Aristotele University of Thessaloniki
MSc Computational Physics
Master Thesis

Orbital dynamics around a rotating and precessing oblate spheroid

Liagka Aggeliki
Supervisor: Voyatzis George
October,2022

Abstract

The aim of this thesis is to study the orbital dynamics in the gravitational field of a precessing oblate spheroid. We consider the oblate spheroid body to be an asteroid and we also consider a body of negligible mass, let's assume a spacecraft of a mass much smaller than the asteroid's, orbiting the precessing asteroid. For several sets of initial conditions, we integrate the equations of motion in the body-fixed (rotating) frame of reference. To compute the orbital elements for each orbit, we use a transformation (rotating matrix in quaternion form) from the rotating to the inertial frame. Then, using the maximum values of the semi-major axis, the eccentricity and the inclination (a_{max}, e_{max} and i_{max}) we create dynamical maps, that show the stable and unstable areas, regarding the initial conditions. After that, we study individual orbits from both areas. In that way, one can see, how the shape of the asteroid, the initial distance of the spacecraft and some other parameters could affect the orbital stability around the precessing asteroid.

Περίληψη

Στόχος αυτής της διπλωματικής εργασίας είναι να ερευνήσει την τροχιακή δυναμική στο βαρυτικό πεδίο ενός πεπλατυσμένου σφαιροειδούς με περιστροφή και μετάπτωση. Θεωρούμε το πεπλατυσμένο σφαιροειδές σώμα ως έναν αστεροειδή και θεωρούμε επίσης, ένα σώμα αμελητέας μάζας, ας υποθέσουμε ένα διαστημόπλοιο πολύ μικρότερης μάζας από του αστεροειδή, να περιφέρεται γύρω από αυτόν. Για αρκετά σύνολα αρχικών συνθηκών, ολοκληρώνουμε τις εξισώσεις της κίνησης στο περιστρεφόμενο σύστημα αναφοράς. Για να υπολογίσουμε τα τροχιακά στοιχεία κάθε τροχιάς, χρησιμοποιούμε έναν μετασχηματισμό (πίνακα στροφής με τη χρήση *quaternions*) από το περιστρεφόμενο στο αδρανειακό σύστημα. Έπειτα, χρησιμοποιώντας τις μέγιστες τιμές του μεγάλου ημιάξονα, της εκκεντρότητας και της κλίσης (a_{max}, e_{max} και i_{max}), δημιουργούμε δυναμικούς χάρτες, οι οποίοι δείχνουν τις ευσταθείς και ασταθείς περιοχές, όσων αφορά τις αρχικές συνθήκες. Στη συνέχεια, μελετούμε τροχιές και από τις δύο περιοχές. Κάποιος μπορεί να δει, με αυτόν τον τρόπο, πώς το σχήμα του αστεροειδή, η αρχική απόσταση του διαστημοπλοίου και κάποιες άλλες παράμετροι, επηρεάζουν την τροχιακή ευστάθεια, γύρω από τον περιστρεφόμενο με μετάπτωση αστεροειδή.

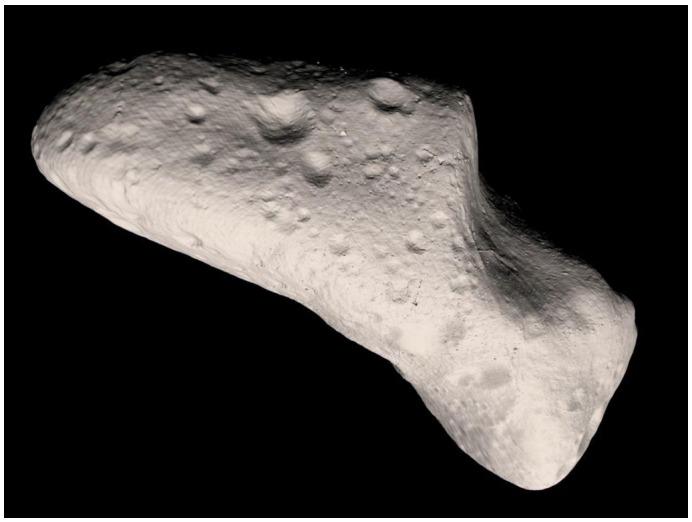
Contents

List of Figures

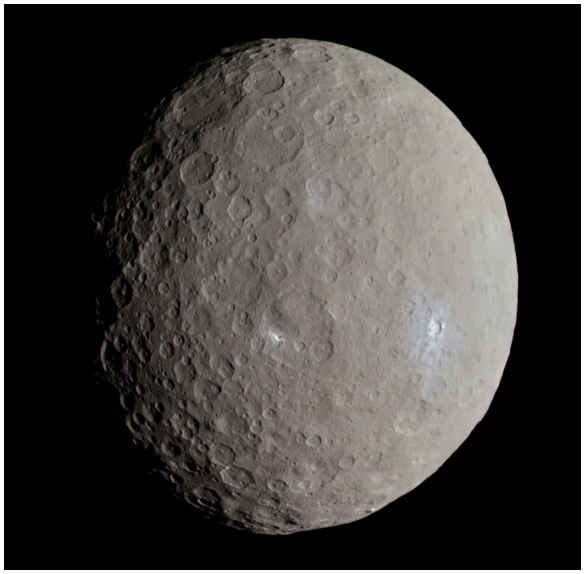
1 Introduction

1.1 Asteroids

Asteroids, also called minor planets, are small bodies of the inner Solar System; rocky or metallic, with no atmosphere. Their shapes and sizes vary, ranging from 1-meter rocks to a dwarf planet almost 1000 km in diameter. Some examples can be seen below. A body smaller than an asteroid is called a meteoroid. Asteroids are probably remnants left over from the early formation of our Solar System a few billion years ago. [1]



(a) Eros 433



(b) Ceres (dwarf planet)

(c) Vesta

Figure 1: Asteroid examples

According to *NASA* the current known asteroid count is: $1,113,527$[3]. The majority

of that number is orbiting our Sun between Mars and Jupiter within the main asteroid belt. An other region surrounding our Solar System, farther Neptune's orbit, where many asteroids can be found, is the Kuiper belt. Asteroids found orbiting Sun in the Main belt are mainly composed of silicon rocks and metals. An exception to this is Ceres because a big part of it is iced water. On the other hand, the asteroids of the Kuiper belt are mainly composed of ice.

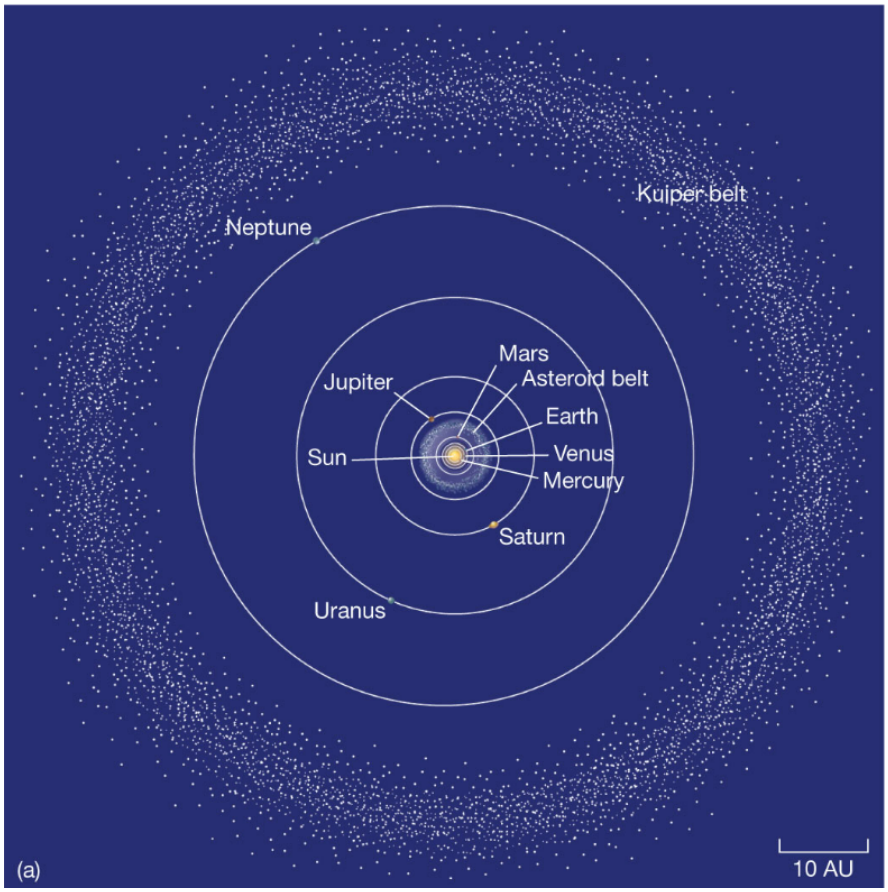


Figure 2: Main Asteroid and Kuiper belt.

As far as asteroid classification goes there are two more worth mentioning categories; Trojans and Near-Earth Asteroids. Trojans are asteroids that share their orbit with a larger planet, but do not collide with it because they gather around two special places in the orbit (called the L4 and L5 Lagrangian points). The most significant are the Jupiter Trojans; an extended population of asteroids sharing their orbit with Jupiter (seen in the figure below). Finally, Near-Earth Asteroids are objects with orbits passing close to the Earth's orbit. [3]

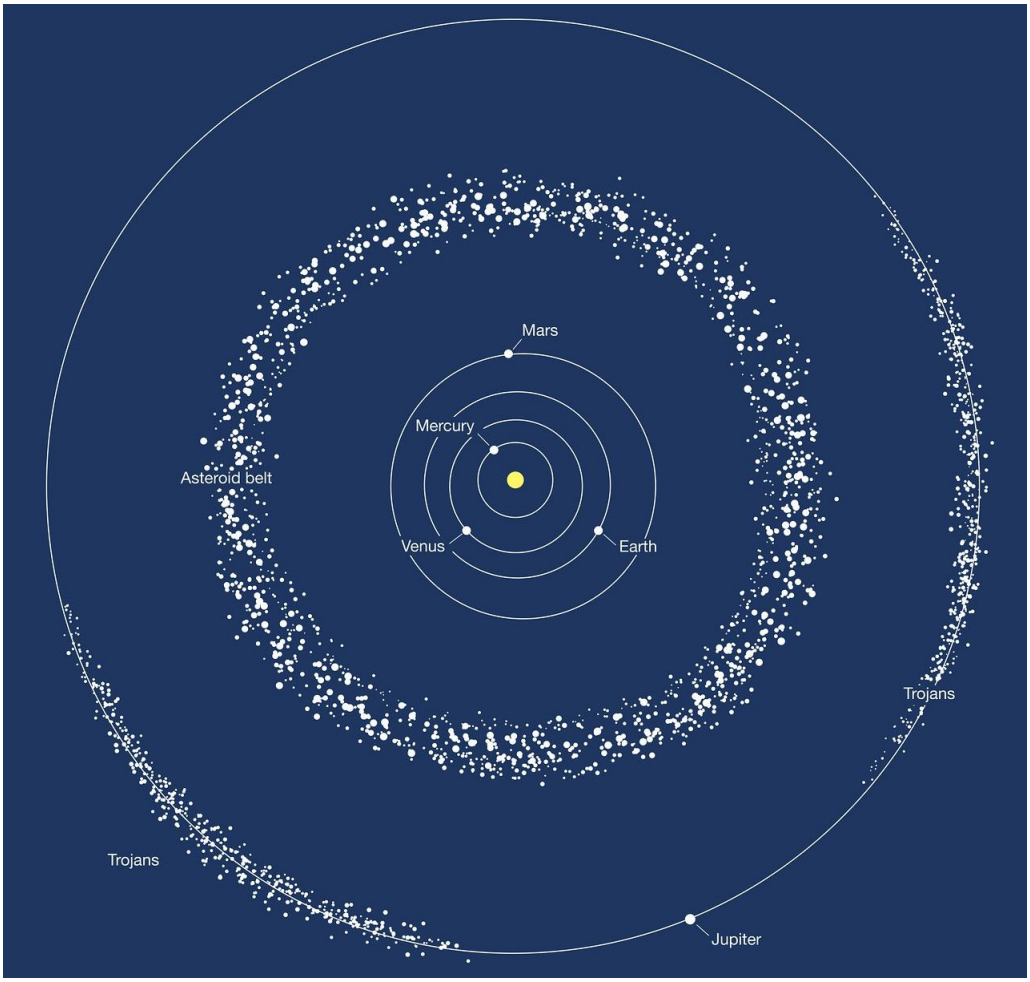


Figure 3: Jupiter Trojans.

As we mentioned, the size and shape of the asteroids varies. Most of them have an irregular shape, but a few of the biggest are nearly spherical due to the influence of their self gravity. As they revolve around the Sun in (weakly) elliptical orbits, the asteroids also rotate. On the time-scale of millions of years the rotation speed of asteroids change due to the momentum of photons (from the Sun) absorbed, reflected and emitted (Yarkovsky effect) from the surface of asteroids, and due to the impact of meteorites. It is known that most of the asteroids rotate with a period longer than 2.3 hours which can be understood as a result of the influence of the centrifugal force on a collection of solid fragments bound together mainly by the gravity force. However, many small asteroids rotate much faster because of different consistency. [4]

1.2 Asteroid Space Missions

The orbital dynamics of a particle in the gravity of an asteroid is an important issue for space missions, but a quite complex one as well, due to the irregular shape many of the asteroids have. The irregular shape forms an irregular gravitational field close to its

surface. This gravitational field in combination with other external perturbations such as solar gravity and solar radiation pressure, is what makes this problem more difficult than a simple description of the two-body problem.[5]

The study of the motion of such particle is used in choosing the right orbits for close proximity operations, landing paths, parking orbits etc. On a further note, there are space missions, such as landings, that help us discover more about the early stage of our Solar System, but there are also missions useful for planetary defence.

This research topic applies to many successful missions. It is worth mentioning the Near Earth Asteroid Rendezvous (NEAR) whose landing on Eros 433 marked the first time a U.S. spacecraft landed on an asteroid (2000) and returned valuable data. NEAR didn't survive the extreme cold and the contact was lost on February 2001. The Dawn mission successfully visited the two largest objects in the main asteroid belt, Vesta (2011) and Ceres (2015). The Dawn spacecraft is currently in a stable orbit around Ceres.[6]

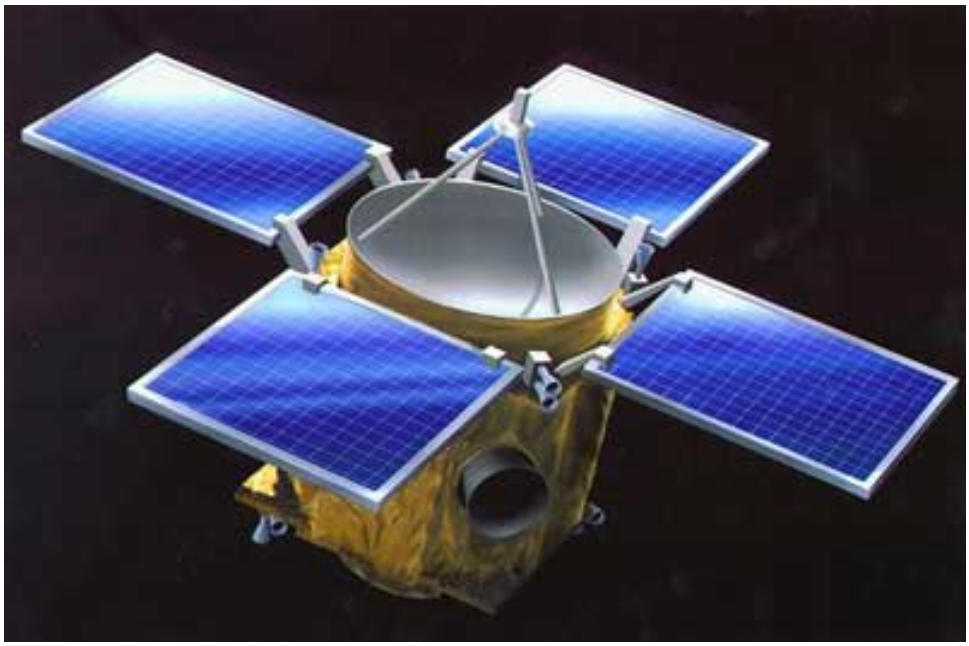


Figure 4: NASA's NEAR

Furthermore, JAXA also designed its own remarkable space missions. Two of them are Hayabusa (2003) and Hayabusa 2 (2018) that landed Itokawa and Ryugu, respectively. Hayabusa was the first spacecraft returning to Earth having brought asteroid samples and the second one in history descended to the surface of an asteroid. [7]

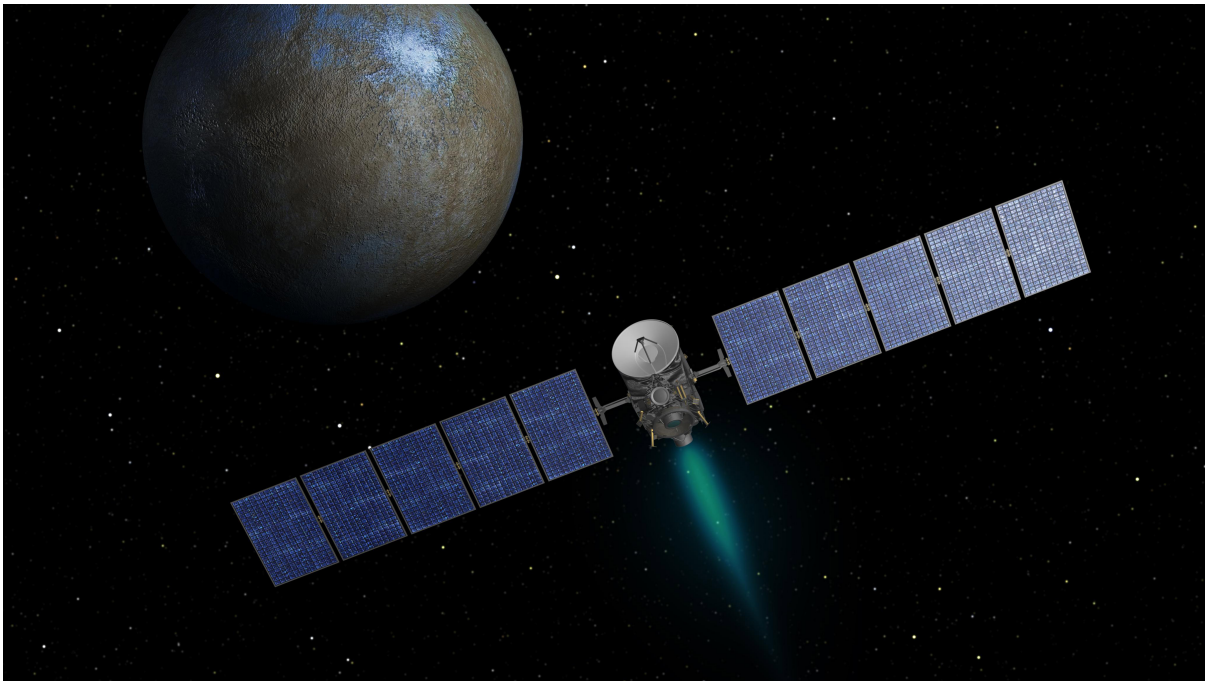
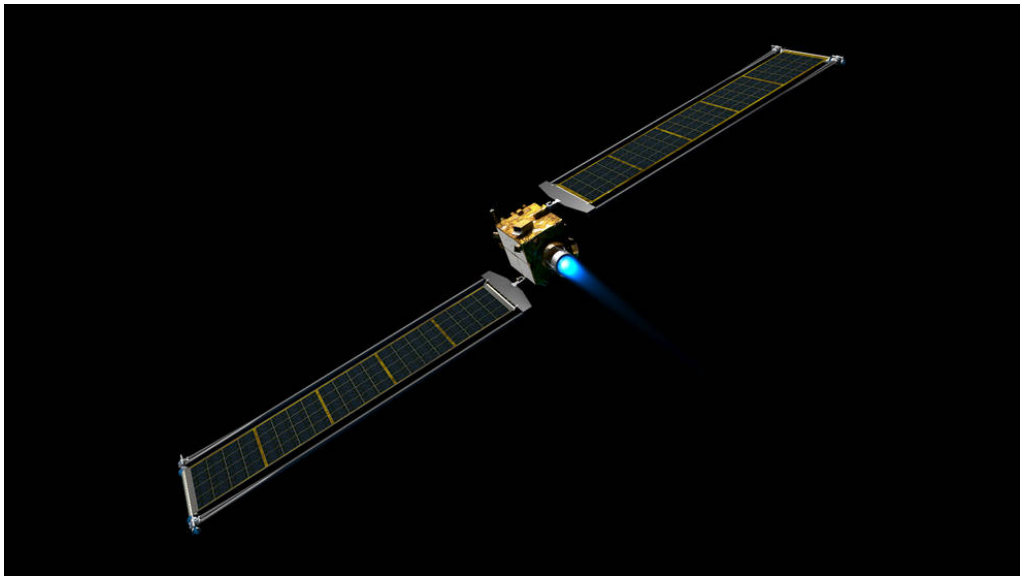


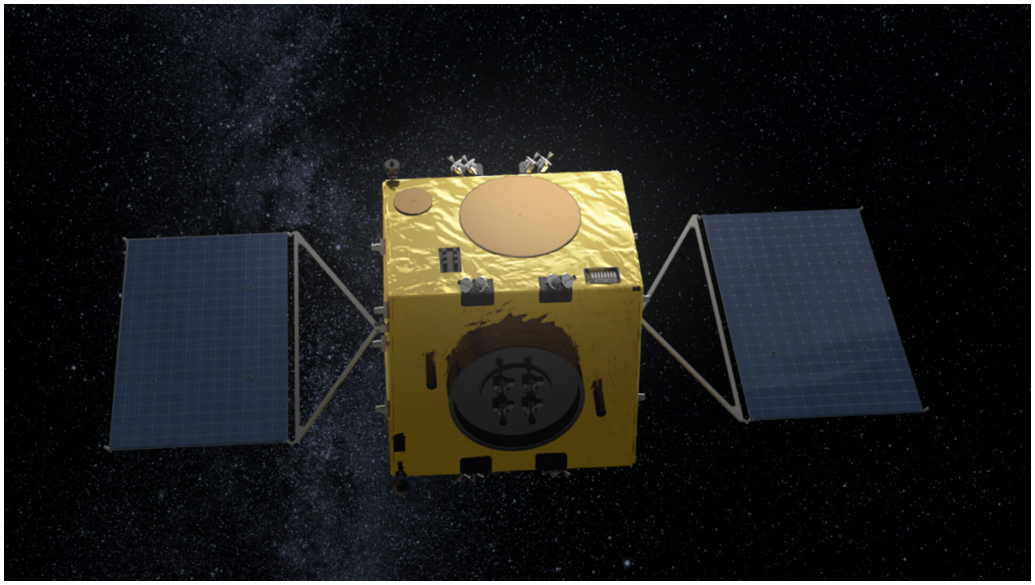
Figure 5: NASA's Dawn

Figure 6: JAXA's Hayabusa

Let us end the conversation about asteroid missions here by mentioning a big planetary defense mission NASA designed and launched in November 2021 (DART), with ESA's Hera support that is scheduled to evaluate the effects of DART. DART is the first-ever mission aiming at demonstrating deflection of an asteroid by changing its motion in space through kinetic impact. This spacecraft will strike Dimorphos, moon of the asteroid Didymos, and scientists will study the change of the kinetic dynamics after the impact.[6]



(a) NASA's DART



(b) ESA's Hera

Figure 7: Planetary defense missions

1.3 Asteroid precession and nutation

As mentioned before, asteroids orbiting our Sun also rotate. There are already several well-established theories to model the rotational motion of a rigid body with a quite simple mathematical approach, but still there is a lot of work to be done involving the rotational properties of asteroids. An interesting study has been published in 2014 aiming to a high precision model of precession and nutation of Ceres, Vesta, Eros, Steins and Itokawa. The study is concentrated in particular on the motion of their spin axis in space [8]. The rotation of a celestial body in the size and shape of an asteroid can also be chaotic. An example of that, is Hyperion, a moon of Saturn. Hyperion is characterised by its irregular shape, its chaotic rotation and its unexplained sponge-like appearance. Chaotic rotation have been investigated by M.Tarnopolski [9], who used a chaos control method to suppress chaos and even turn chaotic motion into periodic. Numerical examples were presented in his work with parameters suitable of Hyperion, considering the shape of the satellite to be a triaxial ellipsoid.

To understand in more detail the rotating motion of an asteroid let us explain the terms of precession and nutation. When the orientation of the rotational axis of a rotating body changes, then the body precesses. In an appropriate reference frame it can be defined as a change in the first Euler angle (described in the next chapter). In other words, one can simply say that the body precesses if its rotational axis also rotates about a second axis. Nutation is a nodding motion in the axis of rotation and in astronomy it is caused by the gravitational forces of other nearby bodies acting upon the spinning object. In an appropriate reference frame it can be defined as a change in the second Euler angle (also described in the next chapter). Although precession and nutation are defined as a change of orientation of the rotation axis and they are caused by the same effect (gravity), they separate so that precession being a steady long-term change in the axis of rotation, and nutation being the combined effect of similar shorter-term variations. In physics, there are two types of precession: torque-free and torque-induced.

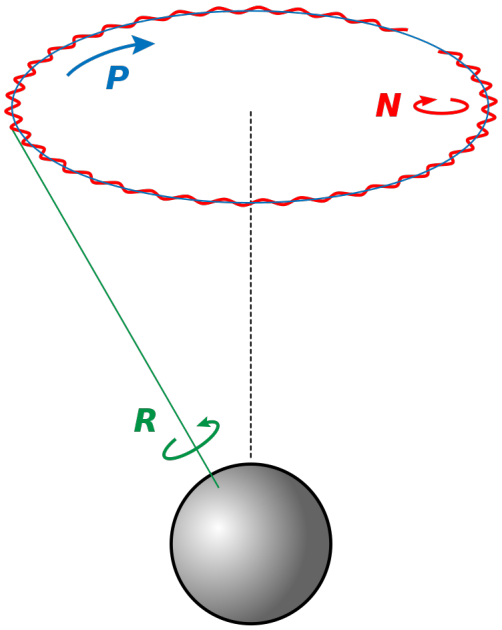


Figure 8: Rotation *(Green)*, Precession *(Blue)* and Nutation *(Red)*.

1.4 Further general characteristics

As we previously noted, asteroids vary greatly in size. In the diagram below, one can find the asteroids of our Solar System categorized by size and number. Although the three largest are very much like miniature planets, it is clear that the vast majority are much smaller and are irregularly shaped.

The four largest objects in the asteroid belt are Ceres, Vesta, Pallas and Hygiea. These four asteroids constitute half the mass of the asteroid belt. As seen in the table below [1], their orbital radius is similar, but we can detect differences in inclination and eccentricity. Here, we can also see their orbital and rotation periods. It takes from five to almost fourteen hours for one of these asteroids to complete a full rotation around themselves, and from 3 to five years to complete an orbit around the Sun. Ceres and Pallas have the same orbital radius and orbital period.

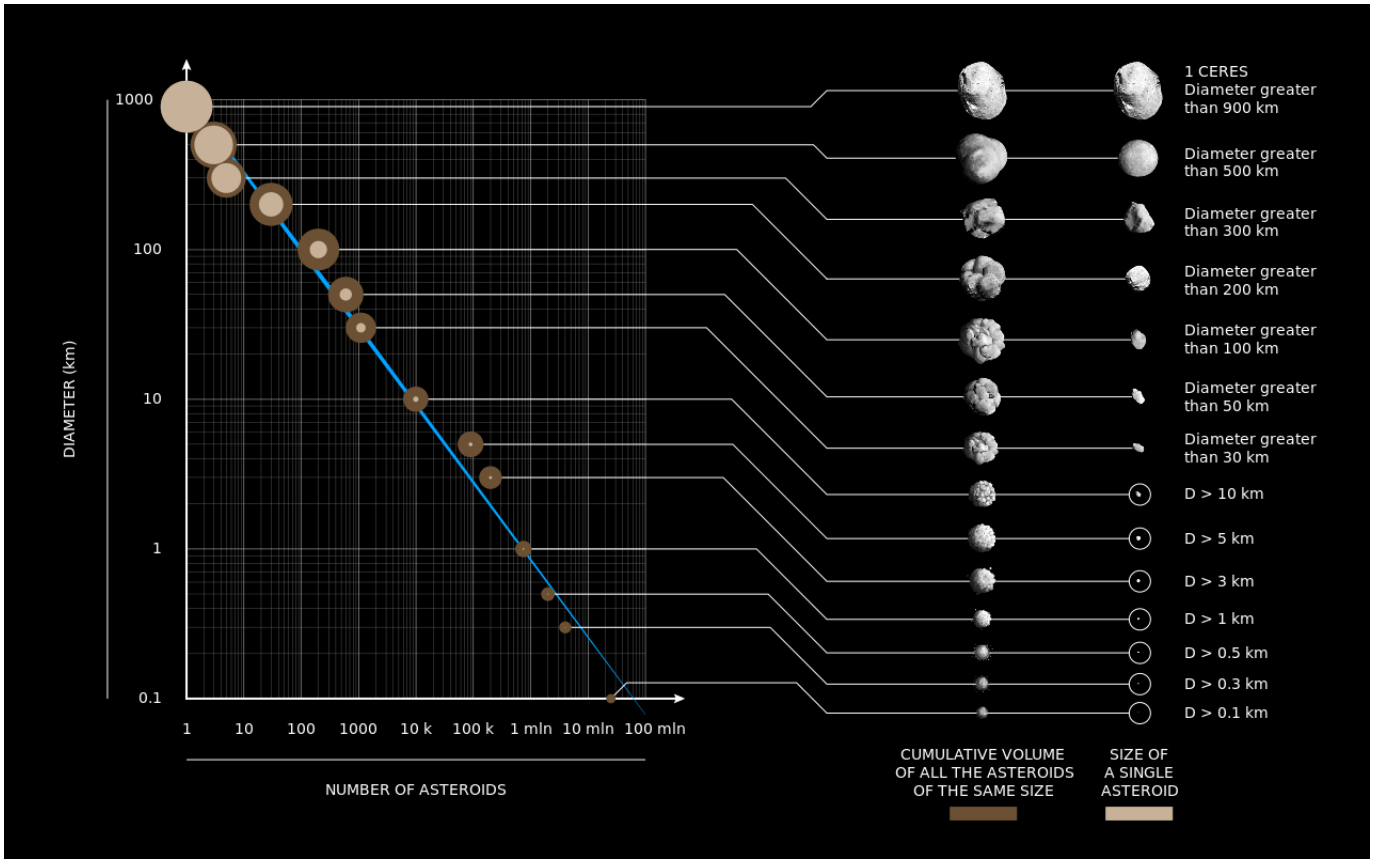


Figure 9: Asteroids by size and number[1].

