MOID USING NEW SETS OF UNIVERSAL FUNCTIONS

AYMAN HOMDA
*Egyptian Space Agency, Cairo, Egypt, 130, ayman.homda@egsa.gov.eg*

Hany R. Dwidar
*Astronomy, Space Science and Meteorology Department Faculty of Science, Cairo University, Cairo, Egypt, 12613*

Abdelaziz A. Bakry
*Astronomy and Meteorology Department Faculty of Science, Al-Azhar University Nasr City, Cairo, Egypt, 11884*

M.N. Mohamad Ismail
*Astronomy and Meteorology Department Faculty of Science, Al-Azhar University Nasr City, Cairo, Egypt, 11884*

AHMED EI-RAFFIE
*Egyptian Space Agency, Cairo, Egypt, 130*

Follow this and additional works at: [https://absb.researchcommons.org/journal](https://absb.researchcommons.org/journal)

Part of the *Astrophysics and Astronomy Commons*

**How to Cite This Article**
HOMDA, AYMAN; Dwidar, Hany R.; Bakry, Abdelaziz A.; Ismail, M.N. Mohamad; and EI-RAFFIE, AHMED (2024) "MOID USING NEW SETS OF UNIVERSAL FUNCTIONS," *Al-Azhar Bulletin of Science*: Vol. 35: Iss. 1, Article 3.
DOI: [https://doi.org/10.58675/2636-3305.1662](https://doi.org/10.58675/2636-3305.1662)

This Original Article is brought to you for free and open access by Al-Azhar Bulletin of Science. It has been accepted for inclusion in Al-Azhar Bulletin of Science by an authorized editor of Al-Azhar Bulletin of Science. For more information, please contact kh_Mekheimer@azhar.edu.eg.
Minimum Orbit Intersection Distance Using New Sets of Universal Functions

Ayman Homda Mohamed a,*, Hany Ramadan Dwidar b, Abdel Aziz Bakry c, Mohamed Nader Ismail c, Ahmed Yahya EL-Raffie a

a Egyptian Space Agency, Egypt
b Department of Astronomy, Space Science and Meteorology, Faculty of Science, Cairo University, Cairo, Egypt
c Department of Astronomy and Meteorology, Faculty of Science, Al-Azhar University, Nasr City, Cairo, Egypt

Abstract

In this paper, based on Goodyear’s time transformation formula, we used a set of modified universal functions to construct the minimum distance function between any two celestial objects. We determined the distance between objects in space under a specific time constraint. We used the continued fractions method for quick convergence of the distance function. We used the inverse series to obtain a first initial guess to solve the convergence equation. Furthermore, the Lagrange multiplier method was used to determine the minimum distance between the two objects under the specified time constraint. We constructed an algorithm and applied it with the Matlab toolbox to numerically simulate the problem under consideration. We found that the deviation in the position and velocity vectors using both Ansys Systems Tool Kit (STK) and MATLAB are \( \frac{7}{52234} \) km and \( \frac{6045}{7} \) km/s, respectively for some satellites. Also, the deviation for distance function is \( 3.956 \times 10^{-5} \) km.

Keywords: Closest approach, Numerical integration, Objective function, Universal functions

1. Introduction

The Minimum Orbit Intersection Distance (MOID) is a crucial parameter in situational awareness, as it allows space operators to monitor and manage the risk of collision between objects in orbit. By continuously updating the MOID for all objects in a given region of space, it is possible to prioritize and mitigate the risk of collision, ensuring the safety and sustainability of space operations. So, it is an important concept in space science and technology, as it provides a way to assess and manage the risk of collision between objects in space [1–6].

We used the Lagrange multiplier technique to handle the dynamic problem under consideration. The Lagrange multiplier technique finds the maximum or minimum of a multivariable function \( f(x, y, \ldots) \) when there is some constraint on the input values you are allowed to use. This technique only applies to constraints that look something like \( g(x, y, \ldots) = cg \) is another multivariable function with the same input space as \( f \) and \( c \) is some constant. The core idea is to look for points where the contour Lines of \( f \) and \( g \) are parallel to each other, [7]. In addition, we used Newton’s iterative method, which is used to solve simultaneous nonlinear equations. This method solves a system of two nonlinear equations with two unknown \( x, y \) the on form of \( f_1(x, y) = 0 \) and \( f_2(x, y) = 0, \) [8].

Sharaf (1998) presented two approaches for solving the nearest approach problem. The first approach employs one-dimensional minimization, while the second utilizes constraint minimization with nonlinear simultaneous equations. Additionally, the paper proposes adaptive second-order iterative algorithms for solving the global Kepler problem effectively [9].
Sharaf (2016) discussed the investigation of a new set of general functions based on Goodyear’s time transformation formula. The author defines the linear independence of these functions, presents their differential and iterative forms, and explores their relationships with elementary functions. This paper presents the complete list of identities for these functions and presents exact analytical formulations for the general global Kepler equation for time $t \in [t_s, ts]$, along with some orbital parameters. Power series representations of these equations are generated, and several frequency formulas are given to simplify their calculations. Finally, Kepler’s general equation is solved symbolically, and analytical expressions for the series coefficients are presented in Horner form for efficient evaluation [10].

There are several works in this field; for example, Wisdom and Hernandez (2015) presented a fast and accurate solution to the initial value problem for Keplerian motion in universal variables that do not use the Stumpf series. They find that it performs better than methods based on the Stumpf f series [11].

Wisniows and Rickma (2013) described a new numerical and iterative method for computing MOIDs between two heliocentric orbits. The method is compared with Giovanni Gronchi’s algebraic method. The approach involves geometric scanning and tuning, using a meridional plane for initial scanning and an efficient tuning technique to zoom in on the MOID configuration. The method ensures high accuracy and aims to avoid missing the MOID. The authors demonstrate that their method is fast, reliable, and flexible [12].

Recently, (Yassen et al., 2023), numerically investigated the initial-value problem of two body using the universal anomaly approach. To clarify the problem, the authors carried out several numerical examples using a home-made software package. They showed that the universal anomaly approach facilitates the numerical and analytical treatments of the two-body dynamics and works equally well for different types of orbits. The authors continued their work and applied the same idea in canonical theory. They studied the dynamics of Earth satellites that move in low orbits. The force model comprises the gravitational resonance 13 : 1, besides the Earth’s gravitational potential up to the second degree and order. In order to avoid the appearance of singularities, the quasi-Hamiltonian equations of motion are formulated in terms of non-singular universal variables instead of Delaunay variables [13,14].

As satellite clusters or constellations are deployed more often, accurate orbit position is becoming more and more important to maintain network accuracy, operational efficiency, and space safety. The propagation of random errors in the initial orbit location over time highlights the importance of rapid and precise estimating techniques to improve satellite control. Conventional methods are expensive, time-consuming, and may not be as accurate as they may be, especially for large-scale constellations. (Maocai Wang et al., 2023), presented a new random error evaluation model based on the ellipsoid. For satellites in any orbit, the model allows the calculation of beginning locations and error propagation. The efficacy, simplicity, and correctness of the suggested model are validated by experimental findings obtained by the Monte Carlo approach [15]. Ruishan Zhao proposes a nonzero Doppler simulation method to simulate geometric positioning accuracy. A virtual simulation geometric model is constructed using simulated satellite ephemeris and ground control point (GCP) coordinates to simulate geometric positioning accuracy under different errors and imaging observation conditions. Additionally, a geometric accuracy simulation method based on mean value compensation is introduced to simulate geometric positioning accuracy with GCPs. Experimental results demonstrate that Doppler center frequency error and velocity error have a significant impact on the geometric positioning accuracy of medium Earth orbit (MEO) and high Earth orbit (HEO) Synthetic Aperture Radar (SAR) satellites compared with low Earth orbit (LEO) SAR satellites, with maximum errors reaching approximately 1597 m. Moreover, the proposed method achieves geometric positioning accuracies of 1–10 m for MEO SAR satellites and 7–29 m for HEO SAR satellites with GCPs [16].

