp5,'Color','k','Marker','.');
        hold on
    end
    if tiemposdisp6(j)==6
        plot(j,tiemposdisp6,'Color','k','Marker','.');
        hold on
    end
    if tiemposdisp7(j)==7
        plot(j,tiemposdisp7,'Color','k','Marker','.');
        hold on
    end

    eje_x=xlabel('\textbf{Iteraciones}') ;
```

```
    set(eje_x,'Interpreter','latex', 'fontsize', 20);
    eje_y=ylabel('\textbf{Disponibilidad}') ;
    set(eje_y,'Interpreter','latex', 'fontsize', 20);
end
```

Where:

- First the symbolic variables X *and* Y are defined. The reason for this is because MATLAB by default makes numerical calculations, so if you want to perform operations in which you have variables instead of having numbers, MATLAB could not perform the action of the operation. This is why symbolic calculation is used as a key tool for this, where we work with symbolic expressions.

- After this, as you want to graph in 3D, the meshgrid command allows you to define a grid, or a data grid (which can be seen as a set of points that are part of the domain of the function) which has independent variables x_p e y_p

- After this, *ode45* is used, which integrates each system of differential equations, in a determined time and with initial conditions for each particular satellite.

- After this, each satellite is plotted in the form of a point, with a width and size which is defined by the commands *MarkerSize* and *LineWidth*

- Subsequently empty lists are created called *timesdispn* where n is from 1 to 7 for all satellites. These lists should store data later.

- The variable i is assigned the value of 1 which acts as a kind of counter and creates an infinite loop that makes the satellites orbit indefinitely.

- Once the satellites are orbiting around the Earth the different conditions are created in which if the z-image of the symbolic function of the ZPlan evaluated in the coordinate x of the satellite and in the coordinate y of the same satellite is less than the coordinate z, then on the previously defined point corresponding to the satellite hLine, it is assigned the red color and if this condition is not met, it is assigned the blue color, in addition to this when updating to red color also to the empty list is added the value of 1 (since it is the first satellite in orbit) and if it is blue it remains with value 0.

The procedure is repeated for the rest of the satellites, but it goes with 1,2,3 and up to 4 units of time. This means that the time available is stored but presenting the defect considering that every 1 unit of time enters a satellite orbit.

- Finally, the available time of these satellites is plotted

It is important to mention that in the graph, the x axis represents a number of iterations, which corresponds to a time for each satellite. The reason for doing this is because each satellite is carried by a lag of 1,2,3 and up to 4 units of time.

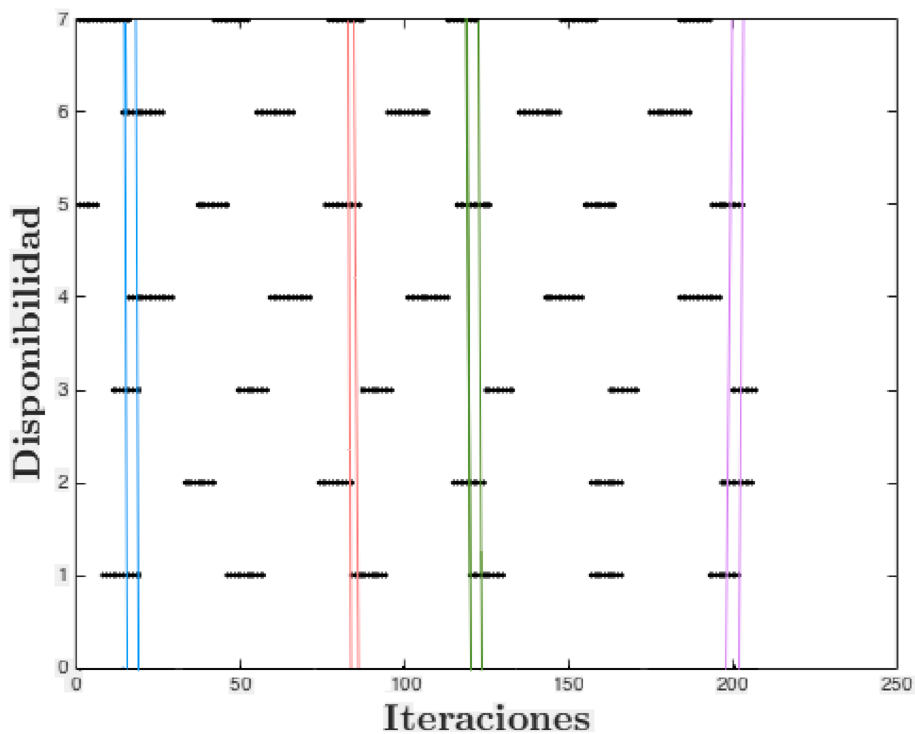

Figure 17: Availability of Satellites.