Attributes of largest asteroids

Name	Orbital radius (AU)	Orbital period (years)	Inclination to ecliptic	Orbital eccentricity	Diameter (km)	Diameter (% of Moon)	Mass ($\times 10^{18}$ kg)	Mass (% of Ceres)	Density (g/cm^3)	Rotation period (hr)
Ceres	2.77	4.60	10.6°	0.079	964×964×892 (mean 939.4)	27%	938	100%	2.16±0.01	9.07
Vesta	2.36	3.63	7.1°	0.089	573×557×446 (mean 525.4)	15%	259	28%	3.46 ± 0.04	5.34
Pallas	2.77	4.62	34.8°	0.231	550×516×476 (mean 511±4)	15%	204±3	21%	2.92±0.08	7.81
Hygiea	3.14	5.56	3.8°	0.117	450×430×424 (mean 433±8)	12%	87±7	9%	2.06±0.20	13.8

Figure 10: The four largest asteroids[1].

2 Rigid Body Rotation

In this chapter we will examine the rotation parameters of a rigid body in three dimensions. A rigid body is considered to be a collection of small mass elements, whose connection imposes that the distance between any two given points on the rigid body remains constant in time, regardless of external forces or moments exerted on it. Hence, there is no deformation on a rigid body.

2.1 Euler angles

According to Euler's rotation theorem, any rotation may be described using three angles. The three angles that Euler introduced, describe the orientation of a rigid body with respect to the fixed coordinate frame. The common notation used for these angles is (ϕ, θ, ψ).

Let's now look in more detail what each of these angles represent. We start in the inertial frame with coordinates X, Y, Z and we gradually take three rotations. As seen in the figure below, ϕ represents the angle of a counterclockwise rotation about the Z axis, leading to a new frame with coordinates X_2, Y_2, Z_2, θ represents the angle of a counterclockwise rotation around the X_2 axis, leading to a new frame with coordinates X_3, Y_3, Z_3 and, finally, ψ represents the angle of an also counterclockwise rotation around the Z_3 axis and the new frame is the body fixed frame with coordinates x, y, z. [10]

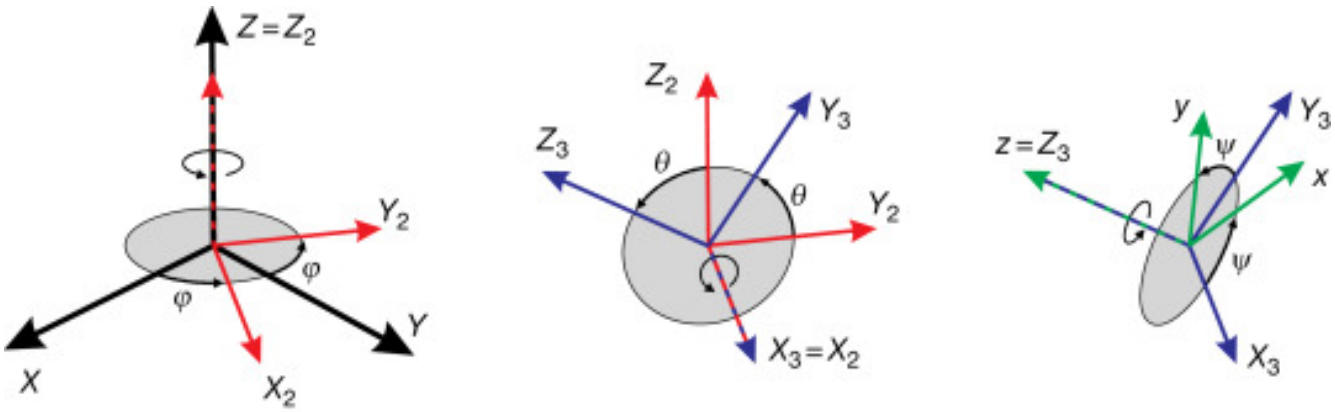


Figure 11: Euler angles.

If the rotations are written in terms of rotation matrices, we respectfully get [11]

$$D_1 = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

$$D_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \tag{2.2}$$

and

$$D_3 = -\begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

The results from the multiplication of the three matrices, beginning from the last one, so $D = D_3 D_2 D_1$, leads us to the 3×3 rotation matrix

$$D = \begin{bmatrix} \cos\psi\cos\phi - \cos\theta\sin\phi\sin\psi & \cos\psi\sin\phi + \cos\theta\cos\phi\sin\psi & \sin\psi\sin\theta \\ -\sin\psi\cos\phi - \cos\theta\sin\phi\cos\psi & -\sin\psi\sin\phi + \cos\theta\cos\phi\cos\psi & \cos\psi\cos\theta \\ \sin\theta\sin\phi & -\sin\theta\cos\phi & \cos\theta \end{bmatrix} \tag{2.4}$$

Using this matrix, the final transformation between the inertial and the rotating frame occurs

$$x = DX \tag{2.5}$$

where x and X are column vectors containing the three position components of the rotating and the inertial frame, respectfully. It is clear that to find the inverse transformation, we need only the inverse D^{-1} matrix, so that

$$X = D^{-1}x. \tag{2.6}$$

On a final note, we can easily derive the above two equations to compute the corresponding velocities. Hence, for the inertial to rotating frame velocity

$$V = \dot{X} = D^{-1}\dot{x} + \dot{D}^{-1}x \tag{2.7}$$

and for the other way around, we have

$$v = \dot{x} = D\dot{X} + \dot{D}X. \tag{2.8}$$

2.2 Instantaneous angular velocity and Euler angles

We consider a rigid body rotating around one of its points, O, that is fixed. Generally, the direction of the rotation axis, that passes through the point O, changes over time. It is clear that the motion of the rigid body would be fully known, if the angular velocity vector is also known as a function of time.

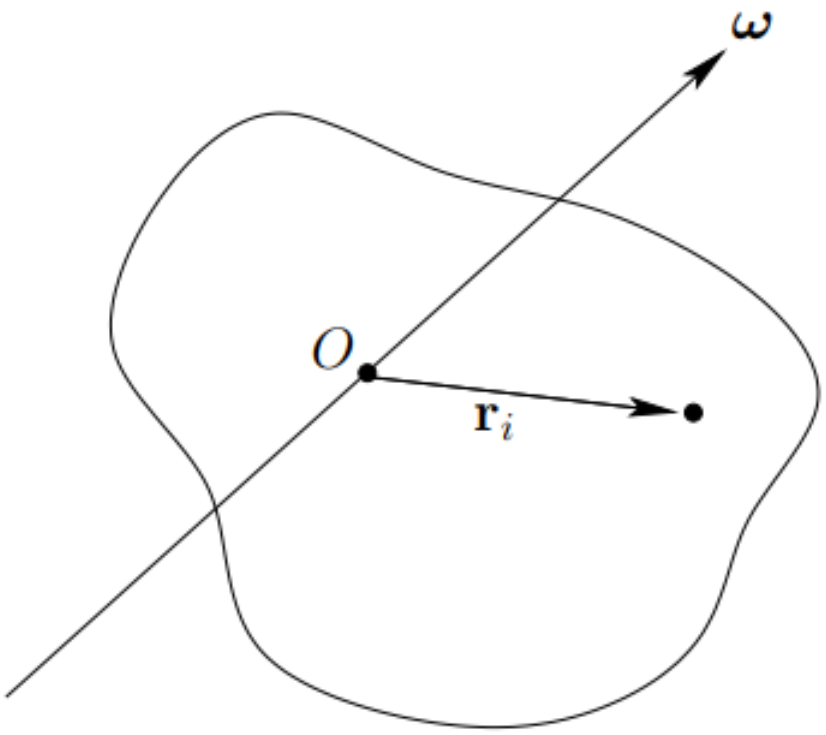


Figure 12: A rigid rotating body.
[10]

Let us now express the angular velocity vector in terms of the Eulerian angles. The ω component for the ϕ angle has a magnitude of $\dot{\phi}$ and a direction along the Z axis. If we consider the unit vector along this axis as $\mathbf{e}_Z$, we can write

$$\boldsymbol{\omega}_\phi = \dot{\phi}\mathbf{e}_Z. \tag{2.9}$$

The component associated with θ has a magnitude $\dot{\theta}$ and is directed along the X_2 axis (or X_3 axis, since they coincide). Hence, we can write

$$\boldsymbol{\omega}_\theta = \dot{\theta}\mathbf{e}_{X_2}. \tag{2.10}$$

Likewise, the component associated with ψ has a magnitude $\dot{\psi}$ and is directed along the Z_3 axis. Hence, we can write

$$\boldsymbol{\omega}_\psi = \dot{\psi}\mathbf{e}_{Z_3}. \tag{2.11}$$

Note also that $\mathbf{e}_{Z_3} = \mathbf{e}_z$. Finally, we want to write the total angular velocity

$$\boldsymbol{\omega} = \boldsymbol{\omega}_\phi + \boldsymbol{\omega}_\theta + \boldsymbol{\omega}_\psi. \tag{2.12}$$

To do that properly, we will need some of the transformation equations between the inertial frame and body fixed frame. It can be easily verified that

$$\begin{aligned} \mathbf{e}_Z &= \sin\psi\sin\theta\mathbf{e}_x + \cos\psi\sin\theta\mathbf{e}_y + \cos\theta\mathbf{e}_z \\ \mathbf{e}_{X_2} &= \cos\psi\mathbf{e}_x - \sin\psi\mathbf{e}_y. \end{aligned} \tag{2.13}$$

Using these relations of (2.13) in (2.9) and (2.10), we get for every angular velocity component in the body fixed frame [10]

$$\begin{aligned} \boldsymbol{\omega}_x &= (\sin\psi \sin\theta\, \dot{\phi} + \cos\psi\, \dot{\theta})\mathbf{e}_x \\ \boldsymbol{\omega}_y &= (\cos\psi \sin\theta\, \dot{\phi} - \sin\psi\, \dot{\theta})\mathbf{e}_y \\ \boldsymbol{\omega}_z &= (\cos\theta\, \dot{\phi} + \dot{\psi})\mathbf{e}_z. \end{aligned} \tag{2.14}$$

Working in a similar manner, we find the $\boldsymbol{\omega}$ components for the inertial frame of reference. [11]

$$\begin{aligned} \boldsymbol{\omega}_X &= (\dot{\psi} \sin\phi \sin\theta + \dot{\theta} \cos\phi)\mathbf{e}_X \\ \boldsymbol{\omega}_Y &= (-\dot{\psi} \cos\phi \sin\theta + \dot{\theta} \sin\phi)\mathbf{e}_Y \\ \boldsymbol{\omega}_Z &= (\dot{\psi} \cos\theta + \dot{\phi})\mathbf{e}_Z. \end{aligned} \tag{2.15}$$

2.3 Rotation around a fixed point

2.3.1 Moment of Inertia tensor

The fundamental equation of motion of a rotating rigid body is

$$\frac{d\mathbf{L}}{dt} = \mathbf{T} \tag{2.16}$$

where $\mathbf{L} = \sum \mathbf{r}_i \times m_i \mathbf{v}_i$ is the total angular momentum of the body (about the origin), and $\mathbf{T} = \sum \mathbf{r}_i \times \mathbf{F}_i$ the total external torque (about the origin).

We shall now look into this equation in more detail. The general equation for the velocity of each mass element of the rigid body is

$$\mathbf{v}_i = \mathbf{v}_o + \boldsymbol{\omega} \times \mathbf{r}_i \tag{2.17}$$

with $\mathbf{v}_o$ representing the velocity of the translational motion and $\boldsymbol{\omega} \times \mathbf{r}_i$ representing the velocity of the rotational motion. For the purposes of this thesis we will consider only a rotational motion of the rigid body about an axis passing through the origin (so that $\mathbf{v}_o = 0$) .

The total angular momentum can now be written as

$$\mathbf{L} = \sum m_i \mathbf{r}_i \times (\boldsymbol{\omega} \times \mathbf{r}_i) = \sum m_i \big[{\mathbf{r}_i}^2 \boldsymbol{\omega} - (\mathbf{r}_i \cdot \boldsymbol{\omega})\mathbf{r}_i \big] \tag{2.18}$$

where use has been made of some standard vector identities. It is very useful to write the above formula in a matrix form

$$\begin{bmatrix} L_x \\ L_y \\ L_z \end{bmatrix} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{2.19}$$

or in a shorter form

$$\mathbf{L} = \mathbf{I}\boldsymbol{\omega} \tag{2.20}$$

where

$$I_{xx} = \sum m_i(y_i^2 + z_i^2) \tag{2.21}$$

$$I_{yy} = \sum m_i(x_i^2 + z_i^2) \tag{2.22}$$

$$I_{zz} = \sum m_i(y_i^2 + x_i^2) \tag{2.23}$$

$$I_{xy} = I_{yx} = -\sum m_i x_i y_i \tag{2.24}$$

$$I_{yz} = I_{zy} = -\sum m_i y_i z_i \tag{2.25}$$

$$I_{xz} = I_{zx} = -\sum m_i x_i z_i. \tag{2.26}$$

The I matrix is also called the moment of inertia tensor. As for its values, I_{xx} is called the moment of inertia about the x axis, I_{xy} is called the xy product of inertia, I_{yy} is called the moment of inertia about the y axis, etc. In (2.20) it is understood that $\mathbf{L}$ and $\boldsymbol{\omega}$ are both column vectors and $\mathbf{I}$ is a matrix that represents the moment of inertia tensor.

Generally, the angular momentum vector, $\mathbf{L}$, does not necessarily point in the same direction as the angular velocity vector, $\boldsymbol{\omega}$, that obviously means that generally $\mathbf{L}$ is not parallel to $\boldsymbol{\omega}$.

There are two more things left to be emphasized. The first is that the moment of inertia tensor is symmetric and the second and final is that even though all the above were obtained considering a fixed angular velocity, they also apply at each instant time if the angular velocity varies.

2.3.2 Rotational Kinetic energy

Continuing with the consideration that the rigid body under study only rotates and doesn't transfer, its kinetic energy takes the form

$$K = \frac{1}{2}\sum m_i(\boldsymbol{\omega} \times \mathbf{r}_i)^2 = \frac{1}{2}\boldsymbol{\omega} \cdot \sum m_i \mathbf{r}_i \times (\boldsymbol{\omega} \times \mathbf{r}_i) \tag{2.27}$$

Hence, it follows that

$$K = \frac{1}{2}\boldsymbol{\omega} \cdot \mathbf{L}. \tag{2.28}$$

When using the (2.20) formula the kinetic energy can also be written as

$$K = \frac{1}{2}\boldsymbol{\omega}^\intercal \mathbf{I} \boldsymbol{\omega} \tag{2.29}$$

where $\boldsymbol{\omega}^\intercal$ is the row vector of the angular velocity, the transpose of the column vector $\boldsymbol{\omega}$. If we expand this equation we can get the kinetic energy in terms of components

$$K = \frac{1}{2}(I_{xx}{\omega_x}^2 + I_{yy}{\omega_y}^2 + I_{zz}{\omega_z}^2 + 2I_{xy}\omega_x\omega_y + 2I_{yz}\omega_y\omega_z + 2I_{xz}\omega_x\omega_z). \tag{2.30}$$

2.3.3 Principal Axes of Rotation

As we previously noted the angular momentum vector is not necessarily parallel to the angular velocity vector. We also noted that the moment of inertia tensor represents a real symmetric three-dimensional matrix, which depends on the frame of reference. From the matrix theory we know that the tensor possesses three mutually orthogonal eigenvectors which are associated with three real eigenvalues. Considering that the eigenvectors can be normalized to unit vectors, we can write

$$\tilde{\mathbf{I}}\hat{\boldsymbol{\omega}}_i = \lambda_i \hat{\boldsymbol{\omega}}_i. \tag{2.31}$$

The directions of the three mutually orthogonal unit vectors $\hat{\boldsymbol{\omega}}_i$ define the three so called principal axes of rotation. These three axes are perpendicular to one another and have the property when the rigid body rotates about one of them, the angular momentum vector is parallel to the angular velocity vector.

We choose a coordinate system, body-fixed, with its axes parallel to the principal directions of the moment of inertia tensor. In this new reference frame, the eigenvectors of $\tilde{\mathbf{I}}$ are the unit vectors, $\mathbf{e}_x$, $\mathbf{e}_y$, and $\mathbf{e}_z$, and the eigenvalues are the moments of inertia about these axes, I_{xx}, I_{yy}, and I_{zz}, respectively. Here, we must note that the products of inertia are all zero in the new reference frame. Hence, it is easy to understand that the moment of inertia tensor takes now the form of a diagonial matrix

$$\tilde{\mathbf{I}} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \tag{2.32}$$

Thus, we express the (2.20) in our new system as seen below and it becomes obvious how the angular momentum is paraller to the angular velocity

$$\mathbf{L} = I_{xx}\,\omega_x\,\mathbf{e}_x + I_{yy}\,\omega_y\,\mathbf{e}_y + I_{zz}\,\omega_z\,\mathbf{e}_z. \tag{2.33}$$

Furthermore, the kinetic energy reduces to

$$K = \frac{1}{2}(I_{xx}\,{\omega_x}^2 + I_{yy}\,{\omega_y}^2 + I_{zz}\,{\omega_z}^2). \tag{2.34}$$

2.3.4 Euler's Equations

The fundamental equation of motion of a rotating rigid body we already mentioned above

$$\frac{d\mathbf{L}}{dt} = \mathbf{T} \tag{2.35}$$

works only in an inertial frame. However, from our previous analysis we saw that it is very convenient to express $\mathbf{L}$ in a frame of reference whose axes are aligned along the principal axes of rotation of the body. This frame rotates with the body so it is non-inertial. Since the body-fixed frame co-rotates with the body, it has the same instantaneous angular

velocity. The formula connecting the time derivative of the angular momentum in the inertial frame and the one in the rotating frame is the following

$$\frac{d\mathbf{L}}{dt} = \frac{d_{rot}\mathbf{L}}{dt} + \boldsymbol{\omega} \times \mathbf{L}. \tag{2.36}$$

Since we know that the angular momentum and the angular velocity vector represented in the above rotating frame are

$$\boldsymbol{\omega} = \omega_x \mathbf{e}_x + \omega_y \mathbf{e}_y + \omega_z \mathbf{e}_z \tag{2.37}$$

$$\mathbf{L} = I_{xx}\omega_x \mathbf{e}_x + I_{yy}\omega_y \mathbf{e}_y + I_{zz}\omega_z \mathbf{e}_z \tag{2.38}$$

we can expand (2.36)

$$\frac{d\mathbf{L}}{dt} = \frac{d_{rot}\mathbf{L}}{dt} + \boldsymbol{\omega} \times \mathbf{L} = (I_{xx}\dot{\omega}_x \mathbf{e}_x + I_{yy}\dot{\omega}_y \mathbf{e}_y + I_{zz}\dot{\omega}_z \mathbf{e}_z) + \boldsymbol{\omega} \times \mathbf{L}$$

and after combining the above with the fundamental equation (2.35) we finally get a system of three differential equations where T_x, T_y and T_z are the components of the external torque about Ox, Oy and Oz of the rotating frame, respectively.

$$\begin{aligned} I_{xx}\dot{\omega}_x - (I_{yy} - I_{zz})\omega_y\omega_z &= T_x \\ I_{yy}\dot{\omega}_y - (I_{zz} - I_{xx})\omega_z\omega_x &= T_y \\ I_{zz}\dot{\omega}_z - (I_{xx} - I_{yy})\omega_x\omega_y &= T_z. \end{aligned} \tag{2.39}$$

This system of equations is known as *Euler's equations*. The unknown variables of this system are the instantaneous ω components.

If we want to determine the position of the rigid body rotating about the fixed point O, we need the three Euler angles (ϕ, ψ, θ). Hence, the above system cannot be solved on its own. The additional equations we will need are (2.14) that connect the ω components to Euler angles. The solution of our six differential equations system shows the motion of the body in space, since we find the $\phi(t)$, $\psi(t)$, $\theta(t)$ and the instantaneous angular velocity in terms of the rotating frame of reference, since we find the $\omega_x(t)$, $\omega_y(t)$, $\omega_z(t)$.

Let us now investigate a more special case in which the torque of external forces is zero ($\mathbf{T} = 0$). In this case Euler's equations take the form

$$\begin{aligned} I_{xx}\dot{\omega}_x - (I_{yy} - I_{zz})\omega_y\omega_z &= 0 \\ I_{yy}\dot{\omega}_y - (I_{zz} - I_{xx})\omega_z\omega_x &= 0 \\ I_{zz}\dot{\omega}_z - (I_{xx} - I_{yy})\omega_x\omega_y &= 0. \end{aligned} \tag{2.40}$$

This system can be solved without the use of equations (2.14) and give the position of the angular velocity vector in the rotating frame. It is easy to verify, since there are no external torques, that the vector $\mathbf{L}$ is a constant of the motion in the inertial frame. Furthermore, as seen from (2.16), $d\mathbf{L}/dt = 0$, so the scalar product of this equation with

L leads to $dL^2/dt = 0$. In other words, what this shows is that the magnitude of the vector **L** is also constant.

$$L^2 = {I_{xx}}^2{\omega_x}^2 + {I_{yy}}^2{\omega_y}^2 + {I_{zz}}^2{\omega_z}^2 = constant \tag{2.41}$$

Another constant of the motion is the kinetic energy. To prove that, let us multiply the three equations of (2.40) with ω_x, ω_y and ω_z respectively and add them by parts:

$$\begin{aligned} I_{xx}\dot{\omega}_x\omega_x + I_{yy}\dot{\omega}_y\omega_y + I_{zz}\dot{\omega}_z\omega_z &= 0 \\ \frac{1}{2}\frac{d}{dt}(I_{xx}{\omega_x}^2 + I_{yy}{\omega_y}^2 + I_{zz}{\omega_z}^2) &= 0. \end{aligned}$$

It is now clear that the kinetic energy remains constant as it follows

$$K = \frac{1}{2}(I_{xx}{\omega_x}^2 + I_{yy}{\omega_y}^2 + I_{zz}{\omega_z}^2) = constant. \tag{2.42}$$

Our main concern, although all of the above are useful, is to find the position of the body in space. To do that let us consider at first an inertial frame where the direction of the Z axis coincides with the direction of the angular momentum vector

$$\mathbf{L} = L\mathbf{e}_Z \tag{2.43}$$

where $\mathbf{e}_Z$ is the unit vector pointing in the direction of Z axis of the inertial frame. We also have **L** expressed in the rotating frame from (2.38). The components of each frame connect with the transformation

$$L_{rotating} = DL_{inertial}$$

where D is the transformation matrix given by (2.4). After following some basic mathematical operations we arrive at

$$\begin{aligned} I_{xx}\omega_x &= L\sin\psi\sin\theta \\ I_{yy}\omega_y &= L\cos\psi\sin\theta \\ I_{zz}\omega_z &= L\cos\theta \end{aligned} \tag{2.44}$$

Considering the solution $\omega(t)$ known, we eventually come to a result

$$\begin{aligned} \cos\theta &= \frac{I_{zz}\omega_z(t)}{L} \\ \tan\psi &= \frac{I_{xx}\omega_x(t)}{I_{yy}\omega_y(t)} \end{aligned} \tag{2.45}$$

that gives θ and ψ Euler angles as a function of time. To find ϕ, the third equation of (2.14) is needed to take the form

$$\dot{\phi} = \frac{\omega_z(t) - \dot{\psi}}{\cos\theta}. \tag{2.46}$$

Consequently, ϕ is calculated by a simple integration, since $\dot{\psi}$ is a known function of time. The above three equations refer to the inertial frame.

The above analysis covers the rotation of the rigid body about a fixed point. Let us now investigate the case of a freely rotating rigid body when a symmetry axis exists. This means that two out of three principal axes have the same length, leading to equal moments of inertia. Assuming that $I_{xx} = I_{yy}$, so that the z axis is the symmetry axis, (2.40) takes the form

$$\begin{aligned} I_{xx}\dot{\omega}_x &= (I_{xx} - I_{zz})\omega_y\omega_z \\ I_{xx}\dot{\omega}_y &= -(I_{xx} - I_{zz})\omega_x\omega_z \\ I_{zz}\dot{\omega}_z &= 0 \end{aligned} \tag{2.47}$$

The last equation gives right away

$$\omega_z = constant. \tag{2.48}$$

From the first two equations we eliminate ω_y and we take the differential equation with respect to ω_x

$$\ddot{\omega}_x + \left[\left(\frac{I_{xx} - I_{zz}}{I_{xx}}\right)^2 {\omega_z}^2\right]\omega_x = 0. \tag{2.49}$$

This is a linear second order differential equation and its solution is of the form

$$\omega_x = A\sin(Bt + C) \tag{2.50}$$

where

$$B = \frac{I_{xx} - I_{zz}}{I_{xx}}\omega_z \tag{2.51}$$

and A is a constant depending on the initial conditions. Note here that we choose t_0, such that $\omega_x(t_0) = \omega_x(0) = 0$. Hence, the constant C is zero. By replacing the solution of ω_x in the second equation of (2.47), we find

$$\omega_y = A\cos(Bt). \tag{2.52}$$

Hence, the angular velocity vector is given analytically by

$$\boldsymbol{\omega} = A\sin(Bt)\mathbf{e}_x + A\cos(Bt)\mathbf{e}_y + \omega_z\mathbf{e}_z. \tag{2.53}$$

Note here that $\omega^2 = A^2 + {\omega_z}^2 = constant$ and that the angular velocity vector is not constant over the rotating system $Oxyz$. In fact, it rotates about z axis with constant angular velocity B (see figure 13).

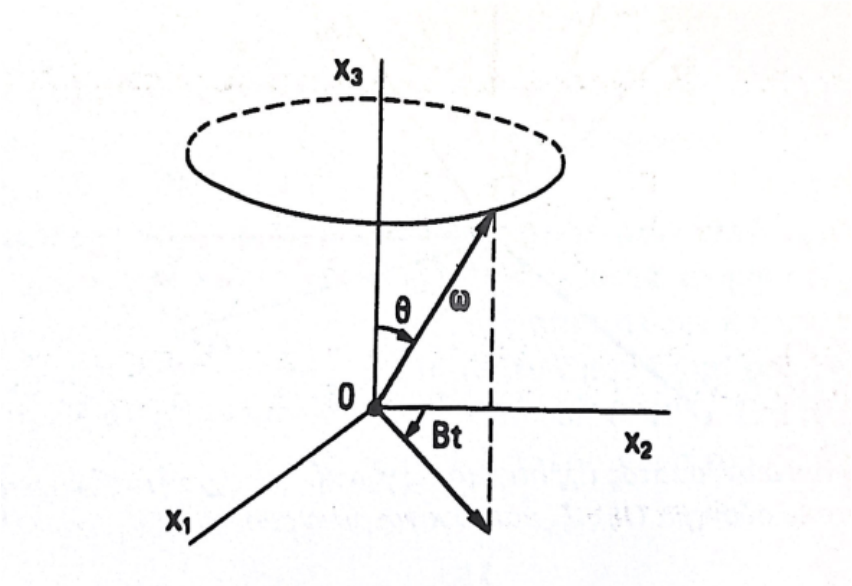


Figure 13: Angular velocity vector.

On another note, we could use the $\omega(t)$ solution to compute the Euler angles from the analysis above. In that way, from (2.45) and (2.46), we have

$$\theta = constant \tag{2.54}$$

$$\psi = Bt \tag{2.55}$$

$$\phi = \frac{L}{I_{xx}}t \tag{2.56}$$

for $\phi(0) = 0$, representing the rotation of the rigid body in the inertial frame.

In the body-fixed frame, the angular velocity vector *precesses* about the z axis (the symmetry axis) with the angular frequency B, tracing out a conical surface around it. In the figure below, the frequency B is represented by the Ω and the direction of z axis of the rotating frame is represented by the unit vector $\hat{\mathbf{n_3}}$. The angular velocity vector also traces out a conical surface around the Z axis of the inertial frame (and rotates about the Z axis with a constant angular velocity L/I_{xx}). The two cones osculate to the instantaneous rotation axis and their axes form a constant angle θ. Assuming that α is the angle between the angular velocity vector and the z axis of the rotating frame, then there is a formula connecting these two and separates two cases for the relative position of the cones.

$$\tan\alpha = \frac{I_{zz}}{I_{xx}}\tan\theta \tag{2.57}$$

From the above equation, the conclusion that easily follows is that when $I_{xx} < I_{zz}$, then $\alpha > \theta$ and the first case (a) applies as seen below. When $I_{xx} > I_{zz}$, then $\alpha < \theta$ and the second case (b) applies.

All the above conclusions refer to freely rotating rigid body, as well as to a rotating rigid body about a fixed point.