In this paper, we have used a constrained minimization technique to determine the point of closest approach of any two objects, whose position and velocity vectors are known. Furthermore, we use the new set Y-universal functions [17]. This function describes all different types of orbits. Moreover, we have used the Y-universal function to construct the minimum distance function between any two objects. Also, we have used the constraint optimization Lagrange multipliers technique to calculate the minimum distance between any two objects. We carried out several numerical explorations to find the minimum distance between different types of orbits. Finally, we have verified our solution using ‘STK’ and ‘Border Hasian Matrix’.
2. Universal formulas for conic orbits

In addition to the well-known varieties of conic motion, perturbing forces acting over finite time intervals can occasionally change the given type of orbit. Additionally, a full interplanetary transfers consists of three main parts: (1) escaping the departure planet, (2) transferring heliocentrically, and (3) being captured by the target planet. Consequently, various types of two-body motion appear during space missions, necessitating the urgent need for a universal formulation of the problem. So, we formulate the equations of the problem in terms Y-universal Function which describes all of these types of orbits. We now proceed to evaluate the Goodyear's time transformations formula. From the law of areas

\[ r^2 \frac{df}{dt} = h \]  

(1)

where \( r \) is the magnitude is position vector \( \vec{r}, f, h, \mu \) are the true anomaly, angular momentum and gravitational parameter, respectively. We obtain for the three known conic sections

\[ dt = r \chi_p; \quad dt = r \chi_e \text{ and } dt = r \chi_h \]  

(2)

where

\[ \chi_p = \sqrt{\frac{p}{\mu}} \left( \tan \frac{f}{2} \right), \quad \chi_e = \sqrt{\frac{a}{\mu}} \frac{dE}{a}, \quad \chi_h = \sqrt{\frac{a}{\mu}} \frac{dH}{a} \]

are the generalized anomalies for parabolic, elliptic and hyperbolic orbits, respectively [18]. It is remarkable that when \( \chi \) is used as the independent variable instead of the time \( t \), the nonlinear equations of motion for two body can be converted into linear constant-coefficient differential equations. The Goodyear's time transformations formula given by Ref. [10].

\[ \frac{dt}{d\chi} = r \]  

(3)

Equation (3) can be written as

\[ \frac{dr}{d\chi} = (\vec{r} \cdot \vec{v}) \]  

(4)

differentiating a second time, we get

\[ \frac{d^2r}{d\chi^2} = \mu (1 - \alpha r) \]  

(5)

It is convenient to write \( \alpha \) for 1/a (the semimajor axis of the orbit), so that, at the initial time \( t \), the constant \( \alpha \) is

\[ \alpha \equiv -\frac{1}{a} = 2 - \frac{v^2}{\mu} \]

(6)

After some mathematical operations we have

\[ \frac{d^2\sigma}{d\chi^2} + \alpha \mu \sigma = 0, \quad \frac{d^2r}{d\chi^2} + \alpha \mu \frac{dr}{d\chi} = 0, \quad \frac{d^4t}{d\chi^4} + \alpha \mu \frac{dt}{d\chi} = 0 \]  

(7)

where \( \sigma = (\vec{r} \cdot \vec{v}) \). Equation (7) represent a set of linear differential equations with constant coefficients satisfied by \( \sigma, r, t, \text{ and } \vec{r} \). Their solutions are better developed in a form utilizing a family of Universal functions. The sign of \( \alpha \) Special relationships depend on the type of orbit, which is defined as follows: a parabola (\( \alpha = 0 \)), an ellipse (\( \alpha > 0 \)), and a hyperbola (\( \alpha < 0 \)). We have to consider the possible signs of \( \alpha \) and find values for \( Y_0, Y_1, Y_2 and, Y_3 \) for all cases of \( \alpha \). The universal functions parameters are linearly independent [10]. Where \( Y_0, Y_1, Y_2 and, Y_3 \) are universal functions parameters.

3. The new family of transcendental functions

To construct the new family of universal functions, we begin determining the power series solution of Eq. (7) [19]. \( \sigma \) is the root in equation (7), which is solved as follows:-

\[ \sigma = \sum_{k=0}^{\infty} a_k \chi^k \]  

(8)

Differentiating (8) twice with respect to \( \chi \) and substituting into equation (7), we get

\[ \sum_{k=0}^{\infty} (k+2)(k+1)a_{k+2} \chi^k + \alpha \mu \sum_{k=0}^{\infty} a_k \chi^k = 0 \]  

(9)

Equating coefficients of like powers of \( \chi \). We get

\[ a_{k+2} = -\frac{\alpha \mu}{(k+2)(k+1)} a_k \text{ for } k = 0, 1, \ldots \]  

(10)

as a recursion formula for the coefficients. Therefore

\[ \sigma = a_0 \sum_{k=0}^{\infty} (-1)^k \frac{\alpha^k \mu^k}{2k!} \chi^2k + a_1 \sum_{k=0}^{\infty} (-1)^k \frac{\alpha^k \mu^k}{(2k+1)!} \chi^{2k+1} \]  

(11)

where \( a_0 \) and \( a_1 \) are two arbitrary constants. We shall designate the two series expansions by \( U_0(\chi; \mu \alpha) \) and \( U_1(\chi; \mu \alpha) \) so that

\[ \sigma = a_0 U_0(\chi; \mu \alpha) + a_1 U_1(\chi; \mu \alpha) \]  

(12)

The function \( U_1 \) is simply integral of \( U_0 \), so we are motivated to define a sequence of functions [19].
\[ U_1(\chi; \mu \alpha) = \int_0^\chi U_0(\chi; \mu \alpha) d\chi, U_2(\chi; \mu \alpha) \]
\[ = \int_0^\chi U_1(\chi; \mu \alpha) d\chi, U_3(\chi; \mu \alpha) \]
\[ = \int_0^\chi U_2(\chi; \mu \alpha) d\chi \]

After execute the integral we get
\[ \chi U_{n-1}(\chi; \mu \alpha) d\chi = \chi^k \sum_{k=0}^\infty (-1)^k \frac{\alpha^k \mu^k}{(2k)!} \chi^{2k} = U_n(\chi; \mu \alpha) \]

Now we define a new function \[ Y_n(\chi; \alpha) = (\sqrt{\mu})^n U_n(\chi; \mu \alpha). \]
\[ \sigma = a_0 U_0(\chi; \mu \alpha) + a_1 U_1(\chi; \mu \alpha) \]
\[ \sigma = a_0 Y_0(\chi; \alpha) + a_1 Y_1(\chi; \alpha) \]

Where \[ Y_0(\chi; \alpha) = \sum_{k=0}^\infty (-1)^k \frac{\alpha^k \mu^k}{(2k+1)!} \chi^{2k} \]
\[ Y_1(\chi; \alpha) = (\sqrt{\mu} \chi) \sum_{k=0}^\infty (-1)^k \frac{\alpha^k \mu^k}{(2k+1)!} \chi^{2k} \]

For any \( n \) non-negative integer, it is easy to show
\[ Y_n(\chi; \alpha) = (\sqrt{\mu} \chi)^n \sum_{k=0}^\infty (-1)^k \frac{\alpha^k \mu^k}{(2k+1)!} \chi^{2k} \]

By differentiating Eq. (18) we get
\[ \frac{dY_n}{d\chi} = \sqrt{\mu} Y_{n-1}(\chi; \alpha) \] (n > 0)
\[ \frac{dY_0}{d\chi} = -\alpha \sqrt{\mu} Y_1(\chi; \alpha) \]

For \( m > n \), we deduce that
\[ \frac{d^{m+1} Y_n}{d\chi^{m+1}} + \alpha \mu \frac{d^{m-1} Y_n}{d\chi^{m-1}}(Y_n) = 0; \quad n = 0, 1, \ldots, m; \]

4. Universal distance function

The concept of a universal distance function, \( S \), for conic sections is not widely used in calculating the minimum distance between different cone sections. Describing a universal distance function between two arbitrary conic sections (such as ellipses, hyperbolas, or parabolas) can be difficult. One reason for the complexity is that conic sections have different shapes and properties, making it complex to create a single distance function that covers all possible scenarios. Additionally, the distance between two conic sections can be affected by various factors, such as their orientation, size, and eccentricity. Due to the mentioned, we use the Lagrange coefficients method to calculate the universal distance function. The Lagrange coefficients are a set of four scalar functions \( (F_i, G_i, F_t, G_t) \) that allow us to determine the orbit given only the initial position and velocity [19].