The unit of time that has been used is in days. In the graph you can see the time that the intersected satellites are available, that is, the four that are on the plane that allow predicting how much time you have so that you can capture the precise image.

4 Conclusions

Based on the results obtained, it can be affirmed that the proposed mathematical model fulfills the objective of providing for a specific time the positions and speeds of the seven satellites. MATLAB was the primary tool and easy to handle in the manipulation of the data, which is beneficial because of not having the ability to process the calculations implied by the model, it would have been necessary to resort to other programming languages less adapted for this type of problems.

It could be obtained that as the number of satellites in orbit increases, the frequency with which four of them are in the exact position to obtain a photograph of a specific place is greater. This brings with it an increased risk of collision, as it increases the density of objects in space.

Despite the success of the results, the model does not adjust in its entirety to what happens in reality since there are external factors that can modify the state of movement of the satellites which are not being considered. For example, solar storms and the gravitational-electromagnetic field generated by the presence of the Moon. It is proposed for subsequent works related to the theme to adjust this mathematical model considering these factors, obtaining a more realistic and accurate description of what really happens. To know the position of the satellite with respect to the Earth, it is necessary to know the eccentricity, the semi-major axis, the perigee time, the perigee angle and the inclination. These measures will deliver results with a much lower error range, so they also intend to introduce in a future work.

5 References

[1] José Manuel Huidobro, 2002, Historia de los satélites de comunicaciones.

[2] Ramón Martínez Rodríguez-Osorio, 2010, Tipos de órbitas. Constelaciones de satélites.

[3] R. Larson & B. H. Edwards, (2010), *Calculo 2: De varias variables, Novena Edicion.* Mexico: Mc Graw Hill.

[4] J. Stewart, (2008), *Calculo: De varias variables Trascendentes tempranas, Sexta Edicion*. CENGAGE Learning.

[5] S. L. Grossman & J. J. Flores Godoy, (2012), *Algebra Lineal, Septima Edicion*. Mexico: Mc Graw Hill.

[6] NATIONAL GEOGRAPHIC. TORMENTAS SOLARES. Available in: http://www.nationalgeo

[7] Universidad de las Americas de Puebla. Mexico. Satelites. Available in: http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/peredo_a_s/capitulo2.pdf

[8] E. Velasco, *MECANICA LAGRANGIANA Y HAMILTONIANA*. España: Universidad de Madrid.

[9] Gonzalo Melchisedec Palomera, (2014), *DINAMICA EN RELATIVIDAD GENERAL: ESTRUCTURA HAMILTONIANA Y MOVIMIENTO SATELITAL*. Chile: Universidad de Santiago de Chile, Facultad de Ciencias, Departamento de Matematica y Cs. de la Computacion.

[10] Jean-Peter Ylén, Simulation of Spacecraft Attitude and Orbit Dynamics. Helsinki University of Technology.

[11] Muhammad Shamsul Kamal Adnan y Radzuan Razali, 2011, Mathematical Modeling and Software Development On Pertubation Effects Towards Small Satellite Mission On Low Earth Orbit. International Journal of Applied Science and Technology.

[12] ADRIÁN GARCÍA-HIDALGO BAEZA, 2013, DISEÑO DEL SISTEMA DE POTENCIA PARA UN SATÉLITE DE ÓRBITA BAJA (LEO). UNIVERSIDAD POLITÉCNICA DE MADRID