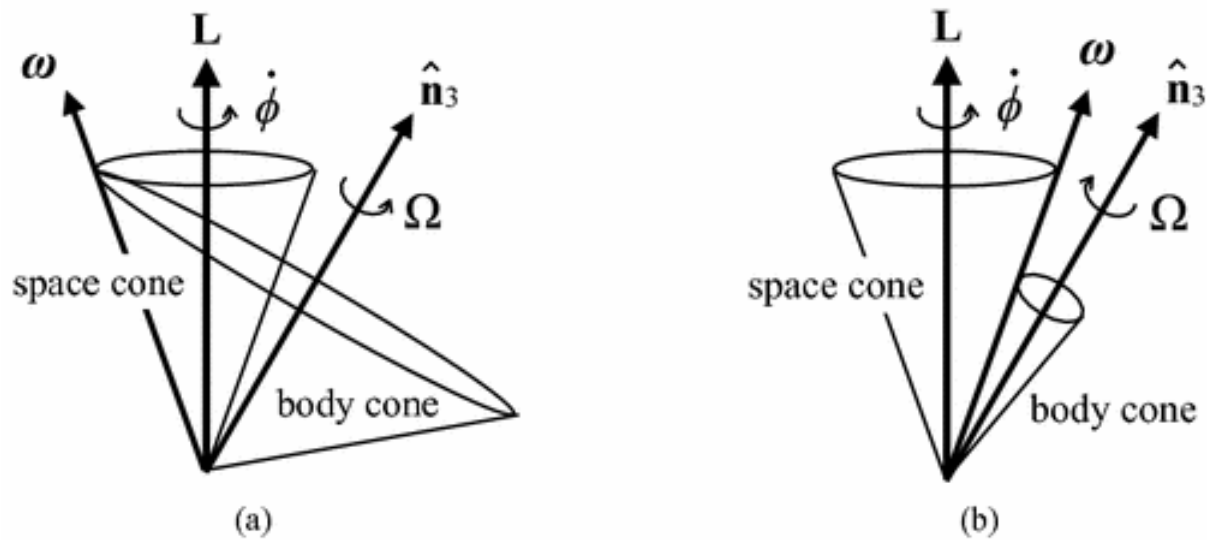


Figure 14: The two cases of the conical surface.

2.4 Quaternions

There is a better and more effective way to represent a rotating rigid body than using a 3×3 rotation matrix. Unit quaternions are used to represent orientations and rotations of elements in three dimensional space. They include information about an axis-angle representation of a rotation around an arbitrary axis. In mathematics, an axis-angle representation of a rotation parameterizes a rotation in a three dimensional Euclidean space by two quantities: a unit vector $\mathbf{e}$ indicating the direction of an axis of rotation, and an angle θ describing the magnitude of the rotation about the axis.

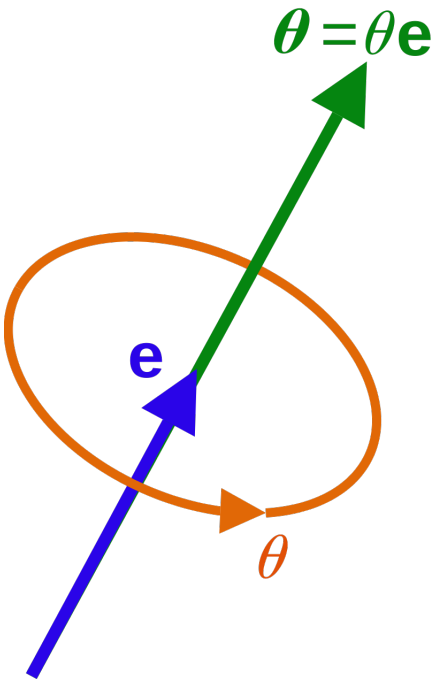


Figure 15: The angle θ and the axis unit vector $\mathbf{e}$ define a rotation.

Before we introduce a more detailed analysis of the quaternion system, let us define a spatial rotation using quaternions, considering a quaternion has four components. A spatial rotation around a fixed point of θ radians about a unit axis (X, Y, Z) is given by the quaternion

$$(C, X\,S, Y\,S, Z\,S) \tag{2.58}$$

where $C = \cos(\theta/2)$ and $S = \sin(\theta/2)$. This information will be useful further on.

2.4.1 Definition

The **quaternion** number system extends the complex numbers and were first described and defined by William Rowan Hamilton (1843). This system is mostly used in pure mathematics as well as in applied mathematics and physics, especially for rotations in three-dimensional space. It can be used alongside other methods of rotation, such as Euler angles and rotation matrices, or as an alternative to them, depending on the application. Hamilton's definition set a quaternion to be the quotient of two directed lines in a three-dimensional space or the quotient of two vectors.

The mathematical form describing a quaternion is the following

$$q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \tag{2.59}$$

where q_0, q_1, q_2, and q_3 are real numbers and i, j, and k are the basic quaternions. The quaternion $q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$ is called the vector part (sometimes imaginary part) of q, and q_0 is the scalar part (sometimes real part) of q.

The multiplication rules for the basis elements $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are

$$\begin{gathered} \mathbf{i}1 = 1\mathbf{i} = \mathbf{i}\,, \quad \mathbf{j}1 = 1\mathbf{j} = \mathbf{j}\,, \quad \mathbf{k}1 = 1\mathbf{k} = \mathbf{k} \\ \mathbf{i^2} = \mathbf{j^2} = -1 \\ \mathbf{ij} = \mathbf{k}\,, \quad \mathbf{ji} = -\mathbf{k} \end{gathered} \tag{2.60}$$

It is then, easily obtained that the remaining product rules are

$$\begin{gathered} \mathbf{jk} = \mathbf{i}\,, \quad \mathbf{kj} = -\mathbf{i} \\ \mathbf{ki} = \mathbf{j}\,, \quad \mathbf{ik} = -\mathbf{j} \\ \mathbf{ijk} = -1\,, \quad \mathbf{k^2} = -1 \end{gathered} \tag{2.61}$$

Figure 16 depices the above multiplication rules. The non commutativity of multiplication is depicted by colored squares.

×	1	i	j	k
1	1	i	j	k
i	i	−1	k	−j
j	j	−k	−1	i
k	k	j	−i	−1

Figure 16: Basic quaternions multiplication table.

2.4.2 Basic algebraic properties

Addition and Multiplication
Addition of two or more quaternions acts component wise. So, let's consider for example the quaternion q from above (1.1) and another quaternion

$$p = p_0 + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k}.$$

Adding those two quaternions leads to

$$p + q = (p_0 + q_0) + (p_1 + q_1)\mathbf{i} + (p_2 + q_2)\mathbf{j} + (p_3 + q_3)\mathbf{k}. \tag{2.62}$$

Moreover, every quaternion q has a negative $-q$ with components $-q_i$, $i = 0, 1, 2, 3$.

As far as multiplication is concerned, the product of the two elements q and p, called *Hamilton product*, is based on the products of the basis elements and the distributive law, that helps us to expand the product so that it becomes a sum of products of basis elements. More specifically, we have

$$\begin{aligned} pq &= (p_0 + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k})(q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}) \\ &= p_0q_0 + p_0q_1\mathbf{i} + p_0q_2\mathbf{j} + p_0q_3\mathbf{k} \\ &+ p_1q_0\mathbf{i} + p_1q_1\mathbf{i}^2 + p_1q_2\mathbf{ij} + p_1q_3\mathbf{ik} \\ &+ p_2q_0\mathbf{j} + p_2q_1\mathbf{ji} + p_2q_2\mathbf{j}^2 + p_2q_3\mathbf{jk} \\ &+ p_3q_0\mathbf{k} + p_3q_1\mathbf{ki} + p_3q_2\mathbf{kj} + p_3q_3\mathbf{k}^2. \end{aligned} \tag{2.63}$$

Now, using the multiplication rules for the basis elements $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ from (1.2) and (1.3), we can get a more concise form as seen below

$$\begin{aligned} pq &= p_0q_0 - p_1q_1 - p_2q_2 - p_3q_3 \\ &+ (p_0q_1 + p_1q_0 + p_2q_3 - p_3q_2)\mathbf{i} \\ &+ (p_0q_2 - p_1q_3 + p_2q_0 + p_3q_1)\mathbf{j} \\ &+ (p_0q_3 + p_1q_2 - p_2q_1 + p_3q_0)\mathbf{k}. \end{aligned} \tag{2.64}$$

Conjugation, the norm and reciprocal

Conjugation of quaternions is analogous to conjugation of complex numbers. Let's consider a quaternion q similar to the above (1.1). Its conjugate will then be

$$q^* = q = q_0 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k}. \tag{2.65}$$

The norm of the quaternion q is now defined by the square root of the product of it and its conjugate and is denoted $||q||$. The mathematical expression goes as follows

$$||q|| = \sqrt{qq^*} = \sqrt{q^*q} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}. \tag{2.66}$$

It easily follows that a quaternion of norm one is a unit quaternion. Dividing a non-zero quaternion q by its norm produces a unit quaternion $\mathbf{U}_q$ called the *versor* of q

$$\mathbf{U}_q = \frac{q}{||q||}. \tag{2.67}$$

Taking into consideration conjugation and the norm we are now able to define the reciprocal of a non-zero quaternion. The product of a quaternion with its reciprocal should equal 1, as well as the product of q and $\frac{q^*}{||q||^2}$. Hence, the reciprocal of q is defined as follows

$$q^{-1} = \frac{q^*}{||q||^2}. \tag{2.68}$$

In our calculations, the condition that the norm of the quaternions equals to one should be satisfied, so we have

$$\sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1. \tag{2.69}$$

2.4.3 Quaternion Derivations

Let us now change the notation of the quaternion so we can easily derive the formula for $\dot{q}(t)$ later on. We will write a quaternion $q = s + v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$ as the pair

$$[s, v].$$

Considering this, a rotation of θ radians around a unit axis u is represented by the unit quaternion

$$[\cos(\theta/2), \sin(\theta/2)u].$$

as discussed earlier in this chapter (see (2.58)). Hence, the scalar part is $\cos(\theta/2)$ and the vector part is multiplied by $\sin(\theta/2)$. Generally, if q_1 and q_2 indicate rotations, then q_2q_1 represents the composite rotation of q_1 followed by q_2. Before showing how the rotation of a body is expressed using quaternions, we need to derive a formula for $\dot{q}(t)$. First of all, we express the angular velocity as a vector $\omega(t)$ with magnitude $|\omega(t)|$. Naturally, the body rotates about $\omega(t)$ axis. Therefore, the rotation of the body after a period of time Δt is represented by the quaternion

$$[\cos\frac{|\omega(t)|\Delta t}{2}, \sin\Big(\frac{|\omega(t)|\Delta t}{2}\Big)\frac{\omega(t)}{|\omega(t)|}].$$

At times $t_0 + \Delta t$ (for small Δt), the orientation of the body is (to within first order) the combination of two rotations; a rotation by q_0 followed by a rotation with angular velocity $\omega(t_0)$ for Δt time.

$$q(t_0 + \Delta t) = [\cos\frac{|\omega(t_0)|\Delta t}{2}, \sin\Big(\frac{|\omega(t_0)|\Delta t}{2}\Big)\frac{\omega(t_0)}{|\omega(t_0)|}]q(t_0). \tag{2.70}$$

We substitute in (2.70) $t = t_0 + \Delta t$ and so we can express the above as

$$q(t) = [\cos\frac{|\omega(t_0)|(t - t_0)}{2}, \sin\Big(\frac{|\omega(t_0)|(t - t_0)}{2}\Big)\frac{\omega(t_0)}{|\omega(t_0)|}]q(t_0). \tag{2.71}$$

Differentiating the expression (2.71) at a time t_0, and since $q(t_0)$ is a constant, we gradually get

$$\frac{d}{dt}\cos\frac{|\omega(t_0)|(t - t_0)}{2} = -\frac{|\omega(t_0)|}{2}\sin\frac{|\omega(t_0)|(t - t_0)}{2} = -\frac{|\omega(t_0)|}{2}\sin 0 = 0 \tag{2.72}$$

$$\frac{d}{dt}\sin\frac{|\omega(t_0)|(t - t_0)}{2} = \frac{|\omega(t_0)|}{2}\cos\frac{|\omega(t_0)|(t - t_0)}{2} = \frac{|\omega(t_0)|}{2}\cos 0 = \frac{|\omega(t_0)|}{2} \tag{2.73}$$

Thus, combining (2.72) and (2.73), at at time t_0, $\dot{q}(t)$ is expressed as follows

$$\begin{aligned}
\dot{q}(t) &= \frac{d}{dt}\Big([\cos\frac{|\omega(t_0)|(t-t_0)}{2}, \sin\frac{|\omega(t_0)|(t-t_0)}{2}\frac{\omega(t_0)}{|\omega(t_0)|}]q(t_0)\Big) \\
&= \frac{d}{dt}\Big([\cos\frac{|\omega(t_0)|(t-t_0)}{2}, \sin\frac{|\omega(t_0)|(t-t_0)}{2}\frac{\omega(t_0)}{|\omega(t_0)|}]\Big)q(t_0) \\
&= [0, \frac{|\omega(t_0)|}{2}\frac{\omega(t_0)}{\omega(t_0)|}]q(t_0) \\
&\quad [0, \frac{1}{2}\omega(t_0)]q(t_0) \\
&= \frac{1}{2}[0, \omega(t_0)]q(t_0).
\end{aligned} \tag{2.74}$$

The product $[0, \omega(t_0)]q(t_0)$ is identical to the expression $\omega(t_0)q(t_0)$. In other words, the term $[0, \omega(t_0)]$ represents the angular velocity, which is of course a vector, as a quaternion with zero scalar part. Finally, the general expression for the $\dot{q}(t)$ is derived

$$\dot{q}(t) = \frac{1}{2}\omega(t)q(t). \tag{2.75}$$

The differential equation (2.75) could be further analyzed for every component of the quaternion q, with additionally using the multiplication formula (2.64), as follows

$$\begin{aligned}
\dot{q_0}(t) &= \frac{1}{2}(-\omega_1(t)q_1(t) - \omega_2(t)q_2(t) - \omega_3(t)q_3(t)) \\
\dot{q_1}(t) &= \frac{1}{2}(\omega_1(t)q_0(t) + \omega_2(t)q_3(t) - \omega_3(t)q_2(t)) \\
\dot{q_2}(t) &= \frac{1}{2}(\omega_2(t)q_0(t) - \omega_1(t)q_3(t) + \omega_3(t)q_1(t)) \\
\dot{q_3}(t) &= \frac{1}{2}(\omega_1(t)q_2(t) - \omega_2(t)q_1(t) + \omega_3(t)q_0(t)).
\end{aligned} \tag{2.76}$$

It is necessary to notice the fact that in formula (2.75) the angular velocity vector $\omega(t) = (\omega_X(t), \omega_Y(t), \omega_Z(t))$ is represented by projections on axes of the inertial frame of reference. In case we want to express the angular velocity vector components in respect to the rotating frame, we follow the transformation below

$$(0, \omega_X(t), \omega_Y(t), \omega_Z(t)) = q(t)(0, \omega_x(t), \omega_y(t), \omega_z(t))q^{-1}(t)$$

where $\omega_x(t)$, $\omega_y(t)$, $\omega_z(t)$ are the angular velocity projections on the axes of the rotating frame. After taking these into consideration and also using (2.75) we get

$$\dot{q}(t) = \frac{1}{2}q(t)\overline{\omega}(t), \tag{2.77}$$

with $\overline{\omega}(t) = (0, \omega_x(t), \omega_y(t), \omega_z(t))$. It is now obvious that (2.77) also can be expressed

in the following form

$$\begin{aligned}
\dot{q_0}(t) &= \frac{1}{2}(-q_1(t)\omega_x(t) - q_2(t)\omega_y(t) - q_3(t)\omega_z(t)) \\
\dot{q_1}(t) &= \frac{1}{2}(q_0(t)\omega_x(t) - q_3(t)\omega_y(t) + q_2(t)\omega_z(t)) \\
\dot{q_2}(t) &= \frac{1}{2}(q_3(t)\omega_x(t) + q_0(t)\omega_y(t) - q_1(t)\omega_z(t)) \\
\dot{q_3}(t) &= \frac{1}{2}(-q_2(t)\omega_x(t) + q_1(t)\omega_y(t) + q_0(t)\omega_z(t)).
\end{aligned} \tag{2.78}$$

2.4.4 Quaternion, Euler angles, rotation matrix

It is now time to express our rotation formula to quaternion in order to get rid of the problems that go along with the rotation matrix and Euler angles.

Firstly, let us consider the following three rotation angles around each axis as shown in the figure below. For a rigid body rotating, the rotation around the front-to-back axis is called roll, the rotation around the side-to-side axis is called pitch and the rotation around the vertical axis is called yaw. Notation wise, we denote

$$(roll, pitch, yaw) = (\theta_X, \theta_Y, \theta_Z) \tag{2.79}$$

These angles are also called the Tait–Bryan angles.

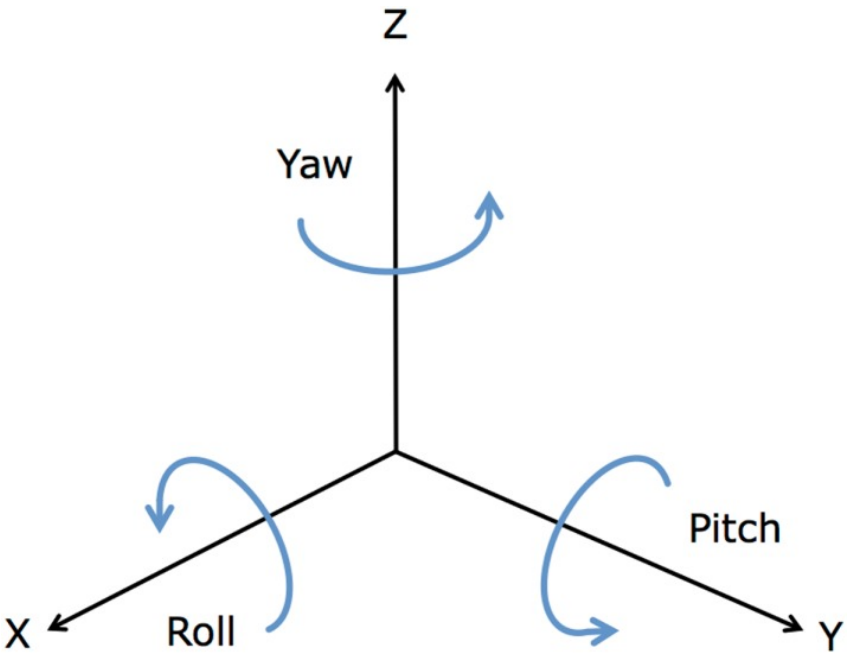


Figure 17: yaw-pitch-roll.

The rotation formula for these angles is similar to the Euler angles we described previously. The only difference is that, if we want to make a rotation of roll-pitch-yaw in

that order, and we multiply the rotations matrices of every angle in the opposite order $D = D_{yaw}D_{pitch}D_{roll}$, we finally take a rotation matrix of the form

$$D = \begin{bmatrix} \cos\theta_Z \cos\theta_Y & \cos\theta_Z \sin\theta_Y \sin\theta_X - \sin\theta_Z \cos\theta_X & \cos\theta_Z \sin\theta_Y \cos\theta_X + \sin\theta_Z \sin\theta_X \\ \sin\theta_Z \cos\theta_Y & \sin\theta_Z \sin\theta_Y \sin\theta_X + \cos\theta_Z \cos\theta_X & \sin\theta_Z \sin\theta_Y \cos\theta_X - \cos\theta_Z \sin\theta_X \\ -\sin\theta_Y & \cos\theta_Y \sin\theta_X & \cos\theta_Y \cos\theta_X \end{bmatrix}. \tag{2.80}$$

This matrix, moves a vector from the body fixed frame of reference to the inertial, like so

$$X = Dx, \tag{2.81}$$

where again x and X are column vectors containing the three position components of the rotating and the inertial frame, respectfully.

In order to express a quaternion in respect to the three angles of rotation around every axis, we need the quaternion form (2.58). Specifically, the quaternion representing a rotation of θ_X angle around X axis is

$$q_X = (\cos\frac{\theta_X}{2}, \sin\frac{\theta_X}{2}, 0, 0), \tag{2.82}$$

the quaternion representing a rotation of θ_Y angle around Y axis is

$$q_Y = (\cos\frac{\theta_Y}{2}, 0, \sin\frac{\theta_Y}{2}, 0), \tag{2.83}$$

and finally the quaternion representing a rotation of θ_Z angle around Z axis is

$$q_Z = (\cos\frac{\theta_Z}{2}, 0, 0, \sin\frac{\theta_Z}{2}). \tag{2.84}$$

A sequence of roll, pitch, yaw angle rotations is equivalent to the quaternion product $q_Z q_Y q_X$. Using the quaternion multiplication rule in (2.64) and denoting the quaternion product $q_E = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$, we can easily express its components in terms of θ_X, θ_Y and θ_Z angles.

$$\begin{aligned} q_0 &= \cos\frac{\theta_X}{2}\cos\frac{\theta_Y}{2}\cos\frac{\theta_Z}{2} + \sin\frac{\theta_X}{2}\sin\frac{\theta_Y}{2}\sin\frac{\theta_Z}{2} \\ q_1 &= \sin\frac{\theta_X}{2}\cos\frac{\theta_Y}{2}\cos\frac{\theta_Z}{2} - \cos\frac{\theta_X}{2}\sin\frac{\theta_Y}{2}\sin\frac{\theta_Z}{2} \\ q_2 &= \cos\frac{\theta_X}{2}\sin\frac{\theta_Y}{2}\cos\frac{\theta_Z}{2} + \sin\frac{\theta_X}{2}\cos\frac{\theta_Y}{2}\sin\frac{\theta_Z}{2} \\ q_3 &= \cos\frac{\theta_X}{2}\cos\frac{\theta_Y}{2}\sin\frac{\theta_Z}{2} - \sin\frac{\theta_X}{2}\sin\frac{\theta_Y}{2}\cos\frac{\theta_Z}{2}. \end{aligned} \tag{2.85}$$

There is of course, the reverse transformation that gives us the angles as a function of the four quaternions components:

$$\begin{bmatrix} roll \\ pitch \\ yaw \end{bmatrix} = \begin{bmatrix} \arctan 2[2(q_0q_1 + q_2q_3), 1 - 2({q_1}^2 + {q_2}^2)] \\ \arcsin[2(q_0q_2 - q_1q_3)] \\ \arctan 2[2(q_0q_3 + q_1q_2), 1 - 2({q_2}^2 + {q_3}^2)] \end{bmatrix} \tag{2.86}$$

In addition, the rotation matrix with the use of quaternions is

$$D = \begin{bmatrix} 1 - 2({q_2}^2 + {q_3}^2) & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & 1 - 2({q_1}^2 + {q_3}^2) & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & 1 - 2({q_1}^2 + {q_2}^2). \end{bmatrix} \tag{2.87}$$

Of course, this matrix is equivalent to the one from the equation (2.80) and the transformation from rotating frame of reference to the inertial is given from the (2.81). As discussed before, the reverse transformation requires the inverse D matrix (D^{-1}).

3 Orbital mechanics near a rotating asteroid

This chapter examines the gravitational potential an ellipsoid forms around it. Specifying the ellipsoid as an asteroid we then examine the dynamical equations of a particle orbiting a rotating asteroid with such potential.

3.1 Ellipsoid Potential

The gravitational potential of an ellipsoid is given approximately by the following equation [12]

$$U = \Big[-\frac{M}{r} - \frac{I_{xx} + I_{yy} + I_{zz}}{2r^3} + \frac{3}{2}\frac{I_{xx}x^2 + I_{yy}y^2 + I_{zz}z^2}{r^5} \Big] G \tag{3.1}$$

where

$$\begin{aligned} I_{xx} &= M\,\frac{b^2 + c^2}{5} \\ I_{yy} &= M\,\frac{a^2 + c^2}{5} \\ I_{zz} &= M\,\frac{a^2 + b^2}{5} \end{aligned} \tag{3.2}$$

are the moments of inertia of an ellipsoid of uniform density and $I_{xy} = I_{yz} = I_{zx} = 0$. The lengths a, b and c are called the principal semi-axes of the ellipsoid. If $a = b > c$, one has an oblate spheroid; if $a = b < c$, one has a prolate spheroid and if $a = b = c$, then one has a sphere. In the most general case, where $a \neq b \neq c$, one has a triaxial ellipsoid. G is the gravitational constant and M the mass of the ellipsoid. [2]

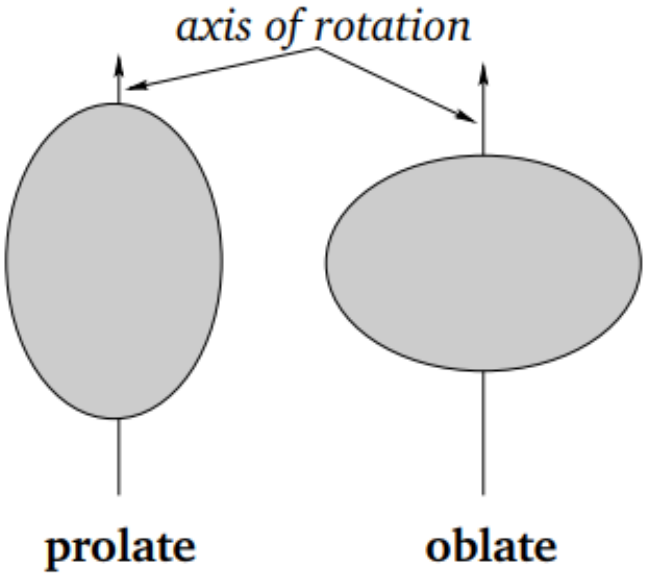


Figure 18: Prolate and oblate spheroids.

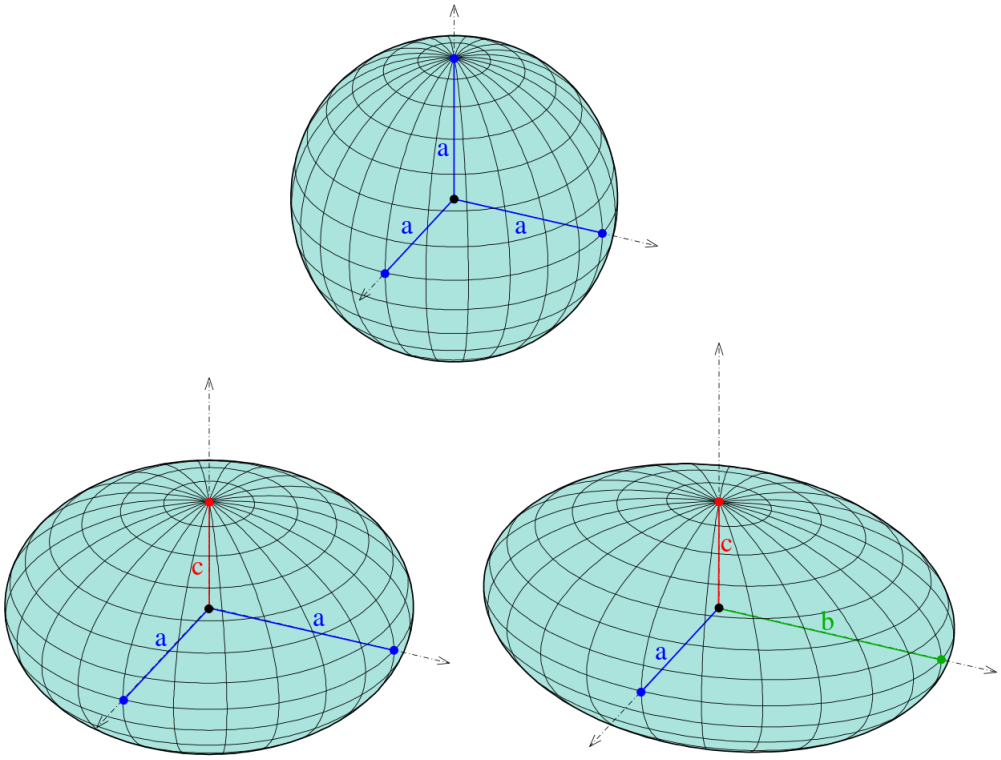


Figure 19: Examples of ellipsoids.(Top:Sphere, Bottom left:Spheroid, Bottom right:Triaxial ellipsoid)[2]

In reality, the potential of an rigid body of arbitrary shape is given by

$$U(\mathbf{r}) = -G \int \frac{dM}{|\boldsymbol{\varrho}|} = -G \int \frac{dM}{|\mathbf{r} - \mathbf{r}'|} \tag{3.3}$$

where $\varrho = \sqrt{(x-\xi)^2 + (y-\eta)^2 + (z-\zeta)^2}$, $\mathbf{r}' = (\xi,\, \eta,\, \zeta)$ is the position of the differential mass element of the rigid body dM and $\mathbf{r} = (x,\, y,\, z)$ is the position of the point particle, where we want to compute the potential (see figure 20).

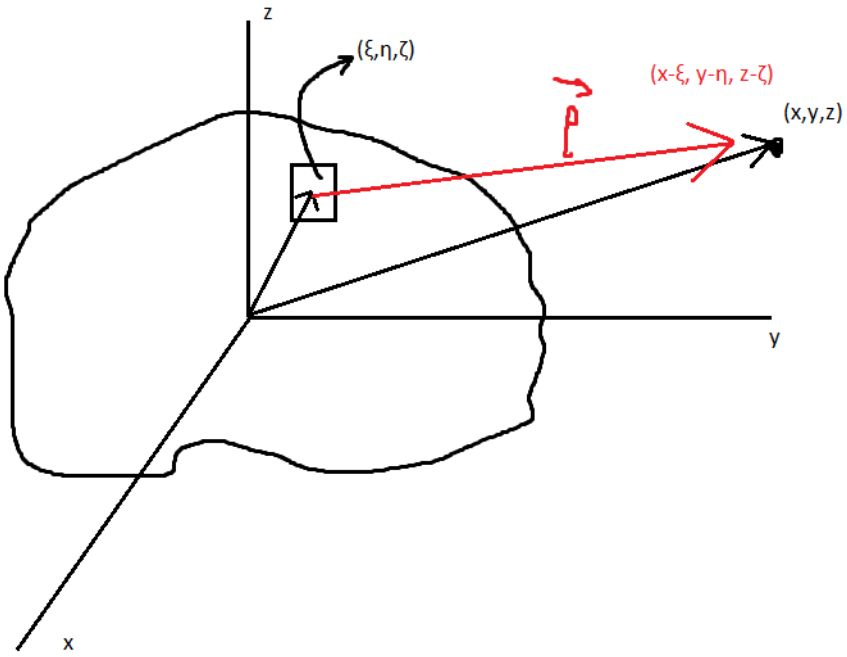


Figure 20: Arbitrary shaped rigid body.