The position and velocity vectors are given as follows
\[ \vec{r}_i = F_i \vec{r}_{0i} + G_i \vec{v}_{0i} \]
\[ \vec{v}_i = F_t \vec{r}_{0i} + G_t \vec{v}_{0i} \]
where \( F_i = 1 - \frac{1}{r_{0i}} Y_2(\chi_i; \alpha_i) \)
\[ G_i = \frac{r_{0i}}{\sqrt{\mu}} Y_1(\chi_i; \alpha_i) + \sigma \frac{\vec{v}_{0i}}{\mu} Y_2(\chi_i; \alpha_i) \]
\[ F_{t,i} = -\frac{\sqrt{\mu}}{r_{0i}} Y_1(\chi_i; \alpha_i) \]
\[ G_{t,i} = 1 - \frac{1}{r_{0i}} Y_2(\chi_i; \alpha_i) \]

Getting the first and second derivatives of these coefficients \( F_i, G_i, \)
\[ \frac{\partial F_i}{\partial \chi_i} = -\frac{\sqrt{\mu}}{r_{0i}} Y_1(\chi_i; \alpha_i) \]
\[ i = 1, 2 \]
\[ \frac{\partial G_i}{\partial \chi_i} = r_{0i} Y_0(\chi_i; \alpha_i) + \sigma \frac{\vec{v}_{0i}}{\sqrt{\mu}} Y_1(\chi_i; \alpha_i) \]
\[ i = 1, 2 \]

Similarly, \( \frac{\partial^2 F_i}{\partial \chi_i^2} \) and \( \frac{\partial^2 G_i}{\partial \chi_i^2} \) \( i = 1, 2 \)
\[ \frac{\partial^2 F_i}{\partial \chi_i^2} = -\frac{\mu}{r_{0i}} Y_0(\chi_i; \alpha_i) \]
\[ i = 1, 2 \]
\[ \frac{\partial^2 G_i}{\partial \chi_i^2} = -\frac{\mu}{r_{0i}} \sigma \alpha_i Y_1(\chi_i; \alpha_i) + \sigma \frac{\vec{v}_{0i}}{\mu} Y_0(\chi_i; \alpha_i) \]
\[ i = 1, 2 \]

The minimum distance function can be written as
\[ S = (\Delta \vec{r})^2 = \left( \vec{r}_1 - \vec{r}_2 \right) \cdot \left( \vec{r}_1 - \vec{r}_2 \right) \]

The closest approach technique finds the minimum difference between the two-position vectors \( \vec{r}_1 \) and \( \vec{r}_2 \) of the two bodies.
Substituting equation (22) in equation (27) After reduction, the equation becomes
\[ S = F_1^2 \vec{r}_{01}^2 + F_2^2 \vec{r}_{02}^2 + G_1^2 \vec{v}_{01}^2 + G_2^2 \vec{v}_{02}^2 + A_1 F_1 F_2 + 2A_1 F_1 G_0 + A_2 F_2 G_1 + A_3 F_1 G_2 + 2A_2 F_2 G_2 + G_1 G_2 \left( -2 \vec{v}_{01} \cdot \vec{v}_{02} \right) \]

(28)

where

\[ \sigma_0 = \vec{r}_{01} \cdot \vec{v}_{01} ; \sigma_0 = \vec{r}_{02} \cdot \vec{v}_{02} ; A_1 = \left( -2 \vec{r}_{01} \cdot \vec{r}_{02} \right) ; A_2 = \left( -2 \vec{r}_{02} \cdot \vec{v}_{01} \right) ; A_3 = \left( -2 \vec{r}_{01} \cdot \vec{v}_{02} \right) \]

4.1. Solution of the universal Kepler’s equation

4.1.1. Time constraint

Solving Kepler’s Equation is very difficult with the usual method because the solution of the first point depends on whether it is close or far from an initial guess. Equation (29) represents the universal Kepler’s equation [20]. So to overcome this problem we use reversing series technique. Considerer Kepler equation

\[ \sqrt{\mu} (t - t_0) = r_0 Y_1(\chi; \alpha) + \frac{\sigma_0}{\sqrt{\mu}} Y_2(\chi; \alpha) + Y_3(\chi; \alpha) \]  

(29)

Where \( t_0 \) is epoch time; \( t \) is final time. Inverting Eq (29) for \( \chi \) using Battein’s algorithm, we get

\[ \Delta = r_0 (\chi \sqrt{\mu}) \sum_{k=0}^{\infty} \frac{(-1)^k \left( a \mu \chi^2 \right)^k}{(1+2k)!} \frac{\sigma_0}{\sqrt{\mu}} \chi \sqrt{\mu} \]

\[ \sum_{k=0}^{\infty} \frac{(-1)^k \left( a \mu \chi^2 \right)^k}{(2+2k)!} \left( \chi \sqrt{\mu} \right)^3 \sum_{k=0}^{\infty} \frac{(-1)^k \left( a \mu \chi^2 \right)^k}{(3+2k)!} \]

We get

\[ \chi = \sum_{k=1}^{n} R_k \Delta^k \]  

(30)

where \( \sqrt{\mu} (t-t_0) = \Delta \). The quantities \( R_k \) are given in the Appendices. Formulating equation (30) for both orbits under time constraint \( g(\chi_1, \chi_2) = 0 \).

\[ g(\chi_1, \chi_2) = \frac{r_0}{\sqrt{\mu}} Y_1(\chi_1; \alpha_1) + \frac{\sigma_0}{\mu} Y_2(\chi_1; \alpha_1) + \frac{Y_3}{\sqrt{\mu}} (\chi_1; \alpha_1) \]

\[ - \frac{r_0}{\sqrt{\mu}} Y_1(\chi_2; \alpha_2) + \frac{\sigma_0}{\mu} Y_2(\chi_2; \alpha_2) \]

\[ + \frac{Y_3}{\sqrt{\mu}} (\chi_2; \alpha_2) + (t_01 - t_02) = 0 \]  

(32)

\( \chi_1, \chi_2 \) represent universal anomalies for first orbit and second orbit, respectively.