Expanding $\frac{1}{|\boldsymbol{\varrho}|}$ into a second order series, taking into account that

$$\begin{aligned} I_{xx} &= \int (\eta^2 + \zeta^2) dM \\ I_{yy} &= \int (\xi^2 + \zeta^2) dM \\ I_{zz} &= \int (\eta^2 + \xi^2) dM \end{aligned} \tag{3.4}$$

and following some calculations, one can derive the equation (3.1).

3.2 The equation of motion in the classical form

Consider a general case, where a particle orbits a rotating asteroid. Then, the equation describing the motion can be expressed in the body fixed frame as a second-order ordinary differential equation [13]

$$\ddot{\mathbf{r}} + 2\boldsymbol{\omega} \times \dot{\mathbf{r}} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) + \dot{\boldsymbol{\omega}} \times \mathbf{r} + \frac{\partial U(\mathbf{r})}{\partial \mathbf{r}} = 0 \tag{3.5}$$

where $\mathbf{r}$ is the radius vector from the asteroid's center of mass to the particle, its derivatives are with respect to the body-fixed frame of reference, $\boldsymbol{\omega}$ is, as mentioned above, the angular velocity vector and $U(\mathbf{r})$ is the gravitational potential of the asteroid, given by (3.1). The inertial frame is represented in the figure below with orange coloured axes and the body fixed, rotating frame with blue coloured axes. The radius vector of the position of the particle (green), as well as the other vectors (velocity, anglular velocity etc.) in (3.5) are given in the body fixed frame.

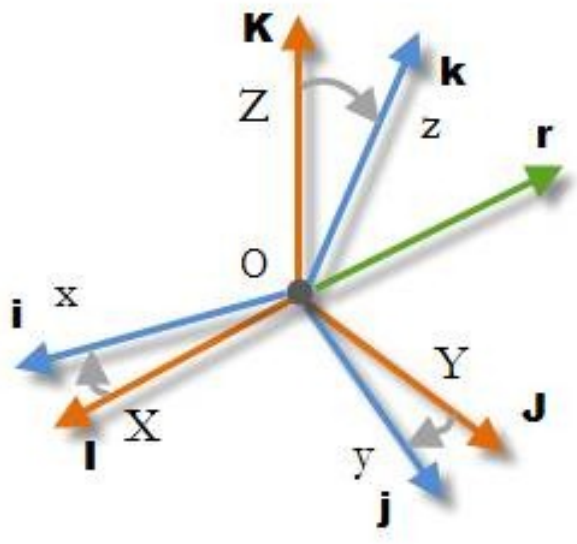


Figure 21: Inertial (orange) and body fixed (blue)frame of reference.

The Lagrangian associated to the motion of a particle in a rotating frame described by (3.5) is [14]

$$\mathcal{L} = \frac{\dot{\mathbf{r}}^2}{2} + \boldsymbol{\omega} \cdot (\mathbf{r} \times \dot{\mathbf{r}}) + \frac{1}{2}(\boldsymbol{\omega} \times \mathbf{r})^2 - U(\mathbf{r}). \tag{3.6}$$

For convenience, we consider for the generalized coordinates $\mathbf{q} = \mathbf{r}$. Hence, the generalized momentum is

$$\mathbf{p} = \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{r}}} = \dot{\mathbf{r}} + \boldsymbol{\omega} \times \mathbf{r}. \tag{3.7}$$

The Hamiltonian is given by

$$\mathcal{H} = \mathbf{p}\dot{\mathbf{q}} - \mathcal{L}. \tag{3.8}$$

Combining (3.6), (3.7) and (3.8), and with some basic calculations, we take the final form of the Hamiltonian

$$\mathcal{H} = \frac{\mathbf{p^2}}{2} - \mathbf{p} \cdot (\boldsymbol{\omega} \times \mathbf{r}) + U(\mathbf{r}). \tag{3.9}$$

If $\boldsymbol{\omega}$ is time invariant, then H is also time invariant and is constant (the Jacobi constant). The mechanical energy can be written as

$$E = K + U(\mathbf{r}) \tag{3.10}$$

where K is the kinetic energy of the particle, which is given by

$$K = \frac{\mathbf{p}^2}{2} = \frac{1}{2}(\dot{\mathbf{r}} + \boldsymbol{\omega} \times \mathbf{r})^2 = \frac{1}{2}\dot{\mathbf{r}}^2 + \dot{\mathbf{r}} \cdot (\boldsymbol{\omega} \times \mathbf{r}) + \frac{1}{2}(\boldsymbol{\omega} \times \mathbf{r})^2. \tag{3.11}$$

We can also write the mechanical energy in terms of the effective potential $V(\mathbf{r})$.

$$E = K + V(\mathbf{r}) \tag{3.12}$$

Let the last be defined as

$$V(\mathbf{r}) = -\frac{1}{2}(\boldsymbol{\omega} \times \mathbf{r})^2 + U(\mathbf{r}). \tag{3.13}$$

The equation of motion (3.5) may be rewritten in terms of the effective potential as

$$\ddot{\mathbf{r}} + 2\boldsymbol{\omega} \times \mathbf{r} + \dot{\boldsymbol{\omega}} \times \mathbf{r} + \frac{\partial V(\mathbf{r})}{\partial \mathbf{r}} = 0. \tag{3.14}$$

3.3 The equation of motion in the scalar form

The radius, velocity, acceleration and angular velocity vectors in (3.5) equation, expressed in the rotating, body fixed frame (blue axes in Figure 21) take the form

$$\begin{aligned} \mathbf{r} &= x\,\mathbf{e_x} + y\,\mathbf{e_y} + z\,\mathbf{e_z} \\ \dot{\mathbf{r}} &= \dot{x}\,\mathbf{e_x} + \dot{y}\,\mathbf{e_y} + \dot{z}\,\mathbf{e_z} \\ \ddot{\mathbf{r}} &= \ddot{x}\,\mathbf{e_x} + \ddot{y}\,\mathbf{e_y} + \ddot{z}\,\mathbf{e_z} \\ \boldsymbol{\omega} &= \omega_x\,\mathbf{e_x} + \omega_y\,\mathbf{e_y} + \omega_z\,\mathbf{e_z} \end{aligned} \tag{3.15}$$

where $\mathbf{e_i}$ with $i = x, y, z$ the unit vectors of every axis of the frame. Hence, the dynamical equations of the orbiting particle (3.5) can be written as [13]

$$\begin{aligned} &\ddot{x} + \dot{\omega}_y z - \dot{\omega}_z y + 2\omega_y \dot{z} - 2\omega_z \dot{y} + \omega_x\omega_y y - {\omega_y}^2 x - {\omega_z}^2 x + \omega_z\omega_x z + \frac{\partial U}{\partial x} = 0 \\ &\ddot{y} + \dot{\omega}_z x - \dot{\omega}_x z + 2\omega_z \dot{x} - 2\omega_x \dot{z} + \omega_y\omega_z z - {\omega_z}^2 y - {\omega_x}^2 y + \omega_x\omega_y x + \frac{\partial U}{\partial y} = 0 \\ &\ddot{z} + \dot{\omega}_x y - \dot{\omega}_y x + 2\omega_x \dot{y} - 2\omega_y \dot{x} + \omega_x\omega_z x - {\omega_x}^2 z - {\omega_y}^2 z + \omega_y\omega_z y + \frac{\partial U}{\partial z} = 0. \end{aligned} \tag{3.16}$$

We can easily transform these equations depending on the angular velocity vector. More specifically, the angular velocity vector could be defined by $\boldsymbol{\omega} = \omega\mathbf{e_z}$, so that its x and y components would be zero, and it could be time invariant so that its derivatives could be zero. In that case, the above equations would be simplified.

$$\begin{aligned} &\ddot{x} - \dot{\omega} y - 2\omega \dot{y} - \omega^2 x + \frac{\partial U}{\partial x} = 0 \\ &\ddot{y} + \dot{\omega} x + 2\omega \dot{x} - \omega^2 y + \frac{\partial U}{\partial y} = 0 \\ &\ddot{z} + \frac{\partial U}{\partial z} = 0. \end{aligned} \tag{3.17}$$

Note here that the solution of the scalar form gives the position of the particle $\mathbf{r} = (x,\, y,\, z)$ is in the body-fixed frame.

3.4 Horizontal stability

Let us now limit our analysis in a plane motion, where $z = 0$, and study the motion of a particle in a central force field. In this plane the gravitational potential from (3.1) takes the form

$$U_{z=0} = \Big[-\frac{M}{r} - \frac{I_{xx} + I_{yy} + I_{zz}}{2r^3} + \frac{3}{2}\frac{I_{xx}x^2 + I_{yy}y^2}{r^5} \Big] G \tag{3.18}$$

where now $r = \sqrt{x^2 + y^2}$. We assume a symmetric rotational body around, so that $I_{xx} = I_{yy}$. Hence,

$$U_{z=0}(r) = \Big[-\frac{M}{r} + \frac{I_{xx} - I_{zz}}{2r^3} \Big] G. \tag{3.19}$$

The gravitational force is

$$F(r) = -\frac{dU_{z=0}}{dr} = \Big[-\frac{M}{r^2} + \frac{3(I_{xx} - I_{zz})}{2r^4} \Big] G. \tag{3.20}$$

The differential equation

$$m\ddot{r} = F(r) + \frac{L}{mr^3} \tag{3.21}$$

describes the motion of a particle in a central force field. The equilibrium points must satisfy the condition

$$m\ddot{r} = 0 \Rightarrow F(r) + \frac{L}{mr^3} = 0. \tag{3.22}$$

It is clear that, if the root of this equation is r_0, the initial conditions $r = r_0$ and $\dot{r} = 0$ correspond to an equilibrium solution that in our case represents a circular motion with center the center of force. Taking into consideration that $\omega = L/mr_0{}^2$ and that $\upsilon = r_0\omega$, we can find the mathematical formula of the velocity needed to launch the particle from distance r_0 so the orbit will be circular. Hence, we have

$$\upsilon^2 = -\frac{r_0\, F(r_0)}{m}. \tag{3.23}$$

The differential equation of motion can also be written as

$$F(r) + \frac{L}{mr^3} = -\frac{dV}{dr} \tag{3.24}$$

where V is the effective potential. The equilibrium points correspond to effective potential's extremums

$$\left(\frac{dV}{dr}\right)_{r_0} = 0. \tag{3.25}$$

Whether this points are stable or unstable has to do with the sign of the second order derivative of V. Hence, if

$$\left(\frac{d^2V}{dr^2}\right)_{r=r_0} > 0 \tag{3.26}$$

the equilibrium point is stable, and if

$$\left(\frac{d^2V}{dr^2}\right)_{r=r_0} < 0 \tag{3.27}$$

the equilibrium point is unstable. Expanding this analysis and deriving (3.24), for the stable case we have

$$\left(\frac{d^2V}{dr^2}\right)_{r=r_0} = F'(r_0) + \frac{3}{r_0}\frac{L^2}{m{r_0}^3} > 0, \tag{3.28}$$

where $F'(r)$ is the F derivative in terms of r. Using (3.25), then the condition of a stable orbit of radius r_0 is expressed as

$$F'(r_0) + \frac{3}{r_0}F(r_0) < 0. \tag{3.29}$$

Note here that circular orbits only exist on attractive force fields and consequently $F(r_0) < 0$, so the condition takes the form

$$\frac{F'(r_0)}{F(r_0)} + \frac{3}{r_0} > 0. \tag{3.30}$$

It is obvious that if we are referring to unstable orbits we have

$$\frac{F'(r_0)}{F(r_0)} + \frac{3}{r_0} < 0. \tag{3.31}$$

These conditions apply to every central force.

In our case,for an oblate spheroid (where $I_{xx} < I_{zz}$), for a force given by the (3.20), and for stable orbits (from (3.30)) we have

$$r > \sqrt{\frac{3}{2}(I_{zz} - I_{xx})}. \tag{3.32}$$

4 Algorithm Description

In this chapter we will present the algorithm we built to compute the orbits of a point mass object of a mass m, which corresponds to a spacecraft, around a rotating rigid body of a mass M, which corresponds to an asteroid. Let the shape of the asteroid for this analysis be an oblate spheroid, as we described it to a previous chapter. That means that the two out of three moments of inertia are equal, therefore the two out of three corresponding principal semi-axes are also equal. Spacecraft's mass m is considered here negligible in relation to asteroid's mass M. We need to note here that, for computational purposes, we consider the gravity constant to be $G = 1$ and the mass of the asteroid to be $M = 1$.

4.1 Main equations

The gravitational potential of the oblate spheroid is given by the equation (3.1) and the equation of motion of the spacecraft in the rotating, body-fixed frame is given by (3.16). The rigid body rotates with an angular velocity vector corresponding to (2.53) in the rotating frame. In other words, the main idea is to enter (2.53) and (3.1) in (3.16), that will eventually give as a solution the position (x, y and z components) of the spacecraft in space. It is clear that every component of position or velocity (and angular velocity) is expressed in the body-fixed frame. Thus, the initial conditions given to integrate the equation of motion are also needed to be expressed in the body-fixed frame.

Nevertheless, for us to know at any moment or to manually set the position of the spacecraft in space, we need its inertial coordinates. The problem is easily solved using the quaternion method we described above for the transformation from the one frame to the other and vice versa. We also need the inertial frame coordinates to calculate the orbital elements of the spacecraft. Hence, although the equations of motion refer to the body-fixed frame, we need information from both the body-fixed and the inertial frame, that we easily get using the transformation formula, with the rotation matrix given by (2.87). Let us note here that the set of the four quaternions represent the orientation of the body-fixed frame in respect to the inertial. For this to be possible, the use of the quaternion differential equations (2.78) is necessary.

The programming language used for this topic is *C++*. The Bulirsch–Stoer method, implemented with the class *ODESBS*, is used for the numerical integration of the (3.16) along with (2.78). The differential equations to solve are seven in total, with the three of them being second order. It is important to understand that the three coordinates in the rotating frame (x, y and z) and the four quaternion components, need to be known for every time step. Moreover, the auxiliary code *restroe41* is used for the orbital elements to be calculated. The program returns three files containing the coordinates of the inertial frame, the coordinates of the rotating frame and the orbital elements respectfully. With these information we visualize an orbit to get the complete picture. In the Appendix B we give the module *dSystemElpsdRotPrec23.cpp* that contains the equations of motion. In the Appendix A we give the *main1c.cpp* function for a run of an individual orbit.

4.2 Initial Conditions and Parameters

As mentioned before, the differential equations need to be solved are seven; three second order differential equations (equations of motion) and four first order differential equations (for the quaternions). Thus, ten initial conditions are needed for this system; three position components, three velocity components and the four initial quaternions. Let us mention again here that the initial quaternions represent the initial orientation of the rotating frame. One chooses the initial position and velocity of the spacecraft in space (inertial frame) and the initial orientation of the rotating frame and eventually gets the evolution of the orbit in time.

We choose initial conditions of position and velocity in the inertial frame for the particle of the form

$$\mathbf{R} = (r_0,\, 0,\, 0) \qquad \mathbf{V} = (0,\, \sqrt{-r_0 F(r_0)},\, 0), \tag{4.1}$$

represented in the Figure 22. The velocity is the one needed for an object of negligible mass to carry out a circular orbit, expressed above by the equation (3.23). Then, we choose that the initial orientation of the rotating frame will be a roll, i.e, a rotation of an angle θ_x around X axis:

$$roll = \theta_{x0} \qquad pitch = 0 \qquad yaw = 0. \tag{4.2}$$

The angle takes values from 0 to $\pi/2$. Using (2.85), one can calculate the corresponding quaternions. Another parameter we need to set is the angular frequency B of the angular velocity's vector $\boldsymbol{\omega}$ (see (2.53)). It takes values from 0 to 1.

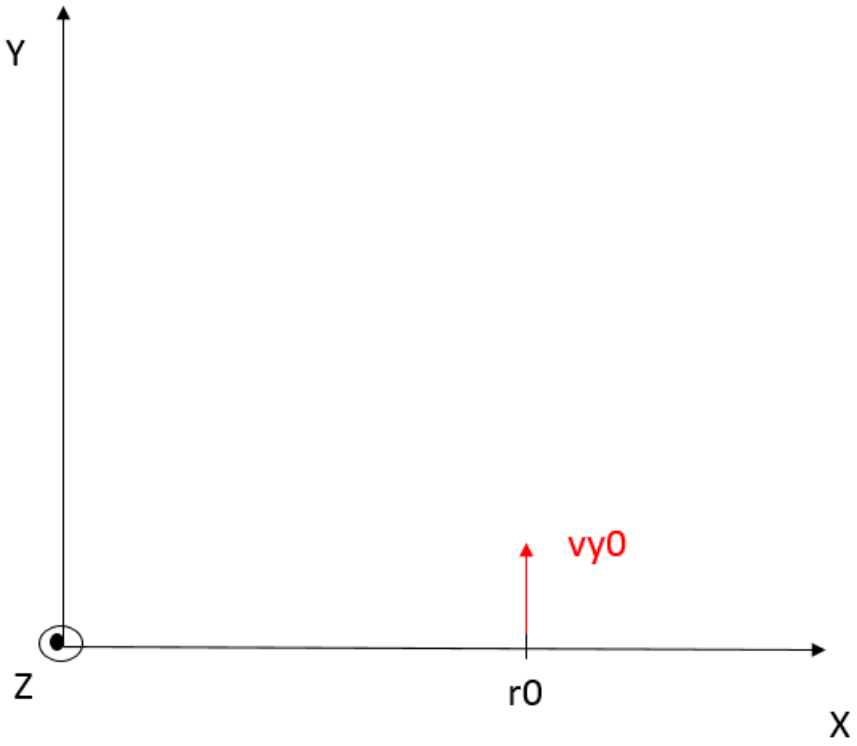


Figure 22: Initial Conditions in the inertial frame.

4.3 Unit Normalization

In a previous section we saw the lengths a, b and c as the principal semi-axes of an ellipsoid. Here, since we are referring to an oblate spheroid, it applies that $a = b$ and $c < a$, expressed lets say in m. This would give us big numbers as a solution, difficult to process. That's why we follow a normalization procedure by applying the following scaling

$$x' = x/a,\ y' = y/a,\ z' = z/a,\ t' = \omega t \tag{4.3}$$

and replacing the semi-axes with the scaled ones

$$a' = 1,\ b' = b/a,\ c' = c/a \tag{4.4}$$

in equations (3.16). Of course, time derivatives refer now to time t'. As discussed before, the magnitude of the angular velocity vector is constant (see (2.53)). Here, in our calculations we consider that

$$\| \omega' \| = \sqrt{A^2 + {\omega_z}^2} = 1. \tag{4.5}$$

For the characteristics of the rigid body (asteroid), we consider

$$a' = b' = 1.0, \quad c' = 0.8 \tag{4.6}$$

for the normalized semi-axes, and

$$\delta = 0.8. \tag{4.7}$$

The equations of motion can now be written as

$$\begin{aligned}
&\ddot{x'} + \dot{\omega}'_y z' - \dot{\omega}'_z y' + 2{\omega_y}'\dot{z'} - 2{\omega_z}'\dot{y'} + {\omega_x}'{\omega_y}'y' - {\omega_y}'^2 x' - {\omega_z}'^2 x' + {\omega_z}'{\omega_x}'z' + \delta\frac{\partial U}{\partial x'} = 0\\
&\ddot{y'} + \dot{\omega}'_z x' - \dot{\omega}'_x z' + 2{\omega_z}'\dot{x'} - 2{\omega_x}'\dot{z'} + {\omega_y}'{\omega_z}'z' - {\omega_z}'^2 y' - {\omega_x}'^2 y' + {\omega_x}'{\omega_y}'x' + \delta\frac{\partial U}{\partial y'} = 0\\
&\ddot{z'} + \dot{\omega}'_x y' - \dot{\omega}'_y x' + 2{\omega_x}'\dot{y'} - 2{\omega_y}'\dot{x'} + {\omega_x}'{\omega_z}'x' - {\omega_x}'^2 z' - {\omega_y}'^2 z' + {\omega_y}'{\omega_z}'y' + \delta\frac{\partial U}{\partial z'} = 0.
\end{aligned} \tag{4.8}$$

where $U = U(x', y', z')$, ${\omega_i}' = \frac{\omega_i}{\|\omega\|}$ (with $i = x,\ y,\ z$) and

$$\delta = \frac{\mu}{\omega^2 a^3}. \tag{4.9}$$

The parameter δ is special for every asteroid and for its calculation one has to substitute the real values of μ, ω and a.

In the following, we omit the primes off the symbols

$$\begin{aligned}
&\ddot{x} + \dot{\omega}_y z - \dot{\omega}_z y + 2\omega_y \dot{z} - 2\omega_z \dot{y} + \omega_x\omega_y y - {\omega_y}^2 x - {\omega_z}^2 x + \omega_z\omega_x z + \delta\frac{\partial U}{\partial x} = 0\\
&\ddot{y} + \dot{\omega}_z x - \dot{\omega}_x z + 2\omega_z \dot{x} - 2\omega_x \dot{z} + \omega_y\omega_z z - {\omega_z}^2 y - {\omega_x}^2 y + \omega_x\omega_y x + \delta\frac{\partial U}{\partial y} = 0\\
&\ddot{z} + \dot{\omega}_x y - \dot{\omega}_y x + 2\omega_x \dot{y} - 2\omega_y \dot{x} + \omega_x\omega_z x - {\omega_x}^2 z - {\omega_y}^2 z + \omega_y\omega_z y + \delta\frac{\partial U}{\partial z} = 0.
\end{aligned} \tag{4.10}$$

Furthermore, the same scaling should be applied in the force we discussed in horizontal stability, so it would give us

$$F = \delta F' \tag{4.11}$$

where

$$F' = -\frac{1}{{r'}^2} + \frac{3(I'_{xx} - I'_{zz})}{2{r'}^4}, \tag{4.12}$$

or, if we omit the primes here too, the force is given by

$$F = \delta\Big(-\frac{1}{r^2} + \frac{3(I_{xx} - I_{zz})}{2r^4}\Big) \tag{4.13}$$

In that way, supposing the equations of motion have a solution x', y' and z' of the position of the spacecraft in the rotating frame, in order to get the real distance, we need to multiply each coordinate with the semi-axis a. For the real time value we need to divide t' with ω.

5 Results

In this chapter we display our results. With the help of the program *levelmap2.exe*, we construct coloured maps for the maximum value of the orbit's semi-major axis, the eccentricity and the inclination. We make a grid of 51×51 in the $x - y$ plane, where $x \in (0, 1.57)$ and $y \in (0, 1)$, creating in total $2,601$ initial conditions. The x values represent the initial θ_x and the y values represent the initial B (angular frequency of the vector $\boldsymbol{\omega}$). For every pair of initial conditions, we integrate the orbit and compute the orbital elements semi-major axis (a), eccentricity (e) and inclination (i) at each step. Finally, we output the a_{max}, e_{max} and i_{max} for each orbit. Each magnitude has its own map. The colours of the map correspond to the value of a_{max}, e_{max} and i_{max}, respectively. In that way we can get the full picture of the region of initial conditions, where the orbit remains stable. Small values generally correspond to stable orbits and large values correspond to unstable orbits, that may lead to collision of the spacecraft with the asteroid or escape.

The dark blue and green coloured regions represent stable orbits, that stay close to the asteroid, the yellow represent orbits that move away from the asteroid, with some of them stay bounded and others lead to escape, and the white for the a_{max} and bright yellow for e_{max} and i_{max} represent collision.

5.1 Dynamical Maps for different pairs of initial radius and asteroid shape

Here, we examine different values of the initial radius of the orbit (r_0) an the size of the asteroid (the parameter c) and see how the orbits are being affected from the changes of these values.

5.1.1 $\mathbf{r_0 = 1.5}$ and $\mathbf{c = 0.7}$

We set for the initial radius of the orbit $r_0 = 1.5$ and for the vertical semi-axis of the asteroid $c = 0.7$ and consider these constant, when constructing the maps for all the pairs of the initial conditions of θ_x and B. Figure 23 represents the dynamical map for a_{max}, figure 24 represents the dynamical map for e_{max} and figure 25 for i_{max}.

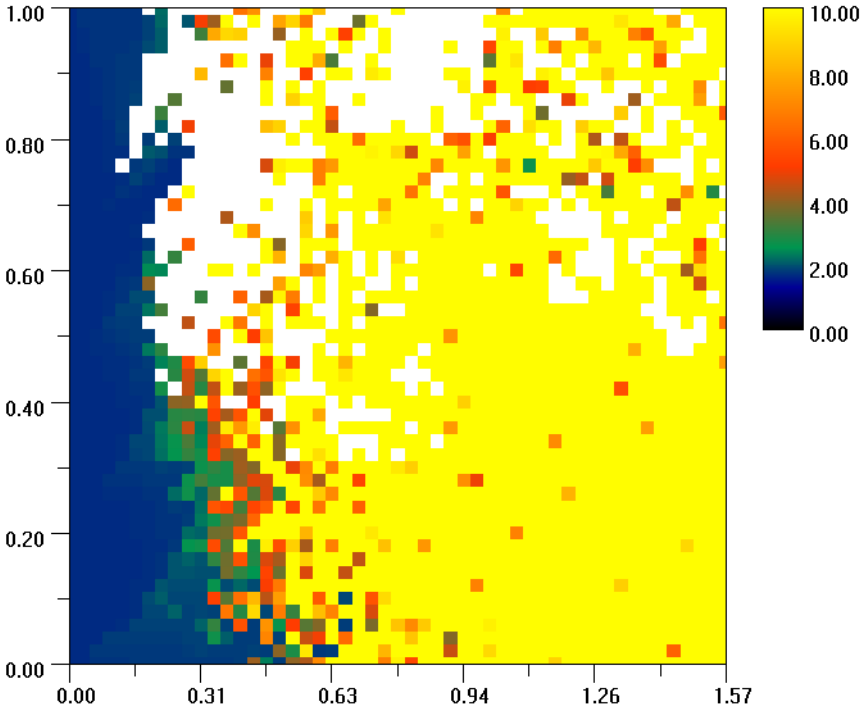


Figure 23: Values of a_{max} for a grid (θ_x, B) and for $r_0 = 1.5$ and $c = 0.7$.

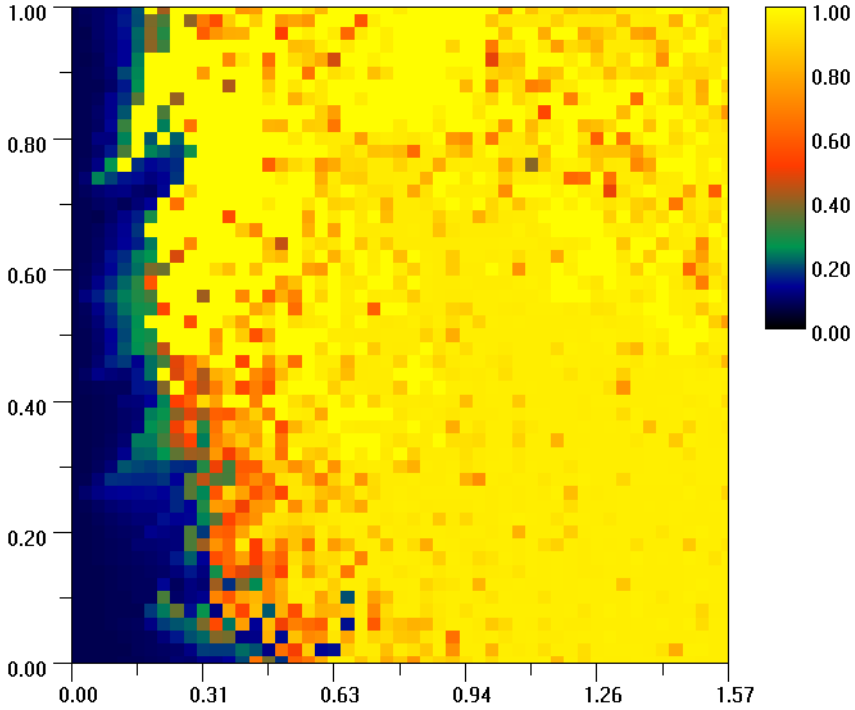


Figure 24: Values of e_{max} for a grid (θ_x, B) and for $r_0 = 1.5$ and $c = 0.7$.