4.1.2. Minimization process

It is convenient to use the Lagrangian multipliers as a mathematical technique used in optimization problems involving constraints associated with the constrained problem. To include the constraint in the minimization process, the new function to be considered is \( L = S + \lambda g \), where \( \lambda \) is an arbitrary constant, \( S \) is objective function and \( g \) is constraints. The partial derivatives of \( L \) with respect to \( \chi_1 \) and \( \chi_2 \) are equated to zero, that are

\[ \frac{\partial L}{\partial \chi_1} = \frac{\partial S}{\partial \chi_1} + \lambda \frac{\partial g}{\partial \chi_1} \]  

(33)

\[ \frac{\partial L}{\partial \chi_2} = \frac{\partial S}{\partial \chi_2} + \lambda \frac{\partial g}{\partial \chi_2} \]  

(34)

And elimination of \( \lambda \) yields

\[ \frac{\partial S}{\partial \chi_1} \frac{\partial g}{\partial \chi_2} - \frac{\partial S}{\partial \chi_2} \frac{\partial g}{\partial \chi_1} = 0 \]  

(35)

Now, we defined

\[ V(\chi_1, \chi_2) = \frac{\partial S}{\partial \chi_1} \frac{\partial g}{\partial \chi_2} - \frac{\partial S}{\partial \chi_2} \frac{\partial g}{\partial \chi_1} \]

(35)

Equations (32) and (35) form the system of the two nonlinear \( V(\chi_1, \chi_2) = 0 \) & \( g(\chi_1, \chi_2) = 0 \) simultaneous equations (in \( \chi_1, \chi_2 \)) of the constraint minimization technique. Solving equations (32) and (35) by Newton’s iterative method for two simultaneous nonlinear equations. (show Algorithm).

Now, calculate the partial derivatives for the solution of this system can be obtained by differentiation of (32) and (35) as

\[ \frac{\partial g}{\partial \chi_1} = \frac{r_0 Y_1(\chi_1; \alpha_1) + \sigma_0}{\sqrt{\mu}} Y_1(\chi_1; \alpha_01) + Y_2(\chi_1; \alpha_1) \frac{\partial g}{\partial \chi_2} \]

\[ = - \frac{r_0 Y_0(\chi_2; \alpha_2) - \sigma_0}{\sqrt{\mu}} Y_1(\chi_2; \alpha_02) - Y_2(\chi_2; \alpha_2) \]

(36)

\[ \frac{\partial^2 g}{\partial \chi_1^2} = \sigma_0 Y_0(\chi_1; \alpha_1) + (1 - r_0 \alpha_01) \sqrt{\mu} Y_1(\chi_1; \alpha_1) \frac{\partial^2 g}{\partial \chi_2^2} \]

\[ = (1 - r_0 \alpha_02) \sqrt{\mu} Y_1(\chi_2; \alpha_2) - \sigma_0 Y_0(\chi_2; \alpha_2) \]

(37)

\[ \frac{\partial V}{\partial \chi_1} = \frac{\partial S}{\partial \chi_1} \frac{\partial^2 g}{\partial \chi_1^2} + \frac{\partial S}{\partial \chi_2} \frac{\partial^2 g}{\partial \chi_2^2} - \frac{\partial S}{\partial \chi_1} \frac{\partial^2 g}{\partial \chi_2^2} - \frac{\partial S}{\partial \chi_2} \frac{\partial^2 g}{\partial \chi_1^2} \]

(38)
\[
\frac{\partial V}{\partial \chi_2} \frac{\partial S}{\partial \chi_1} + \frac{\partial^2 S}{\partial \chi_2 \partial \chi_1} - \frac{\partial S}{\partial \chi_2} \frac{\partial^2 S}{\partial \chi_1^2} - \frac{\partial^2 S}{\partial \chi_2^2} \]
\]

(39) Jacobian matrix as following

\[
J(\chi_1, \chi_2) = \begin{bmatrix}
\frac{\partial^2 S}{\partial \chi_1^2} & \frac{\partial g}{\partial \chi_1} & \frac{\partial g}{\partial \chi_2} & \frac{\partial^2 S}{\partial \chi_1 \partial \chi_2} + \frac{\partial S}{\partial \chi_1} \frac{\partial^2 g}{\partial \chi_2^2} + \frac{\partial S}{\partial \chi_2} \frac{\partial^2 g}{\partial \chi_1^2} + \frac{\partial^2 S}{\partial \chi_1 \partial \chi_2} \\
\frac{\partial g}{\partial \chi_1} & \frac{\partial^2 S}{\partial \chi_1 \partial \chi_2} & \frac{\partial g}{\partial \chi_2} & \frac{\partial^2 S}{\partial \chi_1 \partial \chi_2} + \frac{\partial S}{\partial \chi_1} \frac{\partial^2 g}{\partial \chi_2^2} + \frac{\partial S}{\partial \chi_2} \frac{\partial^2 g}{\partial \chi_1^2} + \frac{\partial^2 S}{\partial \chi_1 \partial \chi_2} \\
\frac{\partial g}{\partial \chi_2} & \frac{\partial g}{\partial \chi_1} & \frac{\partial^2 S}{\partial \chi_1^2} & \frac{\partial^2 S}{\partial \chi_1^2} + \frac{\partial S}{\partial \chi_1} \frac{\partial^2 g}{\partial \chi_2^2} + \frac{\partial S}{\partial \chi_2} \frac{\partial^2 g}{\partial \chi_1^2} + \frac{\partial^2 S}{\partial \chi_1 \partial \chi_2} \\
\frac{\partial g}{\partial \chi_2} & \frac{\partial g}{\partial \chi_2} & \frac{\partial^2 S}{\partial \chi_1^2} & \frac{\partial^2 S}{\partial \chi_1^2} + \frac{\partial S}{\partial \chi_1} \frac{\partial^2 g}{\partial \chi_2^2} + \frac{\partial S}{\partial \chi_2} \frac{\partial^2 g}{\partial \chi_1^2} + \frac{\partial^2 S}{\partial \chi_1 \partial \chi_2}
\end{bmatrix}
\]

Note \( \frac{\partial^2 g}{\partial \chi_2 \partial \chi_1} = 0 \)

We determined Jacobin parameters (see Appendix).

4.1.3. Bordered hessian matrix for universal formulations of the closest approach

The Bordered Hessian matrix, also known as the augmented Hessian matrix, is a mathematical concept used in optimization theory. It is an extension of the Hessian matrix, which is a square matrix of second-order partial derivatives of a scalar function with respect to its variables. The Bordered Hessian matrix incorporates additional rows and columns to include the first-order partial derivatives, where the mathematical form is \( f : R^n \rightarrow R \), where \( n \) is the number of variables. The bordered Hessian matrix is

\[
H(\chi_1, \chi_2) = \begin{bmatrix}
0 & \frac{\partial g}{\partial \chi_1} & \frac{\partial g}{\partial \chi_2} \\
\frac{\partial g}{\partial \chi_1} & \frac{\partial^2 S}{\partial \chi_1^2} + \frac{\partial^2 S}{\partial \chi_1 \partial \chi_2} + \lambda^2 \frac{\partial^2 S}{\partial \chi_2^2} & \frac{\partial^2 S}{\partial \chi_1 \partial \chi_2} + \lambda^2 \frac{\partial^2 S}{\partial \chi_2^2} + \lambda \frac{\partial g}{\partial \chi_2} \\
\frac{\partial g}{\partial \chi_2} & \frac{\partial^2 S}{\partial \chi_1 \partial \chi_2} + \lambda^2 \frac{\partial^2 S}{\partial \chi_2^2} & \frac{\partial^2 S}{\partial \chi_1^2} + \lambda^2 \frac{\partial^2 S}{\partial \chi_2^2} + \lambda \frac{\partial g}{\partial \chi_1}
\end{bmatrix}
\]

(40)

The Bordered Hessian matrix is used in optimization theory to analyze the critical points of a function and determine their nature, such as whether they are maximum points, minimum points, or saddle points. The Sufficient condition for local extremum is

\[ |H(\chi_1, \chi_2)| > 0 \] is a local maximum,
\[ |H(\chi_1, \chi_2)| < 0 \] is a local minimum.

\[ |H(\chi_1, \chi_2)| = 0 \] is a Saddle point [21].

4.2. Computational developments

In this section, the computational developments of analytic formulas will be considered in detail through the following points.