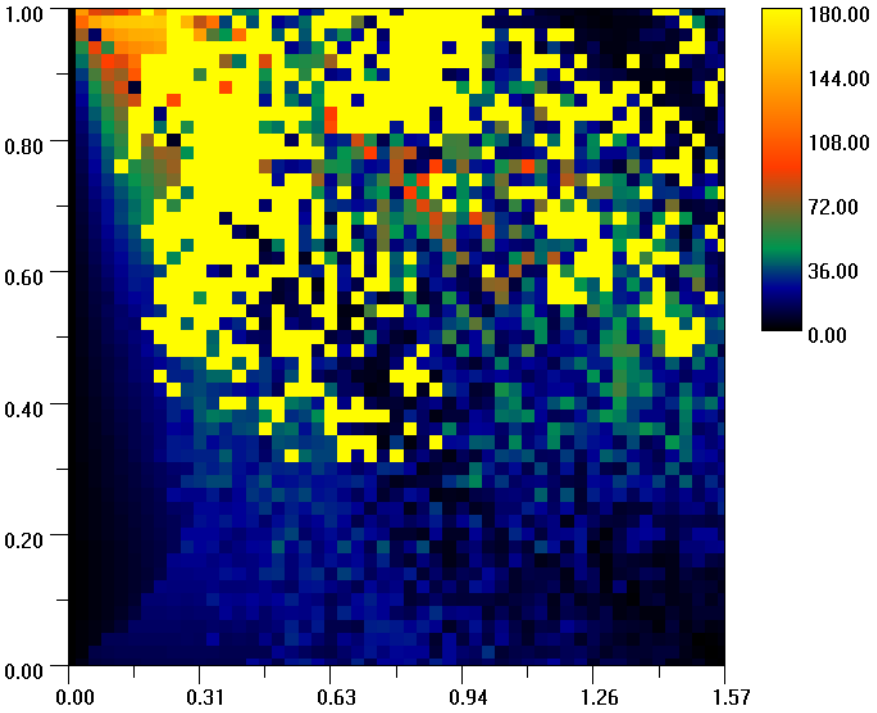


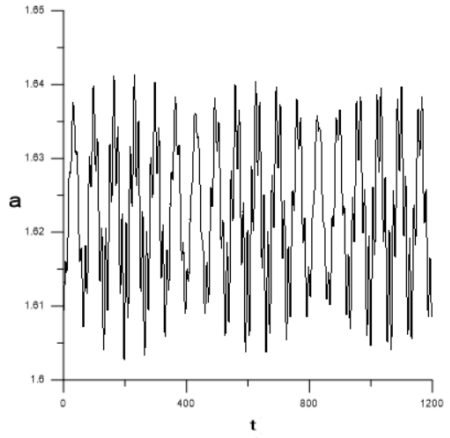
Figure 25: Values of i_{max} for a grid (θ_x, B) and for $r_0 = 1.5$ and $c = 0.7$.

Stable, bounded orbits A few examples of these orbits are given bellow.

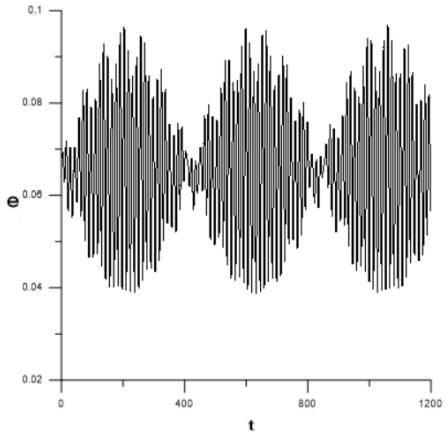
For an orbit with initial conditions

$$\theta_{x0} = 0.0628 \qquad B_0 = 0.46$$

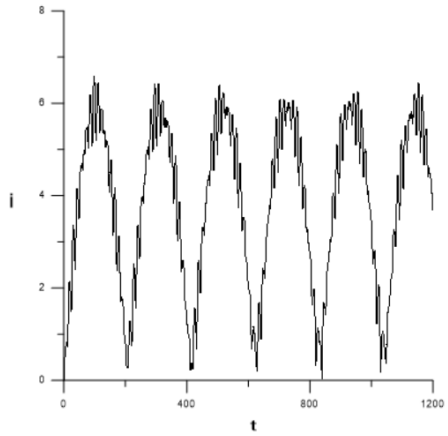
we take the graph of the semi-major axis, the eccentricity, the inclination and the radius of the orbit over time.



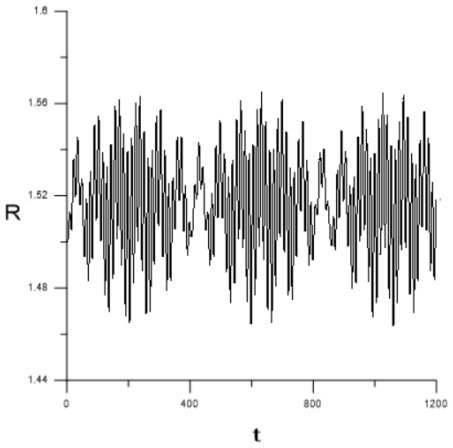
(a) a(t)



(b) e(t)



(c) i(t)



(d) R(t)

Figure 26: $\theta_{x0} = 0.0628 \qquad B_0 = 0.46$

The projection of the orbit in the XY plane of the inertial frame is represented in the figure below with black and with the green circle representing the asteroid.

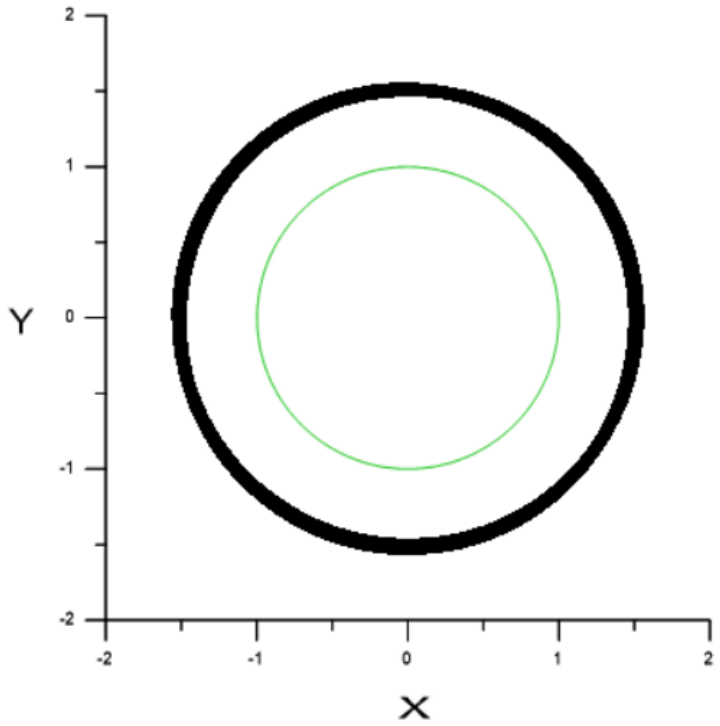


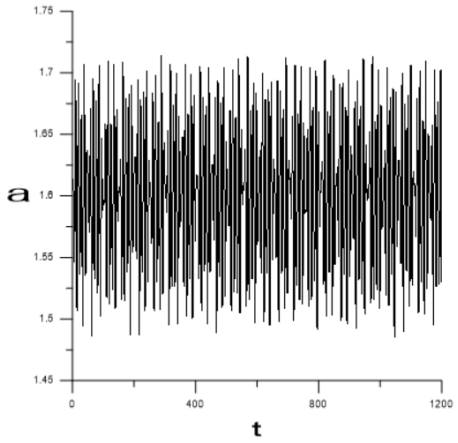
Figure 27: Orbit on the XY plane with $\theta_{x0} = 0.0628 \qquad B_0 = 0.46$.

As we can see, the orbit stays bounded over time and the magnitudes of semi-major axis, eccentricity, inclination and orbit's radius oscillate.

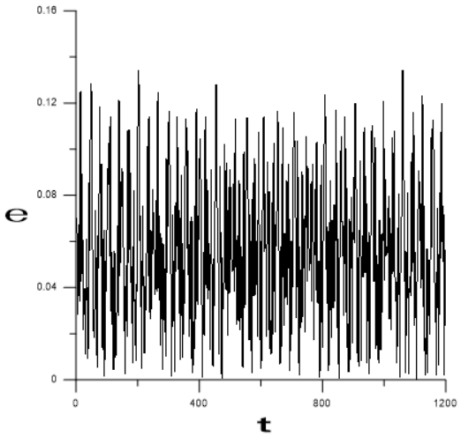
Let another example be an orbit with initial conditions

$$\theta_{x0} = 0.1256 \qquad B_0 = 0.96$$

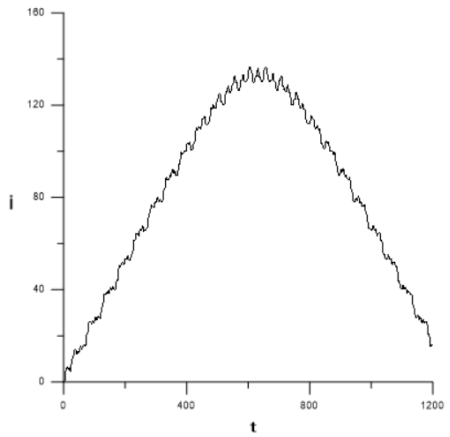
we have



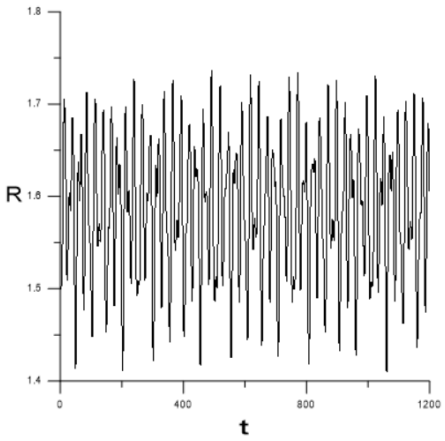
(a) a(t)



(b) e(t)



(c) i(t)



(d) R(t)

Figure 28: $\theta_{x0} = 0.1256 \qquad B_0 = 0.96.$

Here, even though the orbit remains bounded, the inclination increases up to 140 degrees. In this case, the spacecraft doesn't escape the asteroid, its orbit remains almost circular, but the orbit's inclination takes great values. In the projection on the XY plane, one can see that the spacecraft appears to cross the body, but in reality it passes from above.

The inclination seems to being affected by the initial B, since the value here is almost double compared with the previous one. As we can see from the corresponding map in figure 25, for small B, the inclination gets only small values.

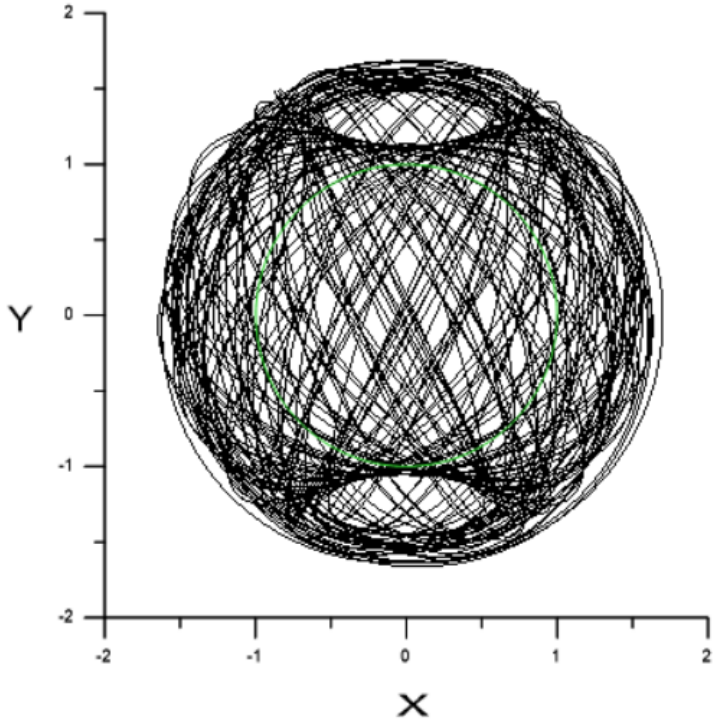
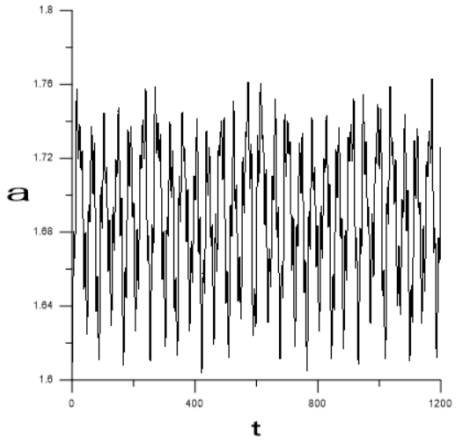


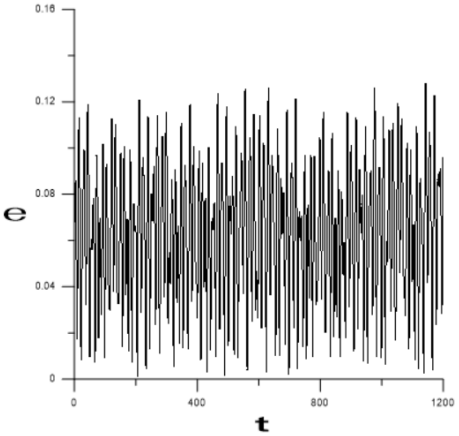
Figure 29: Orbit on the XY plane with $\theta_{x0} = 0.1256 \qquad B_0 = 0.96$.

Let us consider another orbit with a relatively small initial B, so that

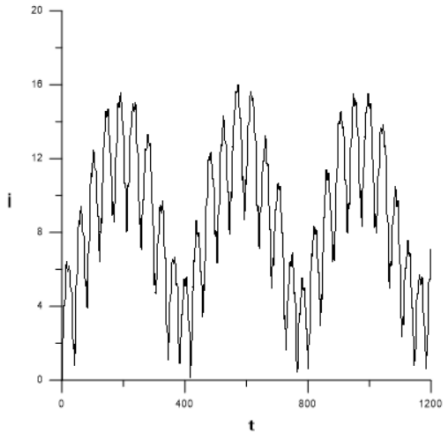
$$\theta_{x0} = 0.2826 \qquad B_0 = 0.24.$$



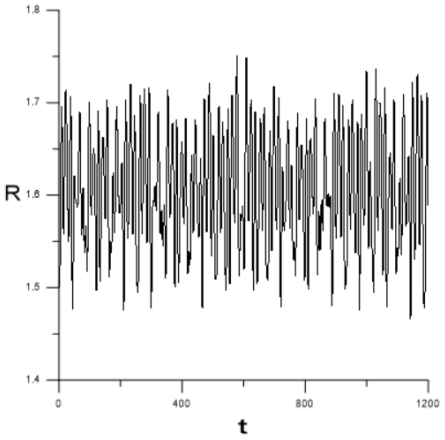
(a) a(t)



(b) e(t)



(c) i(t)



(d) R(t)

Figure 30: $\theta_{x0} = 0.2826 \qquad B_0 = 0.24$

Note here, that a, e, i, and R oscillate, with the inclination having a maximum near 16°. On the XY plane

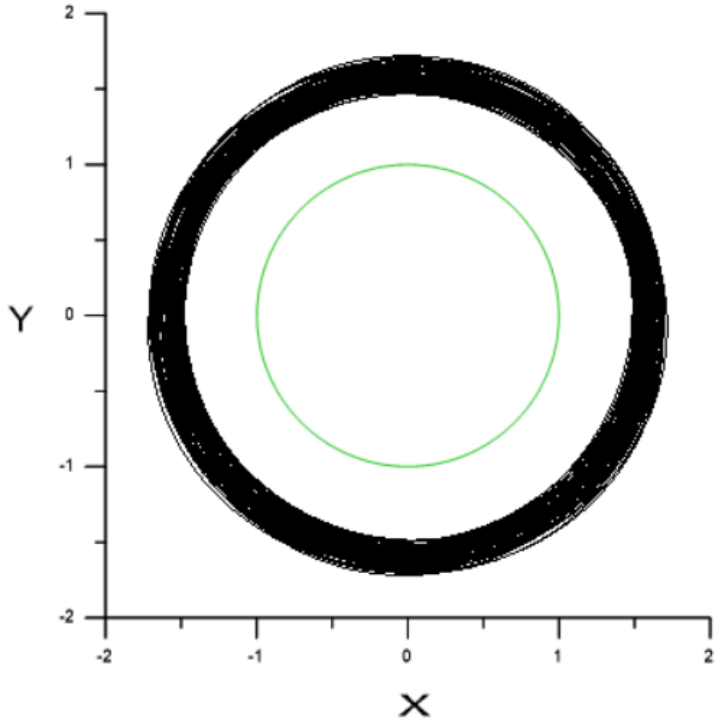


Figure 31: Orbit on the XY plane with $\theta_{x0} = 0.2826 \qquad B_0 = 0.24$.

Collision We consider an example of an orbit from the region representing collision (white for a_{max}-figure 23), with initial conditions

$$\theta_{x0} = 0.4082 \qquad B_0 = 0.78.$$

The collision happens, approximately, when the radius takes a value smaller than 1 (as a normalized unit). Plotting the radius of that orbit over time, one can see in the figure 32 that the radius becomes indeed smaller than 1, which means that the spacecraft collided with the asteroid.

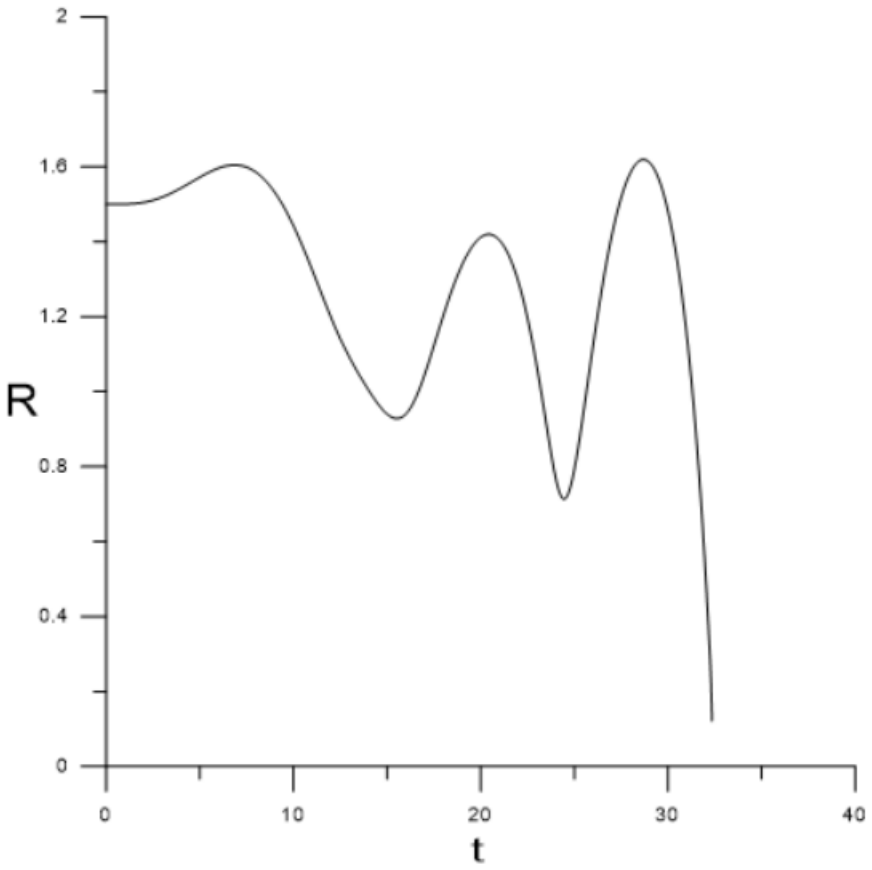


Figure 32: $R - t$ with $\theta_{x0} = 0.4082 \qquad B_0 = 0.78$ (Collision).

Escape Choosing an orbit from the yellow region of a_{max} (figure 23), i.e. an unstable orbit with initial conditions

$$\theta_{x0} = 1.0676 \qquad B_0 = 0.22.$$

and plotting its radius, we have the evolution given in figure 33

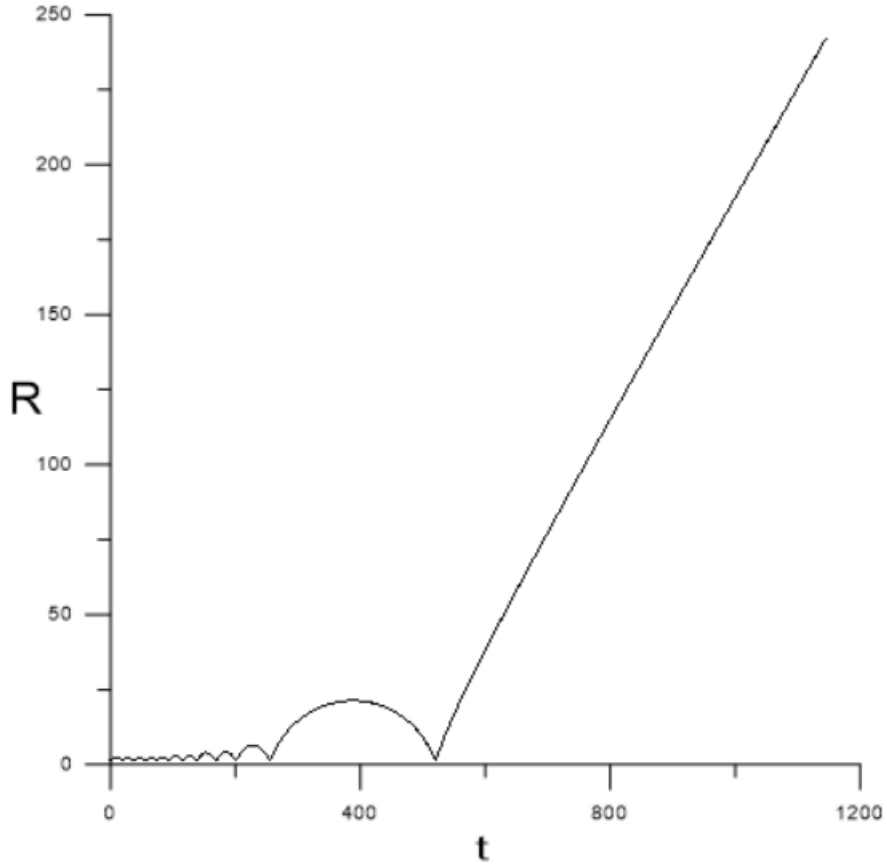


Figure 33: $R - t$ with $\theta_{x0} = 1.0676 \qquad B_0 = 0.22$ (Escape).

One can see that the radius increases up to 250 normalized units. In this case, the orbit is unbounded and the spacecraft has escaped.

5.1.2 $\mathbf{r_0 = 3.0}$ and $\mathbf{c = 0.7}$

Let us now double the initial radius of the orbit, which leads us to new dynamical maps.

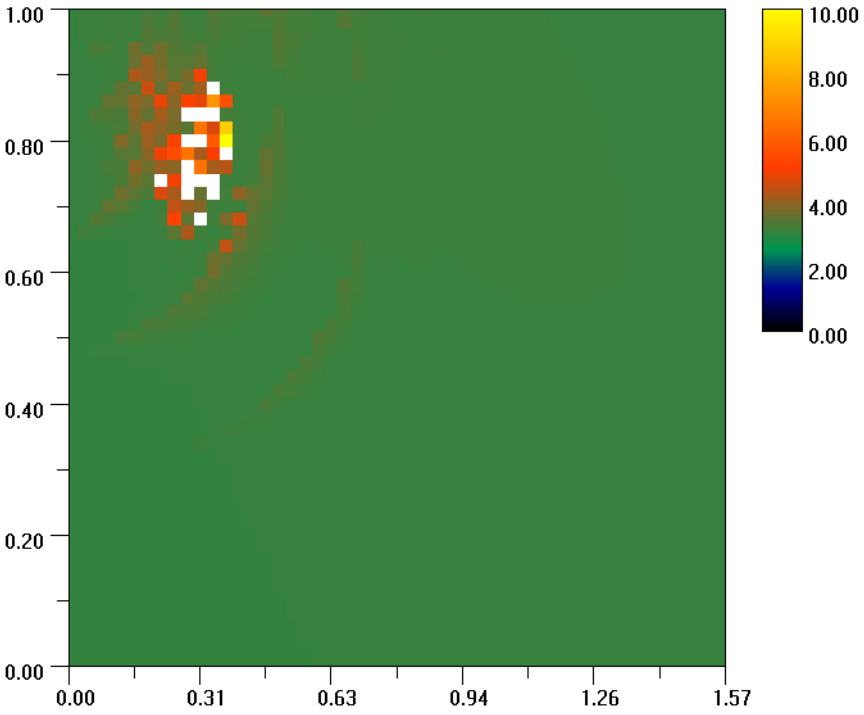


Figure 34: Values of a_{max} for a grid (θ_x, B) and for $r_0 = 3.0$ and $c = 0.7$.

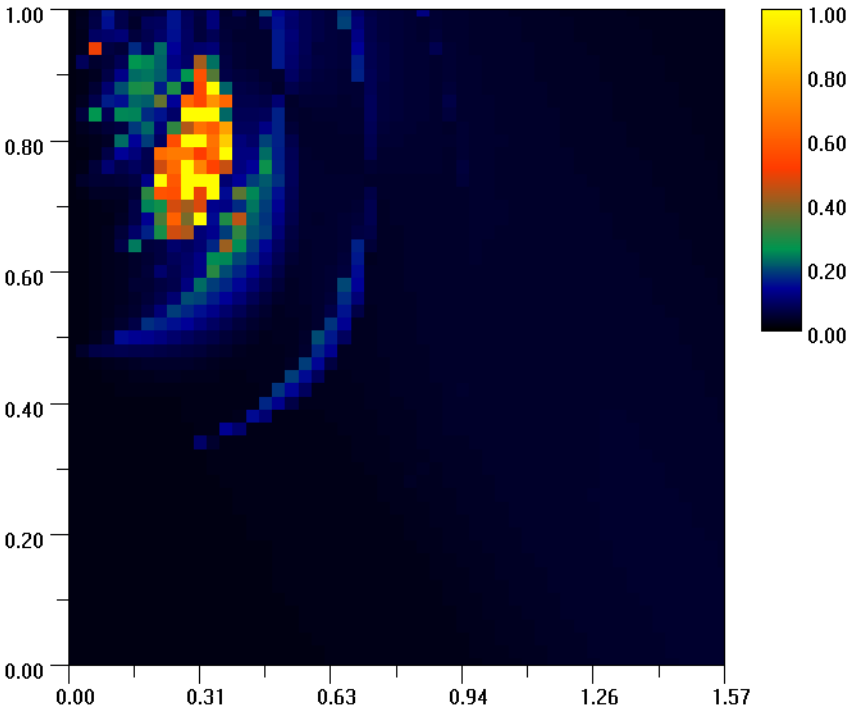


Figure 35: Values of e_{max} for a grid (θ_x, B) and for $r_0 = 3.0$ and $c = 0.7$.

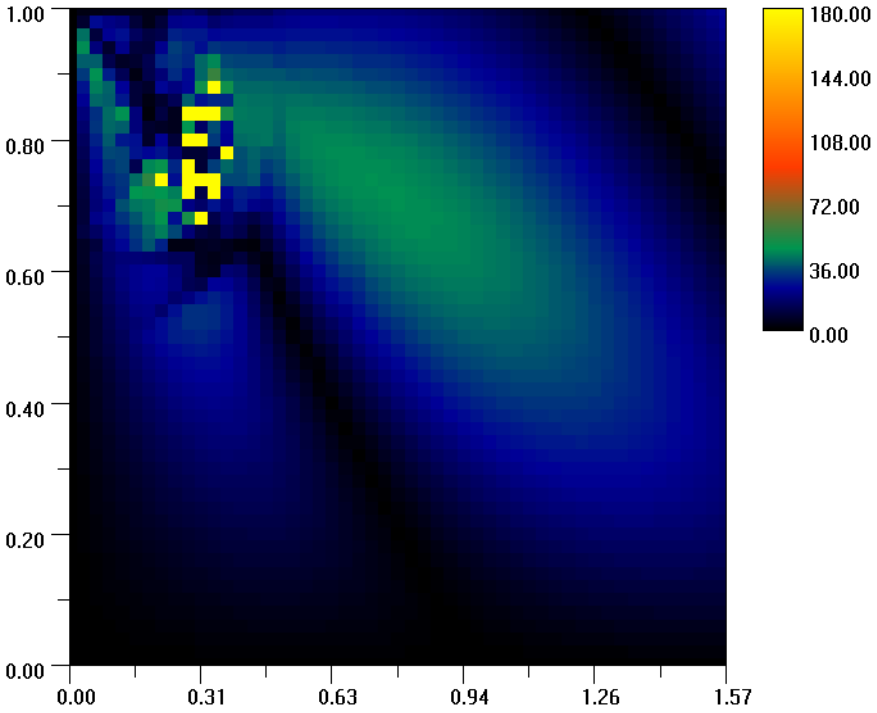


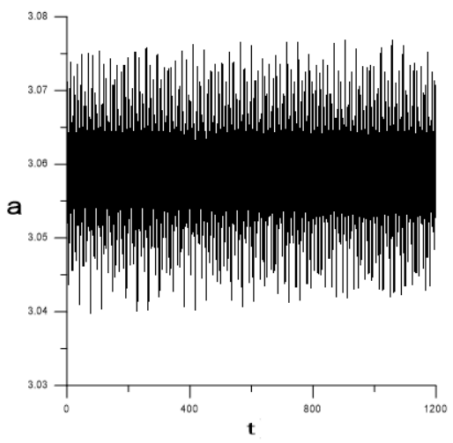
Figure 36: Values of i_{max} for a grid (θ_x, B) and for $r_0 = 3.0$ and $c = 0.7$.

Notice, here, that by increasing the initial distance of the spacecraft from the asteroid, the region of instability decreases significantly. Due to greater distance, the oblate shape of the asteroid becomes more negligible. Taking into account the potential from the equation (3.1), one can understand that as the radius increases, the second and third term of the potential decrease significantly and considered also negligible.