4.2.1. Newton’s iterative method for two simultaneous nonlinear equations

Consider the two simultaneous nonlinear equation \( V(\chi_1, \chi_2) = 0 \) and \( g(\chi_1, \chi_2) = 0 \) in two unknowns \( \chi_1 \) and \( \chi_2 \) [21].

Input
\( \Delta, r_0, \) and \( \sigma_0 \)

Steps

1. Compute initial guess \( \chi_1 \) and \( \chi_2 \) from equation (31)
2. Compute value \( \frac{\partial V}{\partial \chi_1}, \frac{\partial V}{\partial \chi_2}, \frac{\partial g}{\partial \chi_1}, \frac{\partial g}{\partial \chi_2} \) from equation (38), (39) and (36)
3. Use Cramer’s rule for find \( \Delta \chi_1 \) and \( \Delta \chi_2 \)
4. Apply Newton’s formula

\[
\chi_{1,i+1} = \chi_{1,i} + \Delta \chi_1 \\
\chi_{2,i+1} = \chi_{2,i} + \Delta \chi_2
\]

where

\[
\Delta \chi_1 = -\frac{V(\chi_1, \chi_2) \frac{\partial V}{\partial \chi_1}}{J(\chi_1, \chi_2)} + g(\chi_1, \chi_2) \frac{\partial g}{\partial \chi_1} \\
\Delta \chi_2 = -\frac{g(\chi_1, \chi_2) \frac{\partial g}{\partial \chi_1}}{J(\chi_1, \chi_2)} + V(\chi_1, \chi_2) \frac{\partial V}{\partial \chi_2}
\]

\( J \) is Jacobin matrix
Output
1. Root \( \chi_1, \chi_2 \)
4.2.2. Continued fractions of $Y_j(\chi; \alpha)$, $j = 0, 1, 2, 3$

As is well known, the convergence of the continued fraction expansions is faster and more comprehensive than most of the infinite power series, where the relation $Y(\chi; \alpha)$ contains ‘tan’ and ‘tanh’ functions. For this reason, using a continued fraction is useful [18].

Compute $u$ from the continued fraction

$$u = \frac{a_0}{b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \frac{a_3}{b_3 + \cdots}}}}$$

where

$$a_0 = \frac{1}{2} \chi \sqrt{\mu}$$

$$a_j = -\frac{\alpha \mu \chi^2}{4(4 \chi^2 - 1)} j = 1, 2, 3, \ldots$$

Compute of $Y_j(\chi; \alpha)$, $j = 0, 1, 2, 3$

1. $A = 1 + \alpha \frac{\mu \chi^2}{4}$
2. $Y_0(\chi; \alpha) = \frac{1 - \alpha \frac{\mu \chi^2}{4}}{1}$
3. $Y_1(\chi; \alpha) = \frac{\mu \chi}{A}$
4. $Y_2(\chi; \alpha) = uY_1(\chi; \alpha)$
5. $q = \frac{\alpha \mu \chi^2}{4}$
6. Compute $Y_3(2\chi; \alpha)$ from the continued fraction

$$Y_3(2\chi; \alpha) = \frac{\gamma_0}{1 + \frac{\gamma_1}{1 + \frac{\gamma_2}{1 + \cdots}}}$$

Where $\gamma_0 = \frac{4}{3} Y_1^3(\chi; \alpha)$

$$\gamma_n = \begin{cases} 
\frac{(n+2)(n+5)(n+3)}{(2n+1)(2n+3)} q & n \rightarrow \text{odd} \\
\frac{(n)(n-3)}{(2n+1)(2n+3)} q & n \rightarrow \text{even} 
\end{cases}$$

$$Y_3(\chi; \alpha) = \frac{1}{2} Y_3(2\chi; \alpha) - Y_1(\chi; \alpha) Y_2(\chi; \alpha)$$

5. Results and numerical discussions

In this paper, we have applied the $Y$ function to two types of celestial bodies (artificial and natural objects), also found the minimum distance between two objects. We have validated the results using the S.T.K. packages and investigated local minimum distance using bordered hessian matrix.

Firstly: for artificial satellites, we used data from two satellites: Egyptsat-1 and Cosmos 2228.

Secondly: For natural objects, we use different types of data to study comets (elliptic, parabolic, and hyperbolic). We validated the results of the MATLAB code by comparing them with the Ansys Systems Tool Kit (STK) packages for which the Egypt Space Agency has a license.

5.1. Data source

1. For Egyptsat_1, the data are from the Egyptian Space Agency.
2. The data for the Cosmos 2228 satellite are TLE from the site https://www.space-track.org. Data analysis is used to determine the position and velocity vectors using the SGP4 module. As shown in Table 1.
3. For the comets, we get their data from the site https://ssd.jpl.nasa.gov/?sb_elem#legend. As Shown in Table 8.

5.2. Artificial satellites

Artificial satellites are the first application during which we validate the new sets of universal functions (Y-functions). This involves two stages. The first stage is the trajectory prediction process, and the second is calculating the distance function and then using STK software and the Hessian matrix to validate the results.

5.3. Propagation accuracy for the Y-function

Orbit Propagation is the simulation of a satellite orbit. In other words, it is the determination of the position and velocity vectors at a given moment.

Table 1. Cartesian position and velocity for satellites.

<table>
<thead>
<tr>
<th>Name sat.</th>
<th>Time</th>
<th>X km</th>
<th>Y km</th>
<th>Z km</th>
<th>Vx km/s</th>
<th>Vy km/s</th>
<th>Vz km/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Egyptsat_1</td>
<td>January 14 2018 03:56:15.238</td>
<td>-1679.5750</td>
<td>399.7520</td>
<td>-6820.2544</td>
<td>6.885872</td>
<td>2.59443</td>
<td>-1.541923</td>
</tr>
<tr>
<td>COSMOS 2228</td>
<td>January 14 2018 03:56:15.090</td>
<td>-6965.1925</td>
<td>647.1120</td>
<td>-190.460723</td>
<td>-0.274872</td>
<td>-0.9594</td>
<td>7.476153</td>
</tr>
</tbody>
</table>

Table 2. Classical elements.

<table>
<thead>
<tr>
<th>Semi-major axis</th>
<th>Eccentricity</th>
<th>Inclination (i)</th>
<th>Argument of Perigee</th>
<th>True Anomaly</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) = 7018.95 km</td>
<td>(e) = 0.00235285</td>
<td>(l) = 97.7253°</td>
<td>(P) = 72.5672°</td>
<td>(T) = 185.475°</td>
</tr>
<tr>
<td>Longitude of ascending node = 224.119°</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Continued.
Satellite's motion along the path at time $t_0$ is $r_0$ and $v_0$, and after time $t_1$, it reaches $r_1$, $v_1$; this movement is called Orbit Propagation.

### 5.3.1. Trajectory validation

Table 1 represents the position and velocity vectors for both Egyptsat_1 and COSMOS 2228 satellites at the same epoch time. Now we can predict the orbital motion. We perform satellite propagation using the $(Y)$ functions and the STK comparison. We take a long period for propagation until we know the error rate between the $(Y)$ function and STK. The propagation starts (14 Jan 2018 03:56:15.238) to (5 Feb 2018 03:56:15.238). We used Egyptsat_1 data for propagation. Fig. 1 shows the trajectory of Egyptsat_1 from the $(Y)$ function and STK software, which show the same trajectory.

Table 1. Represents the position and velocity vectors for both Egyptsat_1 and COSMOS 2228 satellites at the same epoch time.

<table>
<thead>
<tr>
<th>Object</th>
<th>Start Time</th>
<th>End Time</th>
<th>Initial Position at start time, km</th>
<th>Initial Velocity at start time, km/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secondary Sat.</td>
<td>999999148</td>
<td>5:51:10.9174126</td>
<td>$1824.166573, 720.369976$, $-6716.937384$</td>
<td>$-7.245236, 0.899602$, $-1.886992$</td>
</tr>
</tbody>
</table>

STK, Ansys Systems Tool Kit.
Fig. 1. Comparison between trajectory Y function and STK. STK, Ansys Systems Tool Kit.