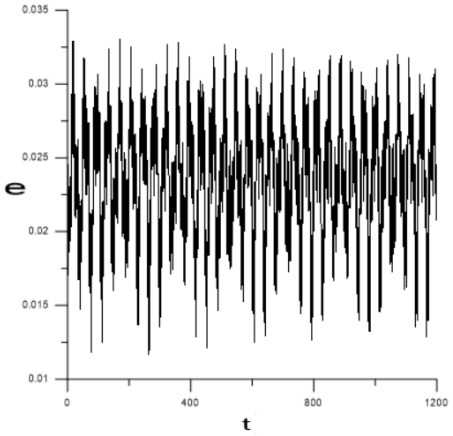
Stable, bounded orbits For an orbit with initial conditions

$$\theta_{x0} = 0.942 \qquad B_0 = 0.28$$

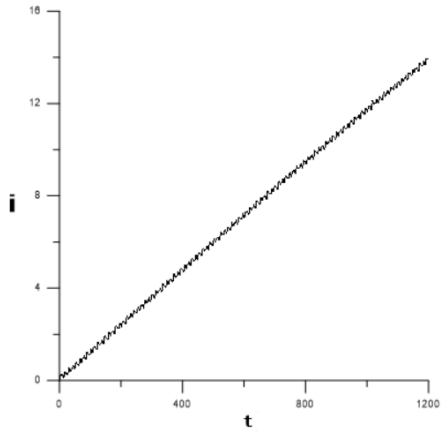
we have



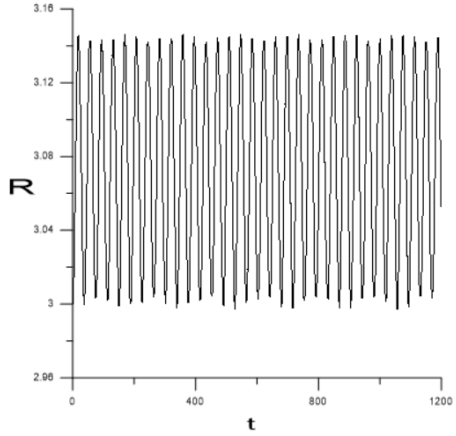
(a) a(t)



(b) e(t)



(c) i(t)



(d) R(t)

Figure 37: $\theta_{x0} = 0.942 \qquad B_0 = 0.28.$

The orbit on the XY plane is given in figure 39 and on the XZ plane is given in figutr **??**

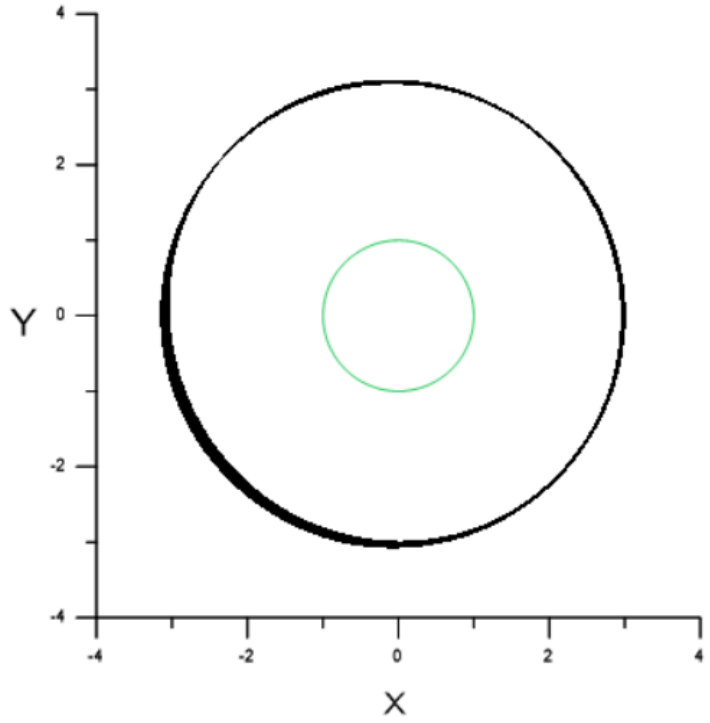


Figure 38: Orbit on the XY plane with $\theta_{x0} = 0.942 \qquad B_0 = 0.28$.

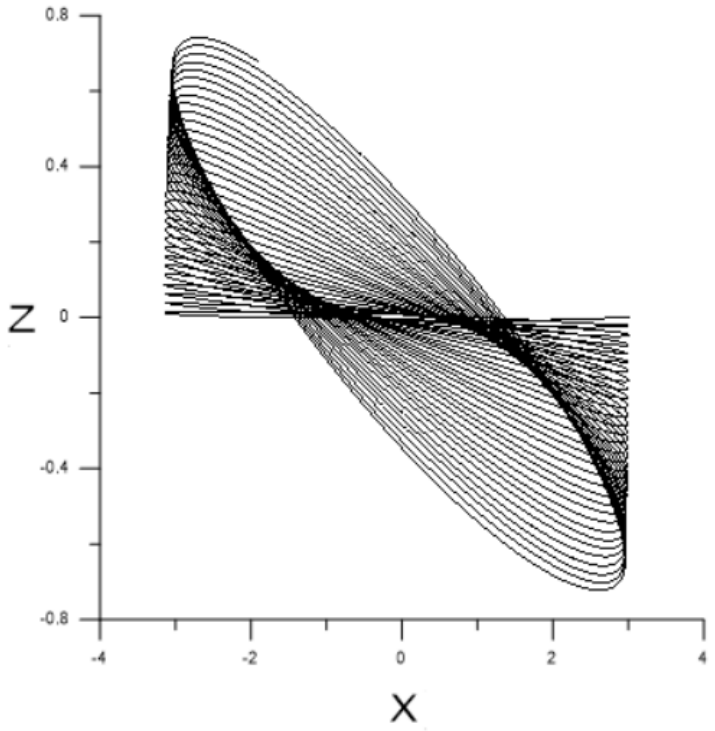


Figure 39: Orbit on the XZ plane with $\theta_{x0} = 0.942 \qquad B_0 = 0.28$.

The inclination appears to continuously increasing, while the other magnitudes oscillate and the orbit is bounded. In fact, the inclination doesn't continuously increase and we can ascertain that by running the orbit for a longer period of time, as seen in the figure below.

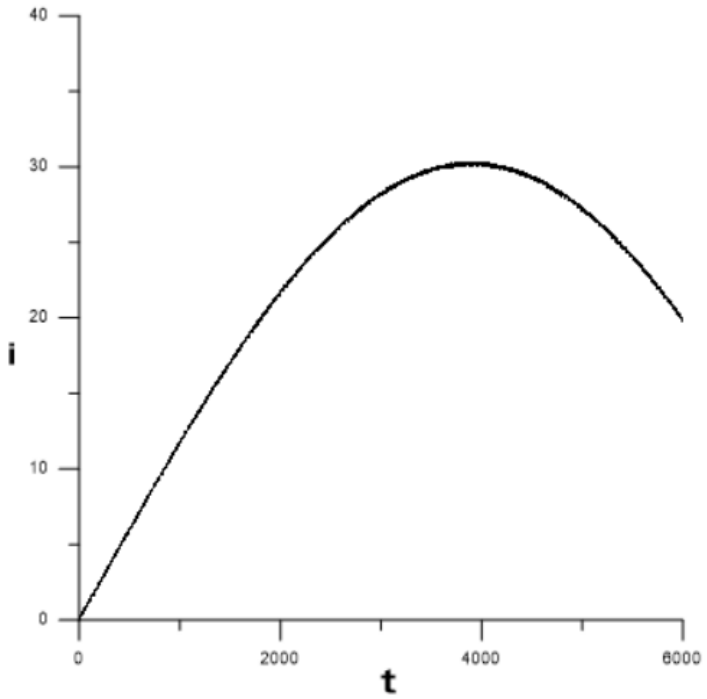
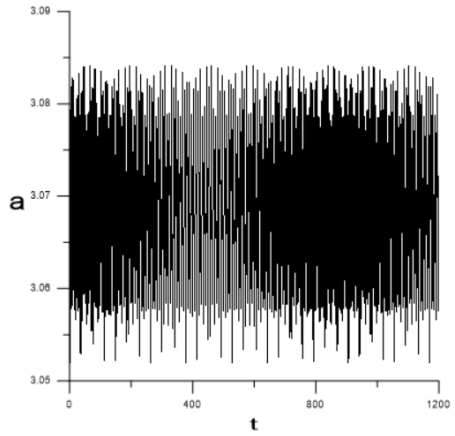


Figure 40: i(t) for a longer period of time with $\theta_{x0} = 0.942 \qquad B_0 = 0.28$.

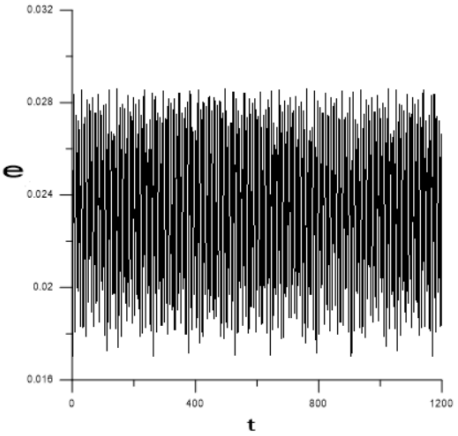
For an orbit with initial conditions

$$\theta_{x0} = 1.3188 \qquad B_0 = 0.98$$

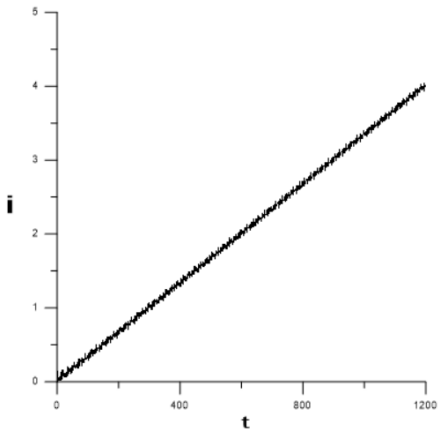
we have the evolution given in figure 41 and the orbit in figure 42



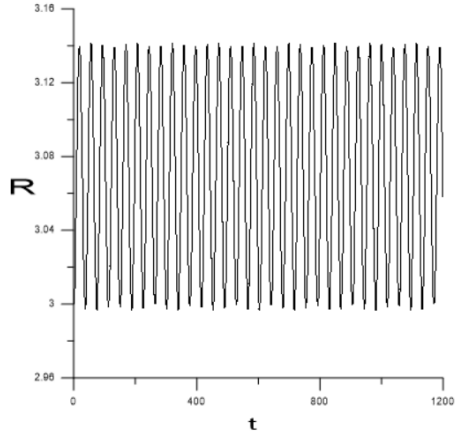
(a) a(t)



(b) e(t)



(c) i(t)



(d) R(t)

Figure 41: $\theta_{x0} = 1.3188 \qquad B_0 = 0.98.$

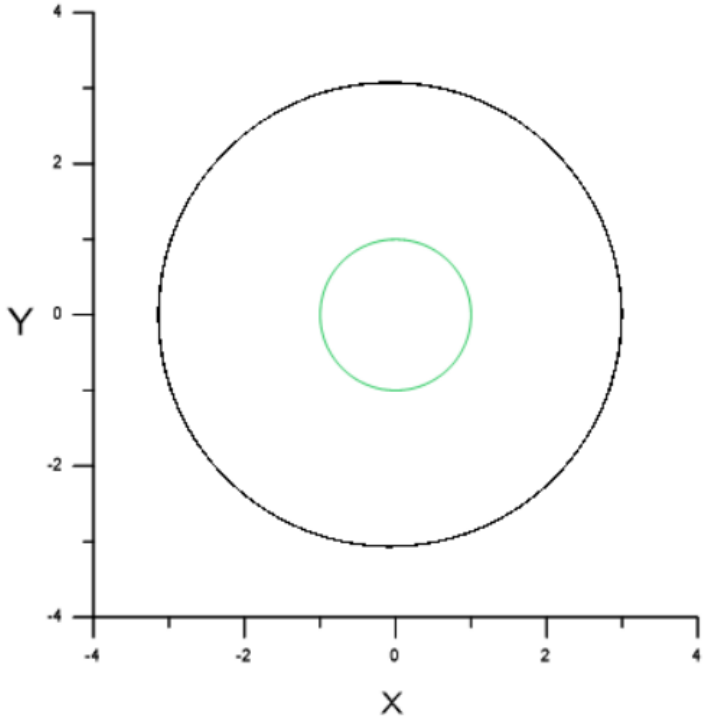
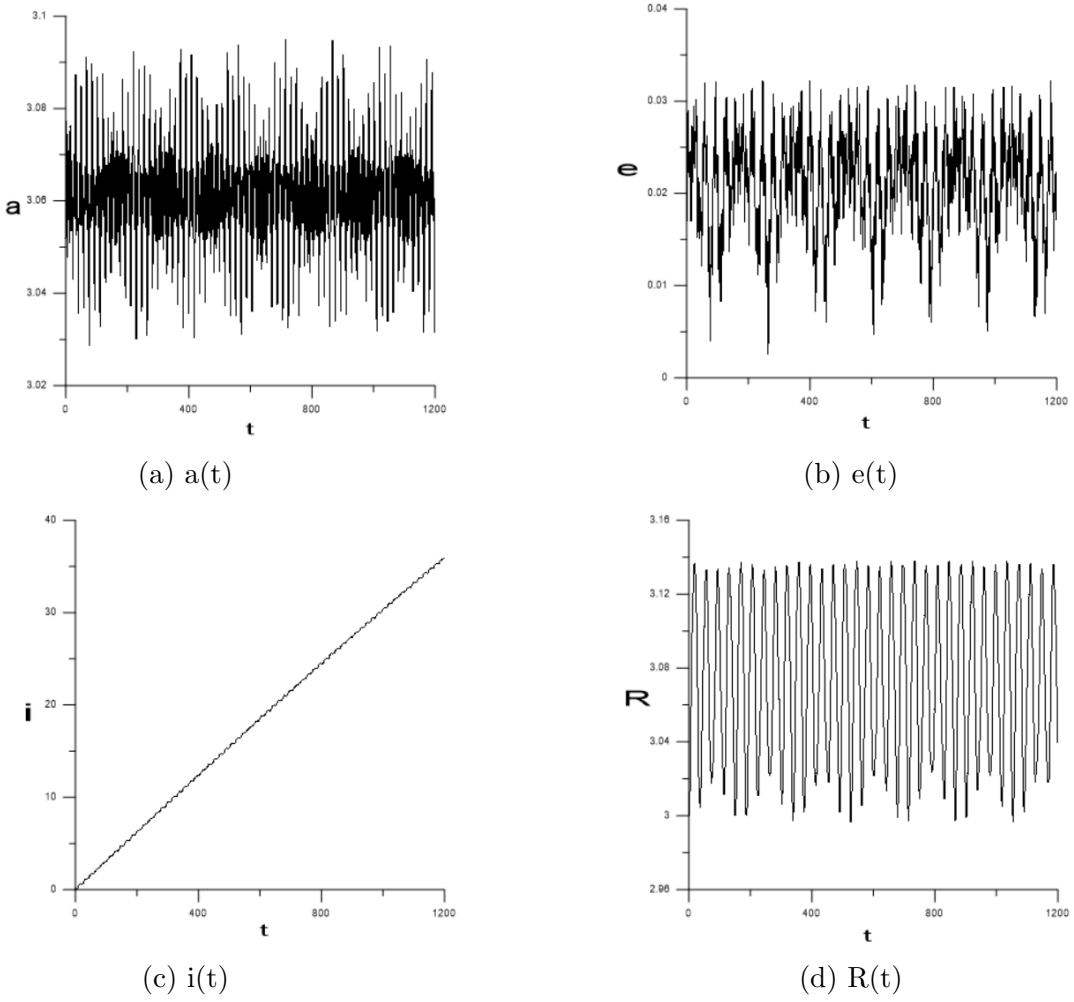


Figure 42: Orbit on the XY plane with $\theta_{x0} = 1.3188 \qquad B_0 = 0.98$.

For an orbit with initial conditions

$$\theta_{x0} = 0.7536 \qquad B_0 = 0.58$$

we have the evolution given in figure 43 and the orbit in figure 44 (XY plane) and in figure 45 (XZ plane)



(a) a(t)

(b) e(t)

(c) i(t)

(d) R(t)

Figure 43: $\theta_{x0} = 0.7536 \qquad B_0 = 0.58$.

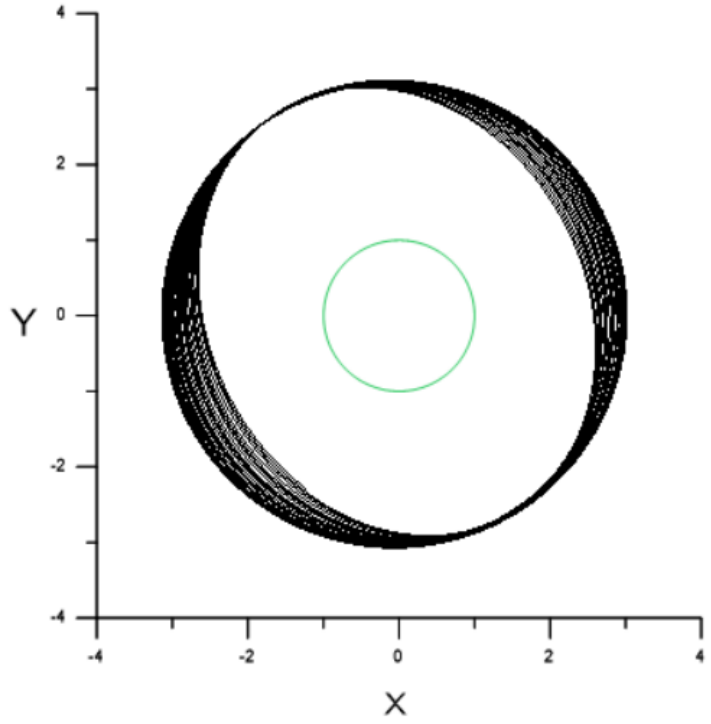


Figure 44: Orbit on the XY plane with $\theta_{x0} = 0.7536 \qquad B_0 = 0.58$.

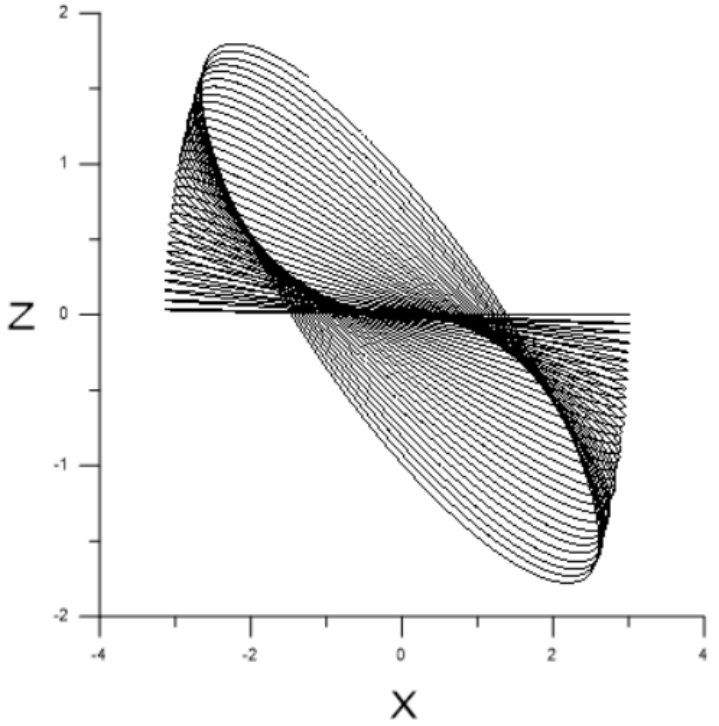


Figure 45: Orbit on the XZ plane with $\theta_{x0} = 0.7536 \qquad B_0 = 0.58$.

Here, the inclination takes bigger values, since the initial conditions correspond to the green area of the i_{max} map in figure 36.

Unstable orbits For an orbit of collision area with initial conditions

$$\theta_{x0} = 0.314 \qquad B_0 = 0.84$$

we have

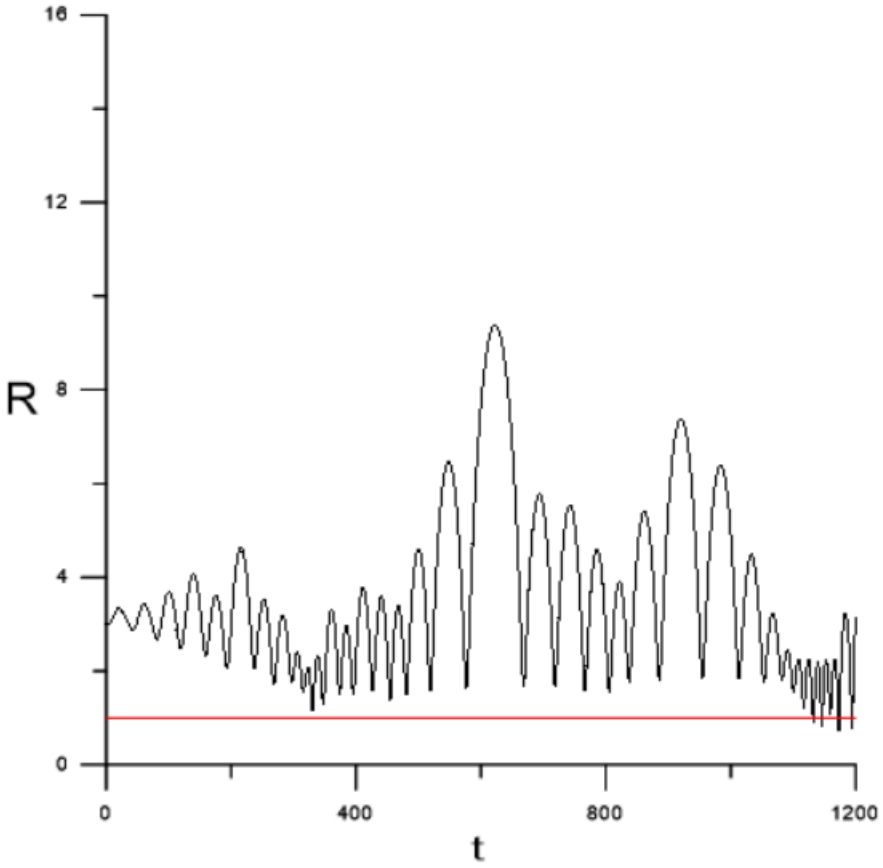


Figure 46: R-t with $\theta_{x0} = 0.314 \qquad B_0 = 0.84$ (Collision).

Here, we can see (figure 46) that the radius value goes below 1, but becomes in general quite large. In this case, the spacecraft moves away from the asteroid but in a bounded orbit, and eventually they collide.

A similar form of time evolution follows the radius of an orbit with initial conditions

$$\theta_{x0} = 0.3768 \qquad B_0 = 0.8,$$

as we can see here

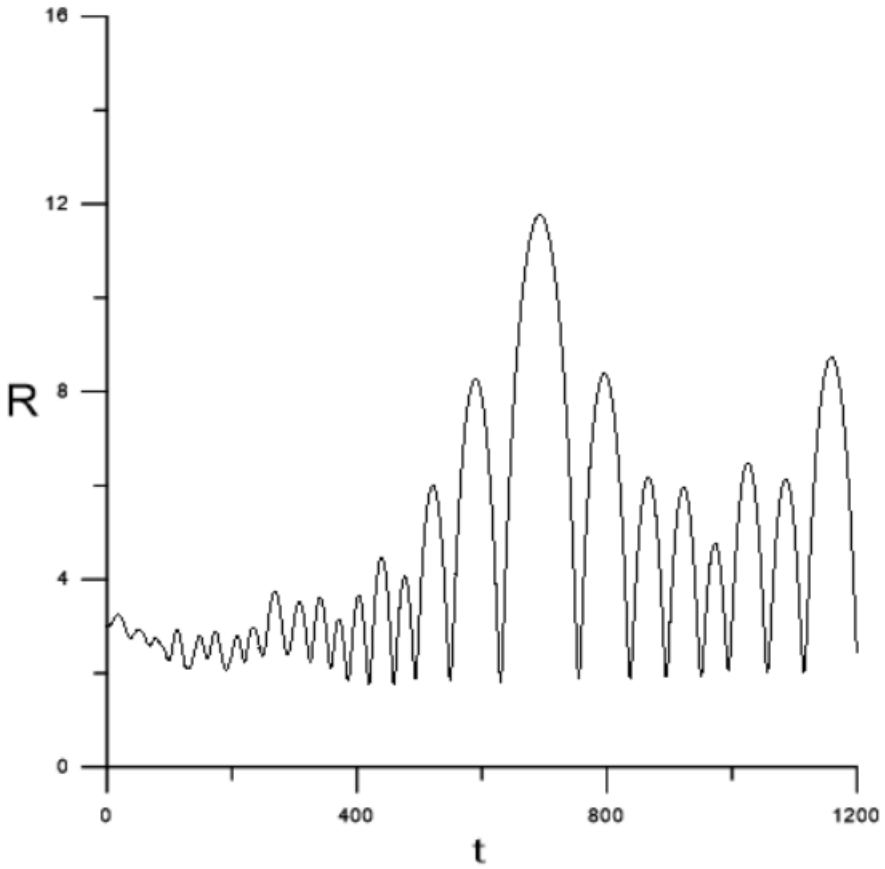


Figure 47: R-t with $\theta_{x0} = 0.3768 \qquad B_0 = 0.8$.

Here, the orbit is bounded, although the radius takes large values, and the two bodies don't collide. In this case, there is a sort of instability, which we can understand better by seen the plot of the semi-major axis (see figure 48).

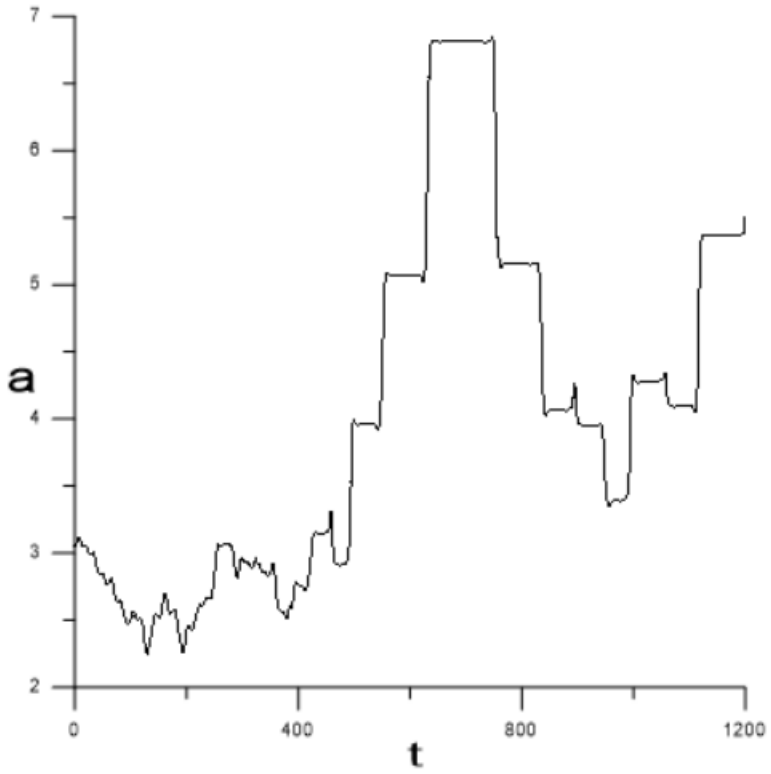


Figure 48: a-t with $\theta_{x0} = 0.3768 \qquad B_0 = 0.8$.

It is clear that the semi-major axis doesn't oscillates around small values, but follows an irregular form and increases significantly.

5.1.3 $\mathbf{r_0 = 1.5}$ and $\mathbf{c = 0.8}$

We return to the initial radius of the first case in the section 5.1.1 and we increase the semi-axis c of the oblate spheroid shape of the asteroid, leading to new dynamical maps seen in figures 49, 50 and 51.

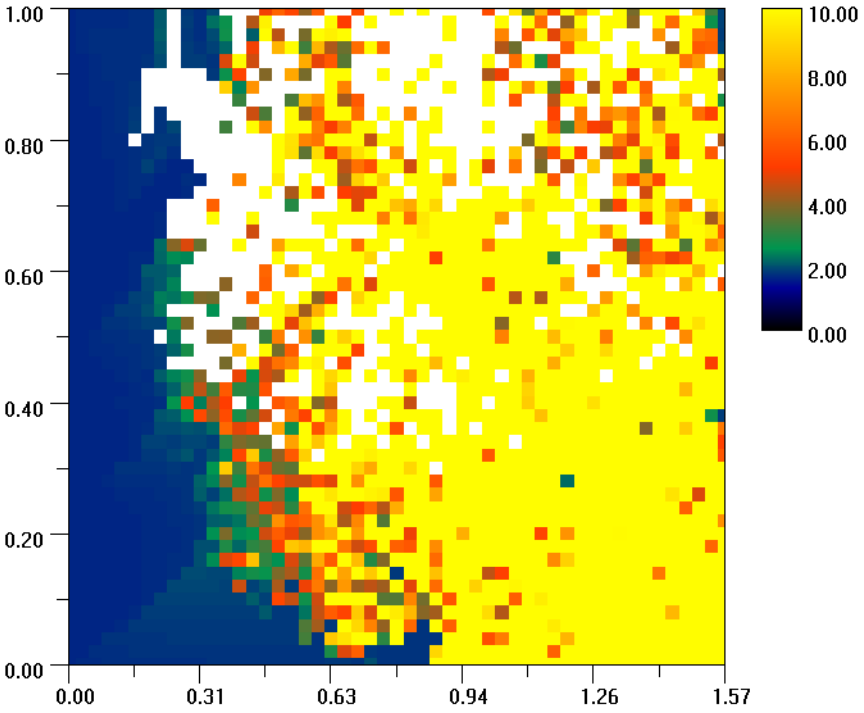


Figure 49: Values of a_{max} of a grid (θ_x, B) for $r_0 = 1.5$ and $c = 0.8$.

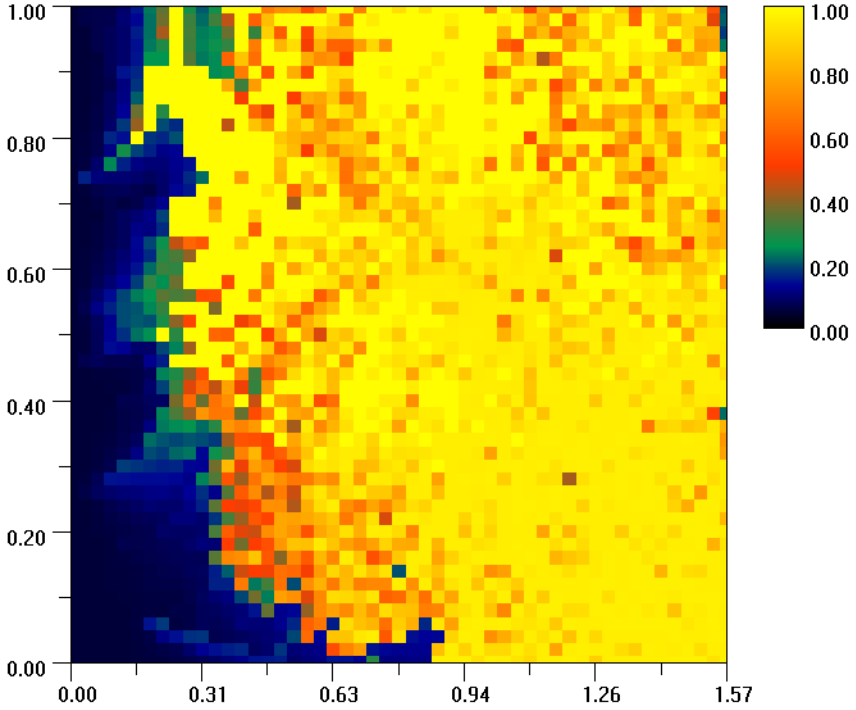


Figure 50: Values of e_{max} of a grid (θ_x, B) for $r_0 = 1.5$ and $c = 0.8$.

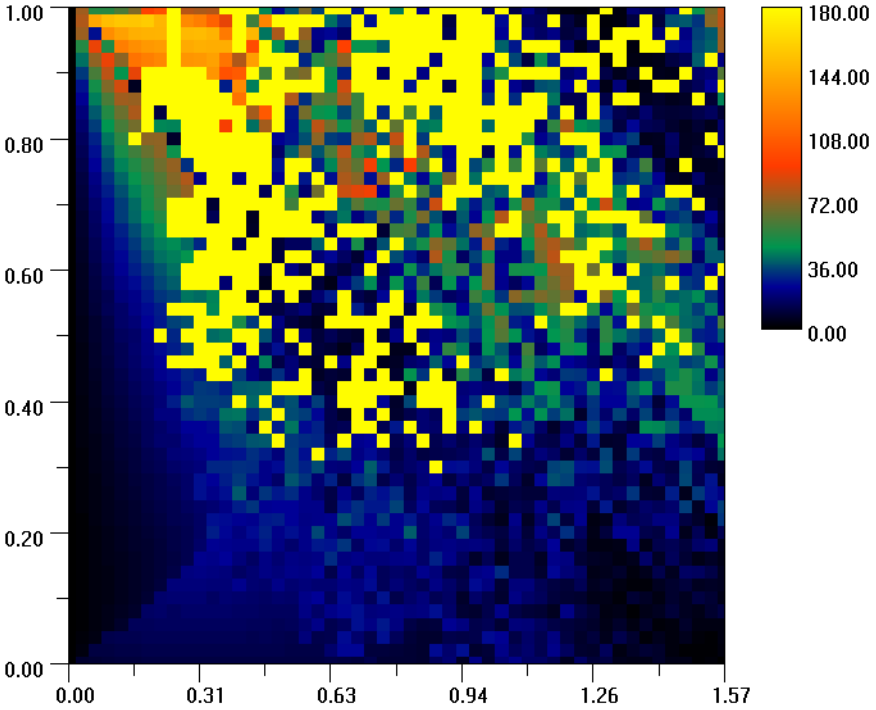


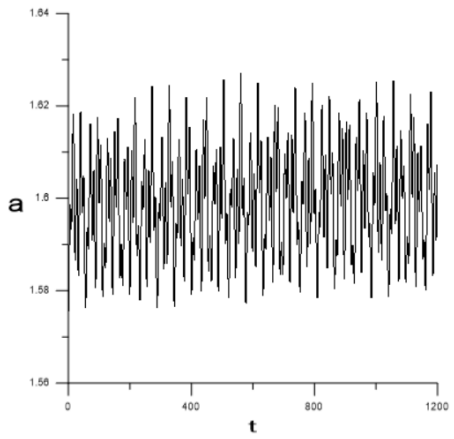
Figure 51: Values of i_{max} of a grid $(\theta_x,\ B)$ for $r_0 = 1.5$ and $c = 0.8$.

Notice that comparatively to the results in the maps of 5.1.1 the white and bright yellow region has been decreased and the blue region has been increased, a fact that is reasonable, since the shape of the asteroid became more spherical.

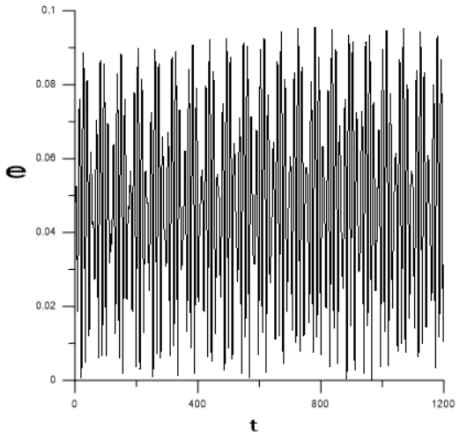
Stable, bounded orbits For an orbit with initial conditions

$$\theta_{x0} = 0.18840 \qquad B_0 = 0.24$$

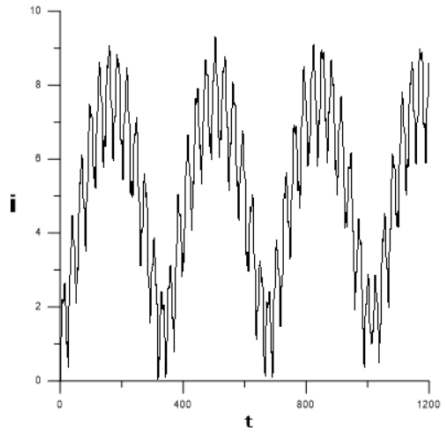
we have the evolution in figure 52 and the orbit in figure 53



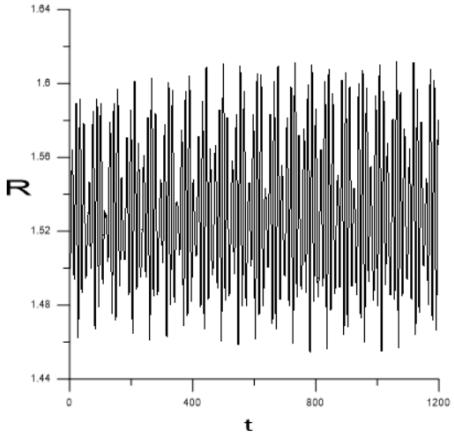
(a) a(t)



(b) e(t)



(c) i(t)



(d) R(t)

Figure 52: $\theta_{x0} = 0.18840 \qquad B_0 = 0.24.$

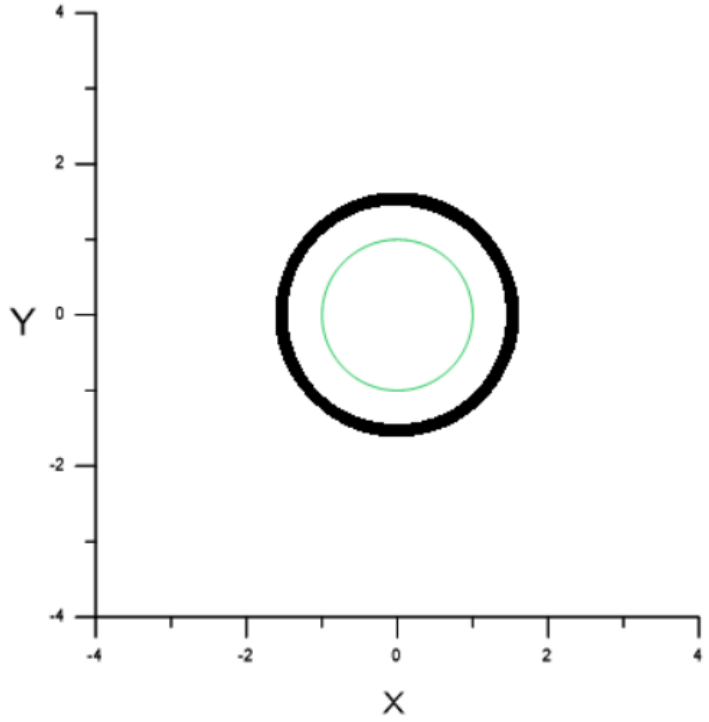
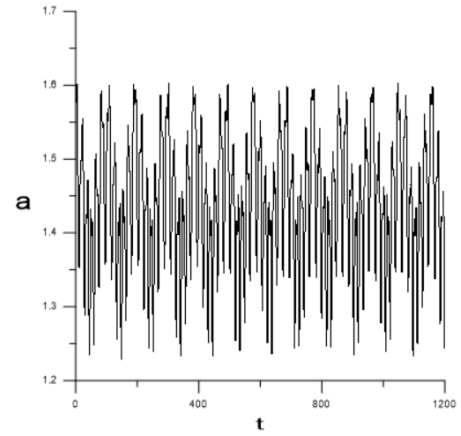


Figure 53: Orbit on XY plane with $\theta_{x0} = 0.18840 \qquad B_0 = 0.24$.

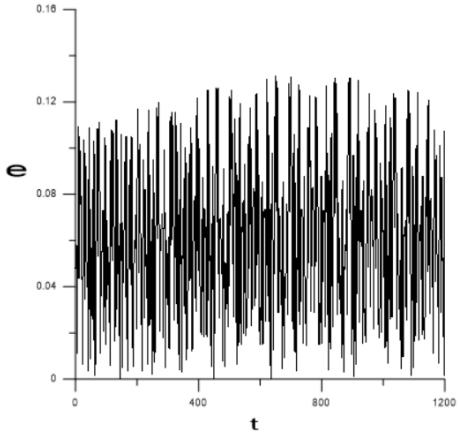
For another stable orbit with initial conditions

$$\theta_{x0} = 0.2512 \qquad B_0 = 0.72$$

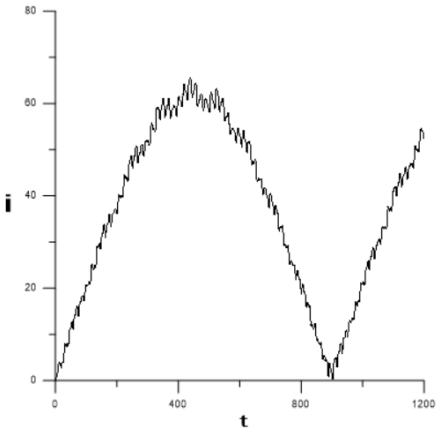
we have for the evolution the figure 54 and for the orbit the figure 55



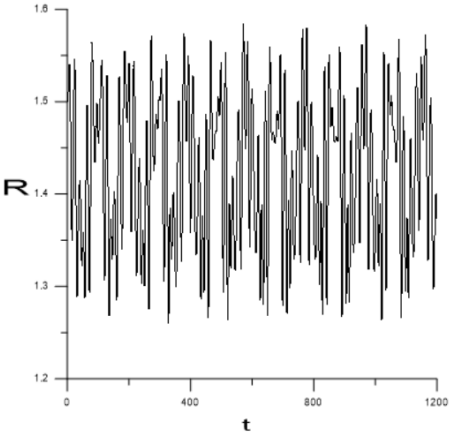
(a) a(t)



(b) e(t)



(c) i(t)



(d) R(t)

Figure 54: $\theta_{x0} = 0.2512 \qquad B_0 = 0.72$.

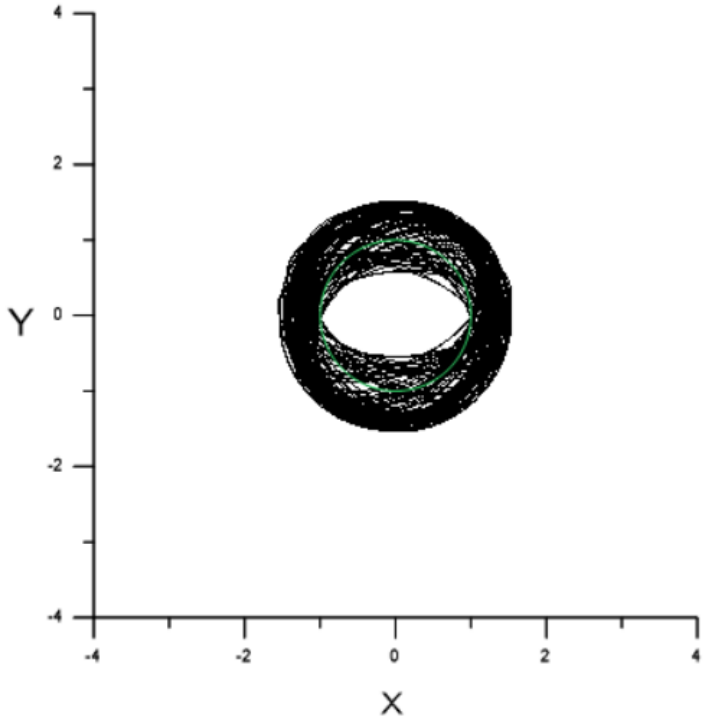


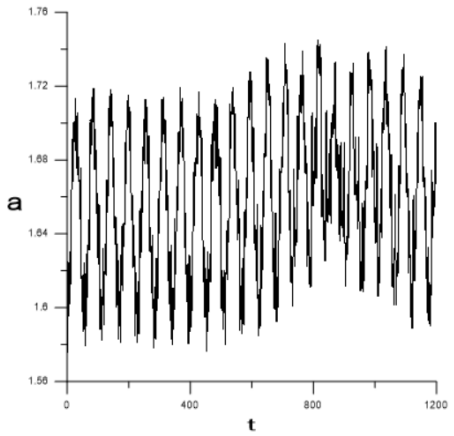
Figure 55: Orbit on the XY plane with $\theta_{x0} = 0.2512 \qquad B_0 = 0.72$.

In this case, we notice that the inclination gets large values and the orbit remains bounded.

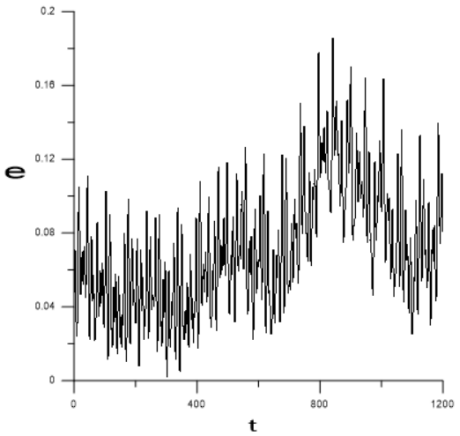
Unstable orbits For an orbit with initial conditions

$$\theta_{x0} = 0.5338 \qquad B_0 = 0.04$$

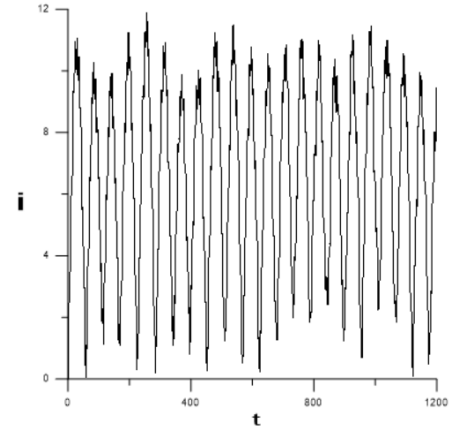
we have the evolution in figure 56 and the orbit in figure 57



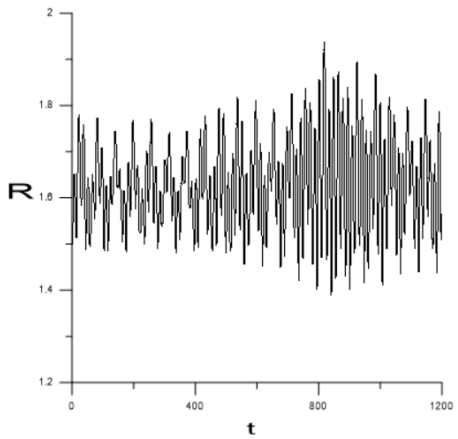
(a) a(t)



(b) e(t)



(c) i(t)



(d) R(t)

Figure 56: $\theta_{x0} = 0.5338 \qquad B_0 = 0.04$.

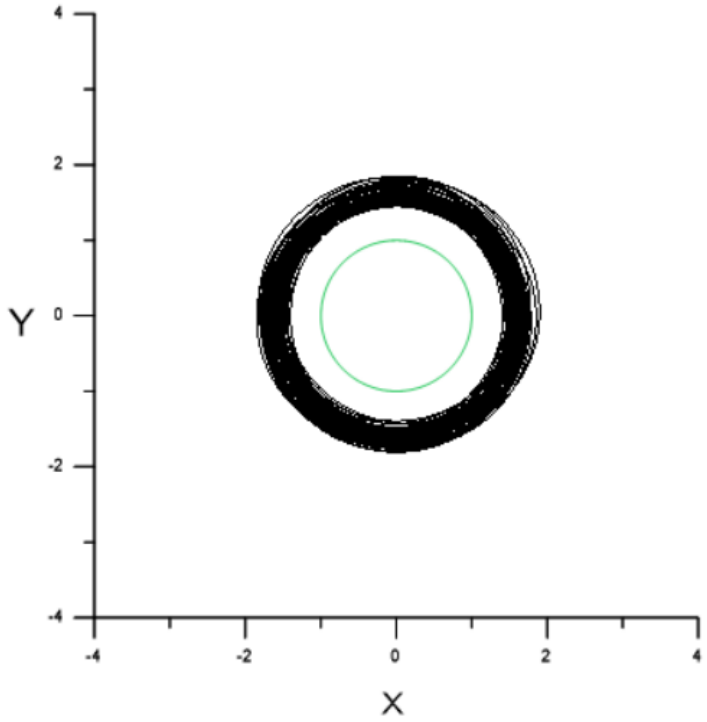


Figure 57: Orbit on the XY plane with $\theta_{x0} = 0.5338 \qquad B_0 = 0.04$.

Notice in figure 56 (d) that the radius of the orbit has a chaotic evolution, so the orbit is unstable. This may lead to escape or collision over time.

Consider an orbit with initial conditions

$$\theta_{x0} = 0.2826 \qquad B_0 = 0.88.$$

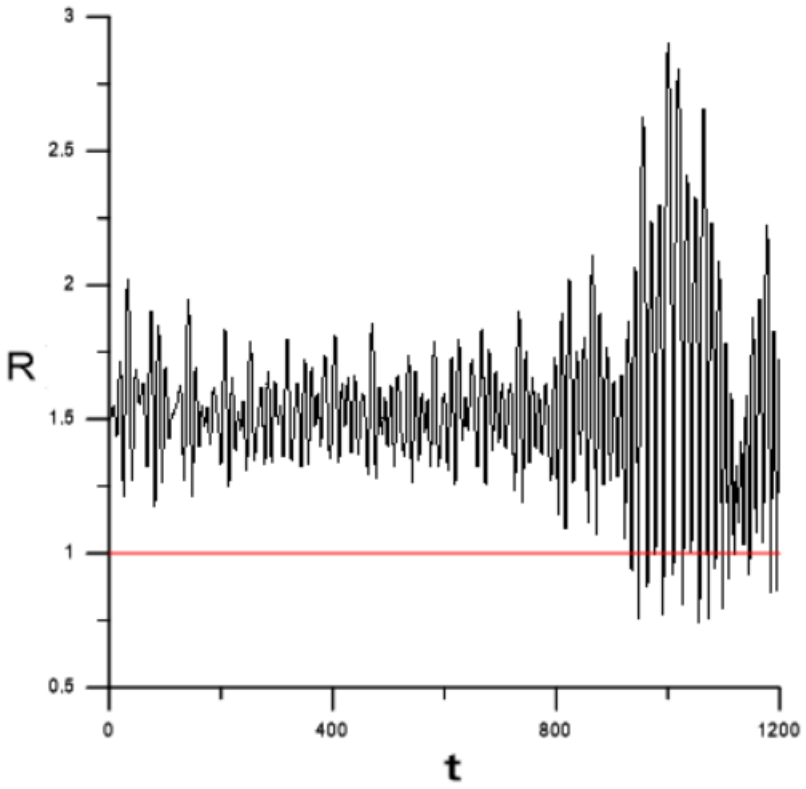


Figure 58: R-t with $\theta_{x0} = 0.2826 \qquad B_0 = 0.88$ (Collision).

This orbit is chaotic and ends up in a collision.

5.1.4 $\mathbf{r_0 = 1.5}$ and $\mathbf{c = 0.4}$

Finally, we set a significantly smaller vertical semi-axis, $c = 0.4$, with an initial radius $r_0 = 1.5$. Figure 59 represents the dynamical map for a_{max}, figure 60 represents the dynamical map for e_{max} and figure 61 for i_{max}.

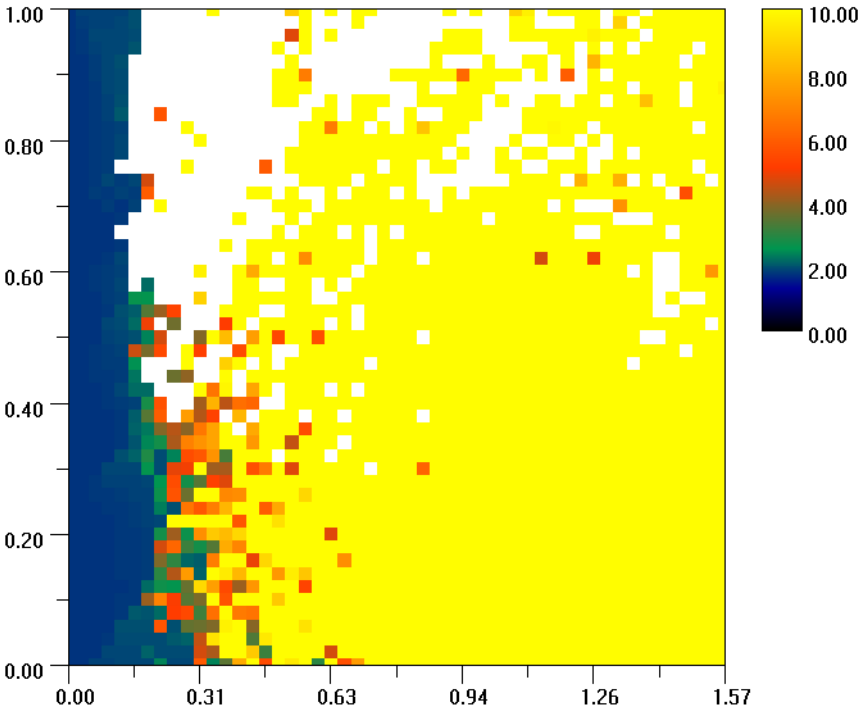


Figure 59: Values of a_{max} for a grid (θ_x, B) and for $r_0 = 1.5$ and $c = 0.4$.

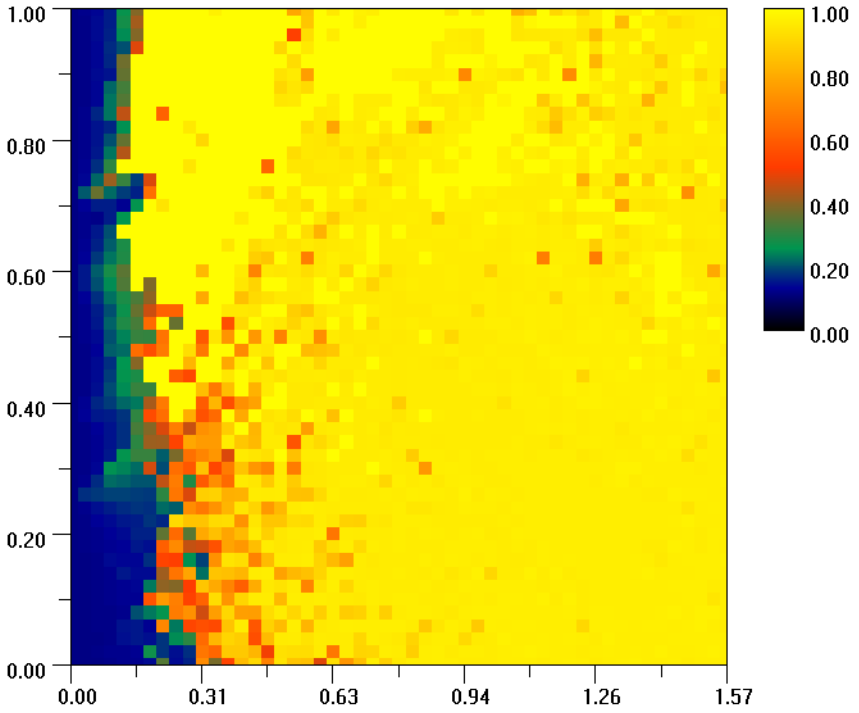


Figure 60: Values of e_{max} for a grid (θ_x, B) and for $r_0 = 1.5$ and $c = 0.4$.

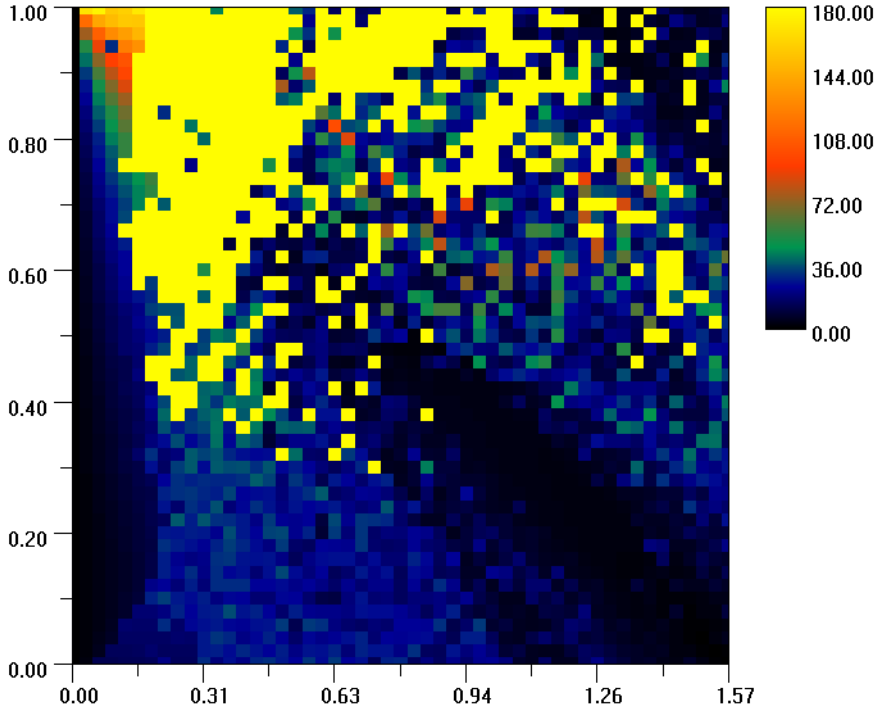


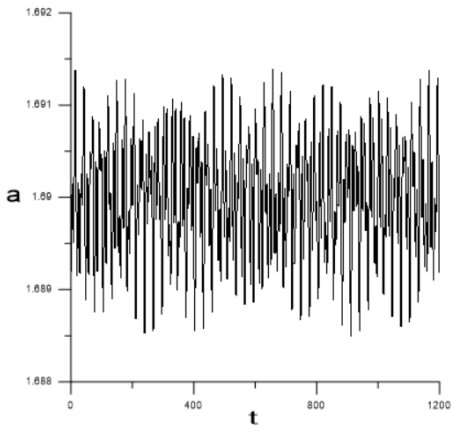
Figure 61: Values of i_{max} for a grid (θ_x, B) and for $r_0 = 1.5$ and $c = 0.4$.

It is important to note here, that in this case the blue region of stable, bounded orbits has decreased compared to the cases of $c = 0.7$ and $c = 0.8$.