Fig. 2. Position vector according to the Y function.

Fig. 3. Velocity vector according to the Y function.
5.3.2. Validate of position and velocity vectors

Figs. 2 and 3 show propagation for a satellite of position and velocity vectors using Y functions, respectively, during a period of one day from the epoch time in Table 1, due to data overlap in the figure. But the propagation was for 20 days.

Fig. 2 represents the propagation in the satellite Egyptsat_1 position vector for one day according to Y functions using Matlab code. The red color represents the change in the position vector in the \((r_x)\) direction, which oscillates between \((-6645.628 \text{ km} \text{ and } 3543.4136 \text{ km})\) at one revolution and the time for one revolution, 1.62 561 h. The blue color represents the change in the position vector in the \((r_y)\) direction, which oscillates between \((-2213.376 550 \text{ km} \text{ and } 356.051 902 \text{ km})\) at one revolution. Finally, the black color represents the change in the position vector in the \((r_z)\) direction, which oscillates between \((-536.710 879 \text{ km} \text{ and } 6029.653 713 \text{ km})\) at one revolution. The time steps in Figs. 2 and 3 are in seconds.

Fig. 3 represents the propagation in the satellite Egyptsat_1 velocity vector for one day according to Y functions using Matlab code. The red color represents the change in the velocity vector in the \((v_x)\) direction, which oscillates between \((0.204 450 \text{ km/s} \text{ and } 6.043 118 \text{ km/s})\) at one revolution and the time for one revolution, 1.62 561 h. The blue color represents the change in the velocity vector in the \((v_y)\) direction, which oscillates between \((1.137 997 \text{ km/s} \text{ and } 2.613 820 \text{ km/s})\) at one revolution. Finally, the black color represents the change in the position vector in the \((v_z)\) direction, which oscillates between \((-7.439 995 \text{ km/s} \text{ and } 3.701 309 \text{ km/s})\) at one revolution.

Figs. 4 and 5 show the propagation of satellite position and velocity vectors using STK, respectively, over the same period. The black colors represent the change in the position and velocity vector in the \((X, V_X)\) direction, the green colors represent the change in the position and velocity vector in the \((Y, V_Y)\) direction, and finally, the blue colors represent the change in the position and velocity vector in the \((Z, V_Z)\) direction. In addition, time propagation is in hours, as shown in Figs. 4 and 5.

The STK program calculates time steps in seconds and only displays a figure of the time using hours. The x-axis represents the time from 6 a.m. to 6 a.m., which is equal to 86 400 s, which is the same period calculated in Matlab code.

5.3.3. Deviation between Y function and STK

Figs. 6 and 7 represent the absolute value of the Deviation between the position and velocity vectors for Y functions and STK, respectively. We note that the difference between the values of position and velocity oscillates around a very small value.

Minimum and maximum change for the position vector between the \((Y)\) function and STK

\[
\text{Err}_{\text{min}} = |R_Y| - |R_{\text{STK}}| = -7.52234 \times 10^{-7} \text{ km}
\]

\[
\text{Err}_{\text{max}} = |R_Y| - |R_{\text{STK}}| = 7.52234 \times 10^{-7} \text{ km}
\]
5.4. Determination of the minimum distance functions between the two satellites

To determine the minimum distance function between the two objects, we assume that the first satellite of interest, which is possibly close to any other object, is the primary satellite, and the other satellite is the secondary satellite. We studied the minimum distance between the two satellites for six days. Table 3 represents the start and end times.

Using the distance function Eq (28) and under constrain Eq. (32), we define the minimum distance for each day, the initial position, and the velocity vector.

\[
\text{Err}_{\text{max}} = |R_Y| - |R_{\text{STK}}| = 7.34606 \times 10^{-7} \text{ km}
\]

where \( |R_Y| = \sqrt{r_X^2 + r_Y^2 + r_Z^2} \); \( |R_{\text{STK}}| = \sqrt{r_X^2 + r_Y^2 + r_Z^2} \)

The Deviation between Y function and STK for velocity are

\[
\text{Err}_{\text{min}} = |V_Y| - |V_{\text{STK}}| = -7.6045 \times 10^{-7} \text{ km/s}
\]

\[
\text{Err}_{\text{max}} = |V_Y| - |V_{\text{STK}}| = 7.52866 \times 10^{-7} \text{ km/s}
\]

where \( |V_Y| = \sqrt{v_X^2 + v_Y^2 + v_Z^2} \); \( |V_{\text{STK}}| = \sqrt{v_X^2 + v_Y^2 + v_Z^2} \)
Table 4 shows the minimum distance at the initial position, velocity vectors, and anomaly value that achieved the solution.

Fig. 8 shows the change in the distance between the two satellites for 6 days.

The minimum distance over the 6 days is illustrated in a bar chart. We note that the minimum distance was $34.55627044$ km and that the change in the anomaly between the two satellites at this minimum distance was, respectively, $0.080847494$ and $0.081404180$. This change in the anomaly over time is shown in Fig. 9.

Table 4 shows the value of the anomalies that achieved the minimum distance between the primary and secondary satellites, we can now determine the position vector, velocity vector, and time that achieve the minimum distance as in Table 4.

Table 5 shows the position vector, velocity vector, and time vector using Matlab Code and Table 6 shows the position vector, velocity vector, and time using STK.

The minimum distance according to STK, Table 6 determines the distance from STK between two satellites.

The Deviation distance between Y function and STK.

\[
\text{STK} = \sqrt{\text{(Second orbit – First orbit)}^2} = 34.55631 \text{ km}
\]

\[
|Y \text{ Function Distance – STK Distance}| = 34.55627044 – 34.55631 = 3.956E^{-5} \text{ km}.
\]

Section 5 showed the accuracy of the Matlab code comparing with the STK software. We
compared propagation and minimum distance, which are as following:

1. Figs. 2 and 4 show propagation for the position vector of the Y function and STK software. Fig. 7 shows the deviation between the Y function and STK, while the difference is oscillating between \((-7.52234 \times 10^{-7} \text{ and } 7.34606 \times 10^{-7}\) km).

2. Figs. 3 and 5 show propagation for the velocity vector of the Y function and STK software. Fig. 6 shows the deviation between the Y function and STK, while the difference is oscillating between \((-7.6045 \times 10^{-7} \text{ and } 7.52866 \times 10^{-7}\) km).

3. The deviation distance between Y function and STK = 3.956E-05 km. This is the method of verification for Matlab code, while the verification for the minimum distance between two objects we used a bordered hessian matrix (Table 7).

5.5. Bordered hessian matrix

Table 7 demonstrates that the constraint is achieved, where (g) represents the constraint shown in Equation (32) and the absolute value of the border hessian matrix is less than zero, which
means that the distance value (s) is at the local minimum.

5.6. Contour line

The figure is a contour line of equations ((28) and (32)). The contour line demonstrates the data analyzed between the primary satellite and the secondary satellite, which achieves the minimal distance function, which is shown in Fig. 10. The g constraint was achieved, as the objective function S (minimal distance between two orbits) has achieved. As is shown in Fig. 10, the black line is the constraint, and the dot represents the value of the distance function (S) corresponding to the anomalies (c₁, c₂) between a primary satellite and a secondary satellite. Furthermore, S is defined as a local minimum. Finally, the constraint (g) is passing through the minimum distance MOID.

5.7. Natural objects

Table 8 shows data for different types of comets using the position and velocity vectors. We analyzed different cases.

5.8. Two elliptical orbits for two comets

5.8.1. Orbit propagation
From Table 8, we notice that the time for the two objects is different; therefore, we make a propagation process for the older ones until the time between the two bodies is equal. Table 9 shows the new initial conditions.

5.8.2. Determination the minimum distance between two elliptical orbits
Now, we determine the minimum distance between two comets. Table 10 demonstrates the value of anomalies (c₁, c₂) that are solved from equations (32,35), which represent the roots of these equations. These anomaly values are such that the minimum distance between these comets in orbits (elliptical and elliptical Specific in Table 8) is achieved.

5.8.3. Achieving the minimum distance
The minimum distance is achieved at time 11/7/2017 4:21:58.541979. Table 11 shows the time, position, and velocity vectors that achieved the minimum distance.

Table 9. New initial conditions.

<table>
<thead>
<tr>
<th>Object</th>
<th>Initial time</th>
<th>Initial position</th>
<th>Initial velocity</th>
<th>S</th>
<th>Distance Km</th>
</tr>
</thead>
<tbody>
<tr>
<td>158 P/Kowal-LINEAR</td>
<td>June 23 2013 00:00:00.000</td>
<td>506312087.151057556525768.086488142069094.645967</td>
<td>–38.359721 9.405538 3.804410</td>
<td>9.77561yr</td>
<td></td>
</tr>
<tr>
<td>331 P/Gibbs</td>
<td>June 23 2013 00:00:00.000</td>
<td>–1.2256 e⁸ –4.2697 e⁸ –1.32355 e⁸</td>
<td>16.0907 –4.27388 0.509055</td>
<td>5.2096</td>
<td></td>
</tr>
</tbody>
</table>

Table 10. Initial time, position, and velocity at the minimum distance function.

<table>
<thead>
<tr>
<th>Object</th>
<th>Initial Time</th>
<th>Initial position</th>
<th>Initial velocity</th>
<th>χ</th>
<th>Distance Km</th>
</tr>
</thead>
<tbody>
<tr>
<td>158 P/Kowal-LINEAR</td>
<td>04/9/2017 11:07:37.27784</td>
<td>509 008 271.5022, 420364 949.4924, 82748231.451801</td>
<td>–9.136600150684, 10.20243725339, 4.15965265144974</td>
<td>0.011430728</td>
<td>251525623.2</td>
</tr>
</tbody>
</table>

The minimum distance is 251 525 623.2 km, which is equal to 1.6813449442 AU.

Table 11. Final time, position, and velocity according to Y function.

<table>
<thead>
<tr>
<th>Object</th>
<th>Final Time</th>
<th>Final position Km</th>
<th>Final velocity Km/s</th>
</tr>
</thead>
</table>
5.8.4. The contour line for two elliptical comets

Fig. 11 shows contour lines representing the minimum values of the objective function $S (\chi_1, \chi_2)$ subject to a constraint of the form $g (\chi_1, \chi_2) = 0$. The $g$ constraint was achieved, as the objective function $S$ (minimal distance between two orbits) has been achieved. As is clear in Fig. 11, the black line is the constraint, and the dot represents the value of the distance function ($S$) corresponding to the anomalies $(\chi_1, \chi_2)$ between two elliptic orbits. Furthermore, $S$ is defined as a local minimum. Finally, the constraint ($g$) is passing through the minimum distance MOID.

5.9. Elliptical and hyperbolic for two different comets

5.9.1. Orbit propagation

From Table 8, we notice that the time for the two objects is different; therefore, we make a propagation process for the older ones until the time between the two bodies is equal. Table 12 shows the new initial conditions.

5.9.2. Determination the minimum distance between two elliptical and hyperbolic orbits

Now, we determine the minimum distance between the two comets. Table 13 demonstrates the value of anomalies that are solved from equations (32,35), which represent the roots of these equations. These anomaly values are such that the minimum distance between these comets in orbits (elliptical and hyperbolic Table 8) is achieved.

Table 13 shows that the minimum distance is $4.366603927.08439 \text{ km} = 29.18894 \text{ AU}$.

5.9.3. Achieving the minimum distance

The minimum distance is achieved at time 10/3/2013 10:00:16.9357. We calculated the time, position, and velocity vectors that achieved the minimum distance which are shown in Table 14.

5.9.4. The contour line for two elliptical and hyperbolic comets

Fig. 12 shows contour lines representing the minimum values of the objective function $S (\chi_1, \chi_2)$

<table>
<thead>
<tr>
<th>Table 12. New initial conditions.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td><strong>158 P/Kowal-LINEAR</strong></td>
</tr>
<tr>
<td>Initial time</td>
</tr>
<tr>
<td>Initial position</td>
</tr>
<tr>
<td>Initial velocity</td>
</tr>
<tr>
<td>period</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>C/1999H3 (LINEAR)</strong></td>
</tr>
<tr>
<td>Initial time</td>
</tr>
<tr>
<td>Initial position</td>
</tr>
<tr>
<td>Initial velocity</td>
</tr>
<tr>
<td>period</td>
</tr>
</tbody>
</table>
subject to a constraint of the form $g(x_1, x_2) = 0$. The $g$ constraint was achieved, as was the objective function $S$ (minimal distance between two orbits) has been achieved. As is clear in Fig. 12, the black line is the constraint, and the dot represents the value of the distance function ($S$) corresponding to the anomalies ($x_1, x_2$) between elliptical and hyperbolic orbits. Furthermore, $S$ is defined as a local minimum. Finally, the constraint ($g$) is passing through the minimum distance (MOID) (Fig. 12).

5.10. Elliptical and parabolic orbit for two comets

5.10.1. Orbit propagation

From Table 8, we notice that the time for the two objects is different; therefore, we make a propagation process for the older ones until the time between the two bodies is equal. Table 15 shows the new initial conditions.

5.10.2. Determination the minimum distance between two elliptical and parabolic orbit

Now, we determine the minimum distance between two comets. Table 16 demonstrates the value of anomalies that are solved from equations (32,35), which represent the roots of these equations. These anomaly values are such that the minimum distance between these comets in orbits (elliptical and parabolic orbit in Table 8) is achieved.

Table 15 shows that the minimum distance is $4721263231.1$ km $= 31.5597$ AU.

![Fig. 12. Closest approach between Elliptical orbits with parabolic minimum orbit intersection distance.](image-url)
5.10.3. Achieving the minimum distance

The final time, position, and velocity vector to achieve this distance are shown in Table 17.

5.10.4. The contour line for two elliptical and parabolic comets

Fig. 13 shows contour lines representing the minimum values of the objective function $S(\chi_1, \chi_2)$ subject to a constraint of the form $g(\chi_1, \chi_2) = 0$. The $g$ constraint was achieved, as was the objective function $S$ (minimal distance between two orbits) has been achieved. As is clear in Fig. 13, the black line is the constraint, and the dot represents the value of the distance function ($S$) corresponding to the anomalies ($\chi_1, \chi_2$) between elliptical and parabolic orbits. Furthermore, $S$ is defined as a local

---

Table 15. New initial conditions.

<table>
<thead>
<tr>
<th>Object</th>
<th>Initial time</th>
<th>Initial position (Km)</th>
<th>Initial velocity (Km/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>158 P/Kowal-LINEAR</td>
<td>June 23 2013 00:00:00.000</td>
<td>5.97354e8, -2.83637 e8, -1.49943 e8</td>
<td>6.19035, 12.1846, 3.25889</td>
</tr>
<tr>
<td>C/1998 V2 (SOHO)</td>
<td>June 23 2013 00:00:00.000</td>
<td>-895758677.7, 4435313781</td>
<td>-912429939</td>
</tr>
</tbody>
</table>

Table 16. Initial time, position, and velocity vector at the minimum distance function.