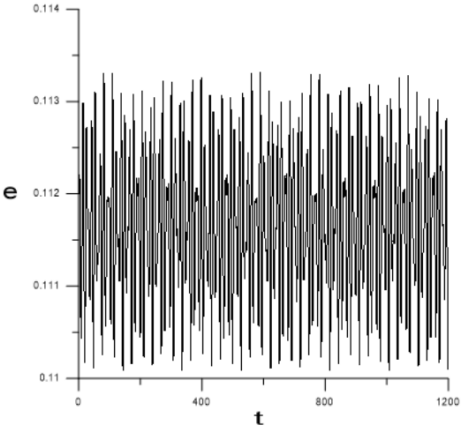
Stable, bounded orbits For an orbit with initial conditions

$$\theta_{x0} = 0.0314 \qquad B_0 = 0.16$$

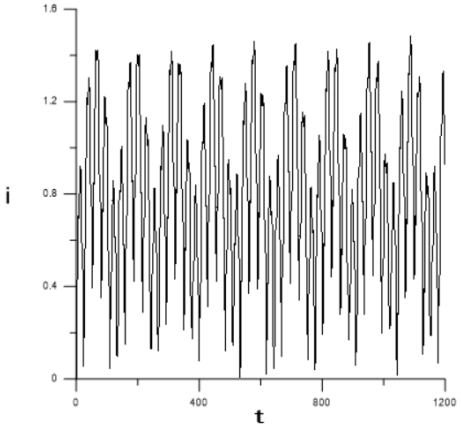
we have the evolution in figure 62 and the orbit in figure 63



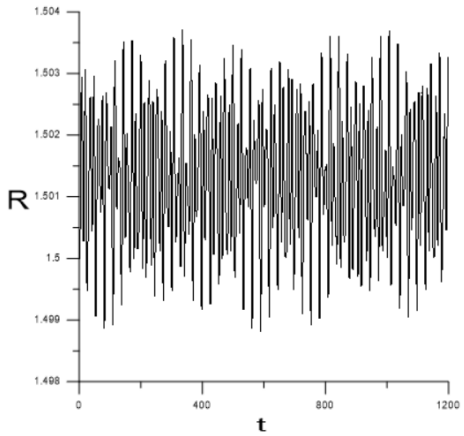
(a) a(t)



(b) e(t)



(c) i(t)



(d) R(t)

Figure 62: $\theta_{x0} = 0.0314 \qquad B_0 = 0.16.$

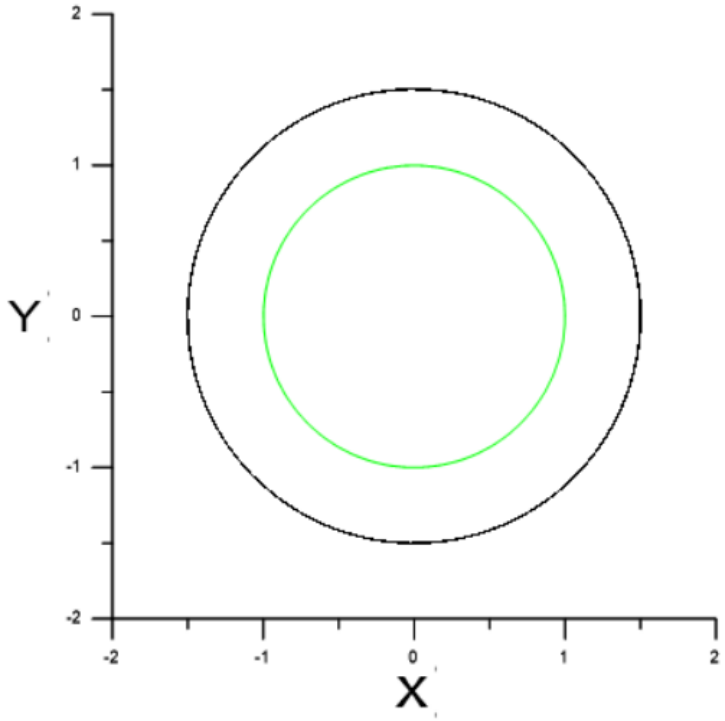


Figure 63: Orbit on XY plane with $\theta_{x0} = 0.0314 \qquad B_0 = 0.16$.

As we can see, the orbit stays bounded over time and the magnitudes of semi-major axis, eccentricity, inclination and orbit's radius oscillate.

Collision We consider an example of an orbit from the region representing collision (white for a_{max} - figure 59), with initial conditions

$$\theta_{x0} = 0.4396 \qquad B_0 = 0.82.$$

and as we can see in figure 64 below, the spacecraft collides with the asteroid within a sort period of time.

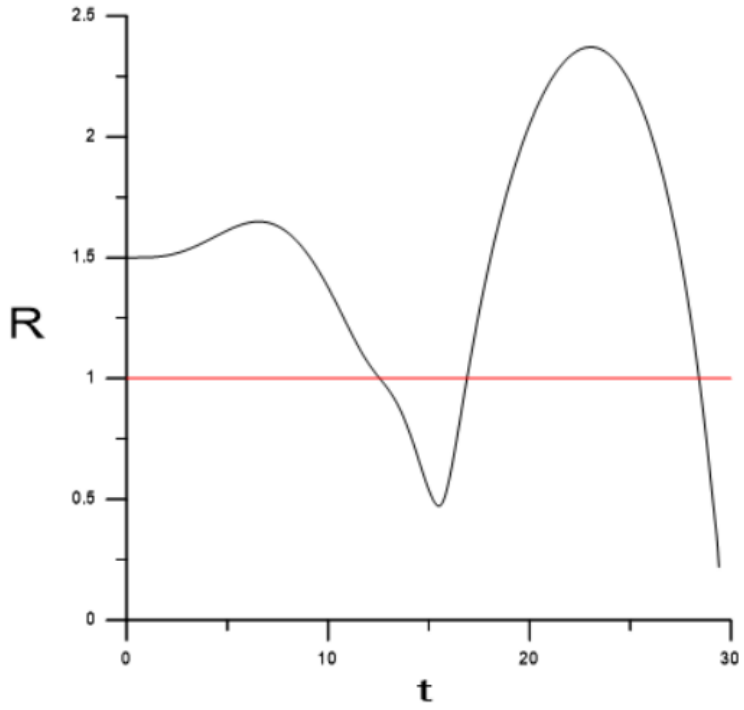


Figure 64: R-t with $\theta_{x0} = 0.4396 \qquad B_0 = 0.82$ (Collision).

Escape Choosing an orbit from the yellow region of a_{max} (figure 59) with initial conditions

$$\theta_{x0} = 1.0362 \qquad B_0 = 0.5.$$

Plotting its radius, we have the evolution given in figure 65

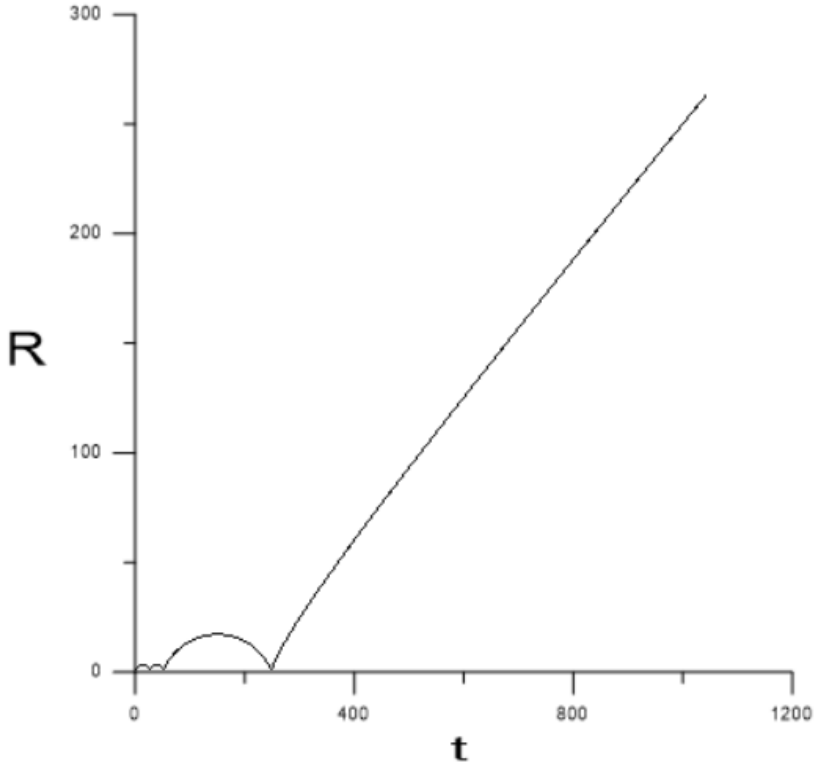


Figure 65: R-t with $\theta_{x0} = 1.0362$ $B_0 = 0.5$ (Escape).

We can see that the orbit leads to an escape of the spacecraft.

6 Conclusions

In this study, we studied the orbital dynamics around a precessing oblate spheroid asteroid. The aim of this study was to expand our knowledge about the dynamics in the gravitational field of an asteroid, that will be useful for future missions, including landings and planetary defense missions. In our model, we used a second order expansion of the gravitational potential of an arbitrary shaped rigid body and made the approximation for the oblate spheroid through the moments of inertia, which include the information of the shape. To enter the precession in our model, we solved Euler's equations for a freely rotating rigid body (the oblate spheroid in our case). We integrated the equations of motion, in the rotating frame, of a spacecraft (with negligible mass) orbiting the precessing oblate spheroid. We followed a transformation from the rotating to the inertial frame of reference, using quaternions, in order to compute the orbital elements. To be able to do this transformation, we integrated the differential equations of quaternions along with the equations of motion. Quaternions include information about the rotation and the asteroid's orientation, since they can be transformed into angles. For several pairs of initial conditions of θ_x and B, we used the a_{max}, e_{max} and i_{max} from each orbit to create dynamical maps. The darker colours (the black, the blue and the green) of the maps represent small values and brighter colours (the yellow and the orange) represent big values.

The results showed that the maps consist of regions of stable orbits and unstable orbits. In these maps there are bounded orbits (in the stable regions), there are other orbits that lead to collision or escape and there are chaotic orbits, whose evolution over time may lead to a collision or escape. Thus, the form of the dynamical maps can change depending on the integration time. From the dynamical maps of a_{max} values, for example, we can say that the yellow region are escape orbits, since the semi-major axis takes large values. The white represents collision. In the blue area, the semi-major axis takes small values, but not all the orbits are necessarily bounded or periodic. That is because the radius of some of these orbits evolves chaotically. Hence, they can end up to a collision or an escape.

By changing the initial radius of the orbit, we noticed that an increase of its magnitude leads to an increase to the stable orbits' region. We doubled the initial radius (from $r_0 = 1.5$ to $r_0 = 3.0$) and noticed the unstable region decreasing significantly.

Another parameter that affects the form of the dynamical maps is the vertical semi-axis c of the oblate spheroid. From our results, we can see that the stable region decreases as the parameter also decreases and the rigid body becomes more oblate.

For future research, we can solve the variational equations, along with the equations of motion and compute the Lyapunov Exponents and the Fast Lyapunov Indicator (FLI) to detect chaos.

One more idea is to extend our analysis by entering the force caused by the solar radiation pressure in the equations of motion.

Appendices

A main1c.cpp

```
#include <iostream>
#include "ODESBS.h"
#include "dSystemElpsdRotPrec23.h"
#include <string.h>
#include "restroe41.h"
#include <cmath>

void ShowConfig();
void WriteConfig();
void FileOutput(int Init, double X[]);


double X0[NEQ];

int FLAG_GETPQ, SCRNMOD;
double timefactor, MaxResc;
double acc,dt0, DT; int Niters;
char fnameX[256], fnameO[256], fnameXIn[256];
FILE *filx, *filo, *filxin; //, *fili;


int main()
{
        double XI[NEQ];
        double theta, A, B, q00, q10, q20, q30, r0;
        printf("RUN 3D-ORBIT in the field of Rotating oblate \n\
           n");
        double a = 1.0, b, c;
        b = 1.0;
        c = 0.7;
        theta=0.7536;
        B=0.58;
        r0=3.0;
        A = sqrt(1.0-cos(theta)*cos(theta));
        initializePrecSystem(b, c, 0.8,A, B, 1.0e-14);

        //initial quaternions
        q00=cos(theta*0.5);
```

```
    q10=sin(theta*0.5);
    q20=0.0;
    q30=0.0;
    InitRestroeB(0, 0.0, 0.8);
    //initial conditions in inertial frame
    XI[0]=0;       //t
    XI[1]=r0;     //X
    XI[2]=0.0;     //Y
    XI[3]=0.0;     //Z
    XI[4]=0;       //VX
    XI[5]=0.0;     //VY
    XI[6]=0.0;     //VZ
    XI[7] = q00; //q0
    XI[8] = q10;    //q1
    XI[9] = q20;    //q2
    XI[10] = q30;  //q3


    double Ix = (b*b + c*c) / 5,  Iy = (a*a + c*c) / 5, Iz =
        (a*a + b*b) / 5;
    double F =0.8*(3*(Ix-Iz)/(2*pow(XI[1],4)) - 1/(XI[1]*XI
       [1])); //3*(Ix-Iz)/(2r^4)-1/r^2
    double va = sqrt(-XI[1]*F); //approximation for circular
        orbit
    XI[5] = va;

InertialtoRotating(XI,X0);

    double Energy0 = Energy(X0);
    //RotatingToInertial(X0, XI);
    OEPOS oe;
    GetOrbitalElements(XI, &oe, 1);
    printf("a=%f   e=%f   i=%f\n",oe.a,oe.e,oe.i);
    printf("M=%f   w=%f   W=%f\n",oe.M,oe.omega, oe.Omega);
    getchar();

    DT = 0.05;
    Niters = 24000;
    dt0 = 0.001;
    //-------------------------
    //ShowConfig();
    strcpy(fnameX,"testx2.dat");
    strcpy(fnameO,"testo.dat");
    strcpy(fnameXIn,"testxin.dat");
```

```
        FileOutput(0, X0);
        double Y0[NEQ], Y1[NEQ];
        for (int j = 0; j < NEQ; j++) Y0[j] = X0[j];
        for (int i = 1; i <= Niters; i++)
        {
                if (!integrate.DTstep(Y0, DT, &dt0, Y1)) {
                   printf("Break: Integration error\n"); break;
                   }
                for (int j = 0; j<NEQ; j++) Y0[j] = Y1[j];
                FileOutput(1, Y0);
                double dee = fabs(Energy(Y0) - Energy0) /
                   Energy0;
        printf("%d   x=%f   y=%f   z=%f   DE/E=%1.1e\n", i, Y0
           [1], Y0[2], Y0[3], dee);
    }
        FileOutput(-1, Y0); //fclose(fili);

        printf("\nEnd of Run"); getchar();
        return 0;
}

////////////////////////////////////////////
void FileOutput(int Init, double X[])
{
        double Xi[NEQ];
        if (Init == -1) { fclose(filx); fclose(filo); fclose(
           filxin); return; }
        if (Init == 0) {
                filx = fopen(fnameX, "wt");
                filo = fopen(fnameO, "wt");
                filxin = fopen(fnameXIn, "wt");
                fprintf(filx, "  t           x              y
                   z             P             Q
                   R             Ener\n");
                fprintf(filo, "  t           a           e
                   i       omegabar    omega    Omega    M
                   lambda    n/n'\n");
                fprintf(filxin, "  t           X             Y
                   Z             VX             VY
                   VZ       Radius\n");
        }
        fprintf(filx, "%f  %11.10lf  %11.10lf  %11.10lf  %11.10
           lf  %11.10lf  %11.10lf  %14.13lf\n",X[0],X[1],X[2],X
           [3],X[4],X[5],X[6],Energy(X));
```

```
        RotatingToInertial(X, Xi);
        double R=sqrt(pow(Xi[1],2)+pow(Xi[2],2)+pow(Xi[3],2));
        fprintf(filxin, "%f  %11.10lf  %11.10lf  %11.10lf
            %11.10lf  %11.10lf  %11.10lf  %11.10lf  %11.10lf\n", Xi[0], Xi
            [1], Xi[2], Xi[3], Xi[4], Xi[5], Xi[6],R);
        OEPOS oe;
        GetOrbitalElements(Xi, &oe, 1);
        fprintf(filo, "%f  %8.7f  %8.7f  %8.5f  %6.3f  %6.3f
            %6.3f  %6.3f    %6.3f  %5.3f\n", Xi[0], oe.a, oe.e, oe
            .i, oe.omegabar, oe.omega, oe.Omega, oe.M, oe.lamda,
            oe.meanmotion);
}
```

B dSystemElpsdRotPrec23.cpp

```
#include<stdio.h>
#include<math.h>
#include "dSystemElpsdRotPrec23.h"
#include "ODESBS.h"


double deltai, pAmpl, pFreq, wz, Ix, Iy, Iz;

ODESBS integrate, varintegrate, quatintegrate;
void dSystemElpsdRotPrec(double t,double X[],double f[]);
void dSystemElpsdRotPrecVar(double t, double X[],double f[]);
void dSystemQuaternions(double t, double q[],double dq[]);
void RotMatrix(double q[], double D[][4]);
void DotRotMatrix(double t, double q[], double DD[][4]);
void BodyFixedtoInertial(double X[], double q[]);
void InverRotMatrix(double q[], double Dinv[][4]);
void DotInverRotMatrix(double t, double q[], double DDinv[][4]);
void InertialtoRotating(double X[], double Y[]);



void initializePrecSystem(double b, double c, double delta,
   double precA, double precB, double acc)
//acc: accuracy of numerical integrations
//a=1, b,c normalized,  delta: metric units (G*mass/(omega^2*a
   ^3)
//pressecion params  : precA=amplitude<1, precB=angular
   frequency,  |omega_0|=1
{
        double a = 1.0;
        Ix = (b*b + c*c) / 5;  Iy = (a*a + c*c) / 5; Iz = (a*a +
            b*b) / 5;
        deltai = delta;
        pAmpl = precA; pFreq = precB;
        wz = sqrt(1 - pAmpl*pAmpl);  //initial norm of angular
           frequency is 1
        integrate.open(NEQ,dSystemElpsdRotPrec,0.000001,0.1,acc
           ,0);
    //varintegrate.open(NEQV,dSystemElpsdRotPrecVar
       ,0.000001,0.1,acc,0);
}
```

```
void GetRotationvector(double t, double W[])
{
    //W[1]=wx, W[2]=wy, W[3]=wz, W[4]=wxdot, ....

        W[1] = pAmpl*sin(pFreq*t);   //wx
        W[2] = pAmpl*cos(pFreq*t);   //wy
        W[3] = wz;   //wz
        W[4] = pAmpl*pFreq*cos(pFreq*t);    //dwx
        W[5] = -pAmpl*pFreq*sin(pFreq*t);    //dwy
        W[6] = 0;    //dwz
}



void dSystemElpsdRotPrec(double t, double X[],double f[])
{
        double x = X[1], y = X[2], z = X[3], P = X[4], Q = X[5],
            R = X[6];
        double ir = sqrt(x * x + y * y + z * z);
        double Ux, Uy, Uz; // , deltai;
        double W[7];


        GetRotationvector(t, W);


        // potential derivatives
        Ux= x*(6*Ix*ir*ir + 3*(Ix+Iy+Iz)*ir*ir + 2*pow(ir,4)
           -15*(Ix*x*x + Iy*y*y + Iz*z*z)) / (2*pow(ir,7));
        Uy= y*(6*Iy*ir*ir + 3*(Ix+Iy+Iz)*ir*ir + 2*pow(ir,4)
           -15*(Ix*x*x + Iy*y*y + Iz*z*z)) / (2*pow(ir,7));
        Uz= z*(6*Iz*ir*ir + 3*(Ix+Iy+Iz)*ir*ir + 2*pow(ir,4)
           -15*(Ix*x*x + Iy*y*y + Iz*z*z)) / (2*pow(ir,7));

        //Fellipsoid(x, y, z, &Ux, &Uy, &Uz);

        // equations of motion
        f[1] = P;          // dx'/dt'
        f[2] = Q;          // dy'/dt'
        f[3] = R;          // dz'/dt'
        f[4] = - W[5] * z + W[6] * y - 2* W[2] * R + 2* W[3] * Q
            - W[1] * W[2] * y + W[2] * W[2] * x + W[3] * W[3] *
```

```
        x - W[3] * W[1] * z  - deltai* Ux;  // dP
    f[5] = - W[6] * x + W[4] * z - 2* W[3] * P + 2* W[1] * R
        - W[2] * W[3] * z + W[3] * W[3] * y + W[1] * W[1] *
        y - W[1] * W[2] * x  - deltai* Uy;  // dQ
    f[6] = - W[4] * y + W[5] * x - 2* W[1] * Q + 2* W[2] * P
        - W[1] * W[3] * x + W[1] * W[1] * z + W[2] * W[2] *
        z - W[2] * W[3] * y  - deltai* Uz;  // dR
    f[7] = 0.5 * (-X[8] * W[1] - X[9] * W[2] - X[10] * W[3])
        ;
    f[8] = 0.5 * (X[7] * W[1] - X[10] * W[2] + X[9] * W[3]);
    f[9] = 0.5 * (X[10] * W[1] + X[7] * W[2] - X[8] * W[3]);
    f[10] = 0.5 * (-X[9] * W[1] + X[8] * W[2] +X[7]* W[3]);
}



double Energy(double X[])
{
    double t = X[0] , x = X[1], y = X[2], z = X[3], P = X
        [4], Q = X[5], R = X[6];
    double ir = sqrt(x * x + y * y + z * z);
    double U,V,H;
    double W[7];

    GetRotationvector(t, W);
    U = -1/ir - ( Ix + Iy + Iz )/( 2*pow(ir,3) ) + 1.5 * (
        Ix * x  * x + Iy * y * y + Iz * z * z)/( pow(ir,5) );
         // gravitational potential U'
    V = -0.5 * ((W[2] * z - W[3] * y) * (W[2] * z - W[3] * y
        ) + (W[3] * x - W[1] * z) * (W[3] * x - W[1] * z) + (
        W[1] * y - W[2] * x) * (W[1] * y - W[2] * x)) +
        deltai*U;  // effective potential V
    H = 0.5 * ( P*P + Q*Q + R*R ) + V;  // Hamilton

    return H;

}


void RotMatrix(double q[], double D[][4])
{
    D[1][1] = 1 - 2*(q[3]*q[3] + q[4]*q[4]);
    D[1][2] = 2*(q[2]*q[3] - q[1]*q[4]);
```

```
    D[1][3] = 2*(q[2]*q[4] + q[1]*q[3]);

    D[2][1] = 2*(q[2]*q[3] + q[1]*q[4]);
    D[2][2] = 1 - 2*(q[2]*q[2]+ q[4]*q[4]);
    D[2][3] = 2*(q[3]*q[4] - q[1]*q[2]);

    D[3][1] = 2*(q[2]*q[4] - q[1]*q[3]);
    D[3][2] = 2*(q[3]*q[4] + q[1]*q[2]);
    D[3][3] = 1 - 2*(q[2]*q[2]+ q[3]*q[3]);
}

void DotRotMatrix(double t, double q[], double DD[][4])
{
    double W[7], dq1, dq2, dq3, dq4;
    GetRotationvector(t, W);

    dq1 = 0.5 * (-q[2] * W[1] - q[3] * W[2] - q[4] * W[3]); //
        dq0
        dq2 = 0.5 * (q[1] * W[1] - q[4] * W[2] + q[3] * W[3]);
            //dq1
        dq3 = 0.5 * (q[4] * W[1] + q[1] * W[2] - q[2] * W[3]);
            //dq2
        dq4 = 0.5 * (-q[3] * W[1] + q[2] * W[2] + q[1] * W[3]);
             //dq3

        DD[1][1] = -4* (q[3]*dq3 + q[4]*dq4);
        DD[1][2] = 2*((dq2*q[3] + q[2]*dq3) - (dq1*q[4] + q[1]*
            dq4));
        DD[1][3] = 2*((dq2*q[4] + q[2]*dq4) + (dq1*q[3] + q[1]*
            dq3));

        DD[2][1] = 2*((dq2*q[3] + q[2]*dq3) + (dq1*q[4] + q[1]*
            dq4));
        DD[2][2] = -4* (q[2]*dq2 + q[4]*dq4);
        DD[2][3] = 2*((dq3*q[4] + q[3]*dq4) - (dq1*q[2] + q[1]*
            dq2));

        DD[3][1] = 2*((dq2*q[4] + q[2]*dq4) - (dq1*q[3] + q[1]*
            dq3));
        DD[3][2] = 2*((dq3*q[4] + q[3]*dq4) + (dq1*q[2] + q[1]*
            dq2));
        DD[3][3] = -4* (q[2]*dq2 + q[3]*dq3);
}
```

```
//multiplication matrxi x vector of size 3
void MxV(double D[][4], double v[], double Vout[])
{
        for (int i = 0; i <= 3; i++) Vout[i] = 0.0;
        for (int i = 1; i <= 3; i++) for (int j = 1; j <= 3; j
           ++)  Vout[i] += D[i][j] * v[j];
}


void RotatingToInertial(double X[], double Y[])
{
        Y[0] = X[0];
        double v1[4], v2[4], v3[4], v4[4], q[5];
        double D[4][4], DD[4][4];
        q[1] = X[7]; q[2] = X[8]; q[3] = X[9]; q[4] = X[10];
        //rotate position
        v1[1] = X[1]; v1[2] = X[2]; v1[3] = X[3];
        RotMatrix(q, D);
        MxV(D, v1, v3);
        Y[1] = v3[1]; Y[2] = v3[2]; Y[3] = v3[3];
        //rotate velocity
        v2[1] = X[4]; v2[2] = X[5]; v2[3] = X[6];
        DotRotMatrix(X[0], q, DD);
        MxV(D, v2, v3);
        MxV(DD, v1, v4);
        Y[4] = v3[1]+v4[1]; Y[5] = v3[2]+v4[2]; Y[6] = v3[3]+v4
           [3];
}


void InverRotMatrix(double q[], double D[][4]) //transpose of D
{
    D[1][1] = 1 - 2*(q[3]*q[3] + q[4]*q[4]);
    D[2][1] = 2*(q[2]*q[3] - q[1]*q[4]);
    D[3][1] = 2*(q[2]*q[4] + q[1]*q[3]);

    D[1][2] = 2*(q[2]*q[3] + q[1]*q[4]);
    D[2][2] = 1 - 2*(q[2]*q[2]+ q[4]*q[4]);
    D[3][2] = 2*(q[3]*q[4] - q[1]*q[2]);

    D[1][3] = 2*(q[2]*q[4] - q[1]*q[3]);
    D[2][3] = 2*(q[3]*q[4] + q[1]*q[2]);
    D[3][3] = 1 - 2*(q[2]*q[2]+ q[3]*q[3]);
```

```
}


void DotInverRotMatrix(double t, double q[], double DD[][4])
{
    double W[7], dq1, dq2, dq3, dq4;
    GetRotationvector(t, W);

    dq1 = 0.5 * (-q[2] * W[1] - q[3] * W[2] - q[4] * W[3]); //
        dq0
        dq2 = 0.5 * (q[1] * W[1] - q[4] * W[2] + q[3] * W[3]);
            //dq1
        dq3 = 0.5 * (q[4] * W[1] + q[1] * W[2] - q[2] * W[3]);
            //dq2
        dq4 = 0.5 * (-q[3] * W[1] + q[2] * W[2] + q[1] * W[3]);
             //dq3

        DD[1][1] = -4* (q[3]*dq3 + q[4]*dq4);
        DD[2][1] = 2*((dq2*q[3] + q[2]*dq3) - (dq1*q[4] + q[1]*
            dq4));
        DD[3][1] = 2*((dq2*q[4] + q[2]*dq4) + (dq1*q[3] + q[1]*
            dq3));

        DD[1][2] = 2*((dq2*q[3] + q[2]*dq3) + (dq1*q[4] + q[1]*
            dq4));
        DD[2][2] = -4* (q[2]*dq2 + q[4]*dq4);
        DD[3][2] = 2*((dq3*q[4] + q[3]*dq4) - (dq1*q[2] + q[1]*
            dq2));

        DD[1][3] = 2*((dq2*q[4] + q[2]*dq4) - (dq1*q[3] + q[1]*
            dq3));
        DD[2][3] = 2*((dq3*q[4] + q[3]*dq4) + (dq1*q[2] + q[1]*
            dq2));
        DD[3][3] = -4* (q[2]*dq2 + q[3]*dq3);
}


void InertialtoRotating(double X[], double Y[])
{
        Y[0] = X[0];
        double v1[4], v2[4], v3[4], v4[4], q[5];
        double Dinv[4][4], DDinv[4][4];
        q[1] = X[7]; q[2] = X[8]; q[3] = X[9]; q[4] = X[10];
        //rotate position
```

```
        v1[1] = X[1]; v1[2] = X[2]; v1[3] = X[3];
        InverRotMatrix(q, Dinv);
        MxV(Dinv, v1, v3);
        Y[1] = v3[1]; Y[2] = v3[2]; Y[3] = v3[3];
        //rotate velocity
        v2[1] = X[4]; v2[2] = X[5]; v2[3] = X[6];
        DotInverRotMatrix(X[0], q, DDinv);
        MxV(Dinv, v2, v3);
        MxV(DDinv, v1, v4);
        Y[4] = v3[1]+v4[1]; Y[5] = v3[2]+v4[2]; Y[6] = v3[3]+v4
            [3];
        Y[7] = q[1]; Y[8] = q[2]; Y[9] = q[3]; Y[10] = q[4];
}
```

References

[1] [Online]. Available: https://en.wikipedia.org/wiki/Asteroid

[2] https://en.wikipedia.org/wiki/Ellipsoid.

[3] [Online]. Available: https://solarsystem.nasa.gov/asteroids-comets-and-meteors/asteroids/in-depth/#otp_many_shapes_and_sizes

[4] B. Persson and J. Biele, "On the stability of spinning asteroids," *Tribology Letters*, vol. 70, no. 2, pp. 1–19, 2022.

[5] D. Karydis, G. Voyatzis, and K. Tsiganis, "A continuation approach for computing periodic orbits around irregular-shaped asteroids. an application to 433 eros," *Advances in Space Research*, vol. 68, no. 11, pp. 4418–4433, 2021.

[6] [Online]. Available: https://solarsystem.nasa.gov/

[7] [Online]. Available: https://global.jaxa.jp/projects/sas/muses_c/

[8] A. Petit, J. Souchay, and C. Lhotka, "High precision model of precession and nutation of the asteroids (1) ceres,(4) vesta,(433) eros,(2867) steins, and (25143) itokawa," *Astronomy & Astrophysics*, vol. 565, p. A79, 2014.

[9] M. Tarnopolski, "Rotation of an oblate satellite: Chaos control," *Astronomy & Astrophysics*, vol. 606, p. A43, 2017.

[10] R. Fitzpatrick, "Newtonian dynamics," 2011.

[11] D. Χατζηδημητρίου, "*θεωρητική* Μηχανική," *Νευτώνεια Μηχανική*, 1983.

[12] D. H. Andrews, "The theory of the potential (macmillan, william duncan)," 1930.

[13] Y. Jiang and H. Baoyin, "Orbital mechanics near a rotating asteroid," *Journal of Astrophysics and Astronomy*, vol. 35, no. 1, pp. 17–38, 2014.

[14] V. Martinusi and D. Condurache, "Remarks on the hamiltonian of a particle in a rotating frame," *Buletinul Institutului Politehnic din Iaşi. Secţia I. Matematică, Mecanică Teoretică, Fizică*, vol. 55, 01 2009.