<table>
<thead>
<tr>
<th>Object</th>
<th>Initial Time</th>
<th>Initial position Km</th>
<th>Initial velocity Km/s</th>
<th>$\chi$</th>
<th>Distance Km</th>
</tr>
</thead>
<tbody>
<tr>
<td>158 P/Kowal-LINEAR</td>
<td>11/14/2014 1:08:37 .21</td>
<td>495 119 084.6 435 378 89 077 074.506787</td>
<td>-9.48346276, 9.90759734, 4.10032974</td>
<td>-0.0022027</td>
<td>472126323</td>
</tr>
<tr>
<td>C/1998 V2 (SOHO)</td>
<td>11/14/2014 1:08:37 .21</td>
<td>-972 181 396.347656 4 842 303 151.48028</td>
<td>-1.30853207, 6.97942053, 1.43756889</td>
<td>-0.002536</td>
<td></td>
</tr>
</tbody>
</table>

Table 17. The final time, position, and velocity for the $Y$ function.

<table>
<thead>
<tr>
<th>Object</th>
<th>Final Time</th>
<th>Final position Km</th>
<th>Final velocity Km/s</th>
</tr>
</thead>
</table>

Fig. 13. Closest approach between Elliptical orbits with parabolic minimum orbit intersection distance.
minimum. Finally, the constraint (g) is passing through the minimum distance minimum orbit intersection distance (Fig. 13).

6. Bordered hessian matrix

Check of the local minimum for elliptical, hyperbolic, and parabolic is shown in Table 18. This table referees to a comparison between the 3 different cases, for the anomaly and minimum distance values.

7. Conclusions

In this paper, we used a set of modified universal functions to construct the minimum distance function between two celestial objects. We verified our results using two types of celestial objects, the comets and artificial satellites. We found that the minimum distance between two satellites is 34.5563 km. Also, we found that the minimum distance in the two Elliptical, Elliptical and parabolic and elliptical and hyperbolic orbits are 251 525 623.2 km, 4 721 263 231.1 km, and 4 366 603 927.08439 km, respectively. Furthermore, we found that the deviation in the position and velocity vectors using both STK and the MATLAB oscillate between $\frac{0}{C0}$ and $\frac{0}{C0}$, respectively. Also, the deviation is in distance function equals $956 \times 10^{-3}$ km.

Conflicts of interest

There are no conflicts of interest.

Appendix

Appendix A Initial guess

Literal analytical expressions of the $R_k; k = 1, 2, \ldots, 8$

<table>
<thead>
<tr>
<th>Hessian parameter</th>
<th>Elliptical orbits with elliptic</th>
<th>Elliptical orbits with Hyperbolic</th>
<th>Elliptical orbits with parabolic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determinant Hessian matrix</td>
<td>$-1.332206995397 \cdot 10^{10}$</td>
<td>$-2.1694924712927 \cdot 10^{14}$</td>
<td>$-5.44513528655686 \cdot 10^{14}$</td>
</tr>
<tr>
<td>$X_1$</td>
<td>$-0.001001464$</td>
<td>$0.003815086$</td>
<td>$-0.00220277$</td>
</tr>
<tr>
<td>$X_2$</td>
<td>$-0.001500496$</td>
<td>$0.000639759$</td>
<td>$-0.002536$</td>
</tr>
<tr>
<td>$G$ (constraint)</td>
<td>$-13.0738258957863$</td>
<td>$-1.9124656487$</td>
<td>$-11.7657559$</td>
</tr>
<tr>
<td>$\sqrt{S}$ (Distance)</td>
<td>$1.68188609659116$ AU</td>
<td>$29.18894$ AU</td>
<td>$31.5597$ AU</td>
</tr>
</tbody>
</table>

Table 18. Local minimum distance.
\[ R_7 = \frac{1}{5040\sqrt{\mu^0}} \left( 10395 \left( \frac{\sigma_0}{\sqrt{\mu}} \right)^6 + 1575 \left( \frac{\sigma_0}{\sqrt{\mu}} \right)^4 r_0(-11 + 9\alpha r_0) + 63 \left( \frac{\sigma_0}{\sqrt{\mu}} \right)^2 r_0^2(100 - 174\alpha r_0 + 75\alpha^2 r_0^2) + r_0^3 \right) - 280 + 784\alpha r_0 - 729\alpha^2 r_0^2 + 225\alpha^3 r_0^3 \right) \]

\[ R_8 = -\frac{1}{40320\sqrt{\mu^0}} \left( \frac{\sigma_0}{\sqrt{\mu}} \right)^4 r_0 \left( -26 + 21\alpha r_0 \right) + 315 \left( \frac{\sigma_0}{\sqrt{\mu}} \right)^2 \]

\[ r_0^2 \left( 440 - 748\alpha r_0 + 315\alpha^2 r_0^2 \right) + r_0^3 \left( -15400 + 41580\alpha r_0 - 37206\alpha^2 r_0^2 \right) + 11025\alpha^3 r_0^3 \right) \]

**Appendix B: Jacobian and Hessian parameters**

\[ \frac{\partial S}{\partial x_1} = 2F_1 \frac{\partial F_1}{\partial x_1} r_0^2 + 2G_1 \frac{\partial G_1}{\partial x_1} v_0 \]

\[ + F_2 \frac{\partial F_1}{\partial x_1} \left( -2r_0 \cdot r_{02} \right) + F_1 \frac{\partial G_1}{\partial x_1} \left( 2r_0 \cdot v_0 \right) \]

\[ + G_1 \frac{\partial F_1}{\partial x_1} \left( 2r_0 \cdot v_0 \right) + F_2 \frac{\partial G_1}{\partial x_1} \left( -2r_0 \cdot v_0 \right) \]

\[ + G_2 \frac{\partial F_1}{\partial x_1} \left( -2v_0 \cdot v_0 \right) \]

\[ + G_2 \frac{\partial G_1}{\partial x_1} \left( -2v_0 \cdot v_0 \right) \]

\[ \frac{\partial^2 S}{\partial x_1^2} = 2 \frac{\partial F_2}{\partial x_2} \left( \frac{\partial F_1}{\partial x_1} r_{01}^2 \right) + 2F_1 \frac{\partial^2 F_1}{\partial x_1^2} r_{01}^2 \]

\[ + 2 \frac{\partial G_2}{\partial x_2} \left( \frac{\partial G_1}{\partial x_1} v_{01} \right) + 2G_1 \left( \frac{\partial^2 G_1}{\partial x_1^2} v_{01} \right) \]

\[ + F_2 \left( \frac{\partial^2 F_2}{\partial x_1^2} \right) \left( -2r_{01} \cdot r_{02} \right) \]

\[ + 2 \frac{\partial F_1}{\partial x_1} \left( \frac{\partial G_1}{\partial x_1} \right) \left( 2r_{01} \cdot v_{01} \right) \]

\[ + G_1 \left( \frac{\partial^2 F_1}{\partial x_1^2} \right) \left( 2r_{01} \cdot v_{01} \right) \]

\[ + F_2 \left( \frac{\partial^2 G_2}{\partial x_1^2} \right) \left( -2r_{01} \cdot v_{01} \right) \]

\[ + G_2 \left( \frac{\partial^2 G_1}{\partial x_1^2} \right) \left( -2v_{01} \cdot v_{01} \right) \]

\[ + G_2 \left( \frac{\partial^2 G_1}{\partial x_1^2} \right) \left( -2v_{01} \cdot v_{01} \right) \]

\[ \frac{\partial S}{\partial x_2} = 2F_2 \frac{\partial F_2}{\partial x_2} r_{02}^2 + 2G_2 \frac{\partial G_2}{\partial x_2} v_{02}^2 \]

\[ + F_1 \left( \frac{\partial F_2}{\partial x_2} \right) \left( -2r_{02} \cdot r_{01} \right) \]

\[ + G_1 \left( \frac{\partial F_2}{\partial x_2} \right) \left( -2r_{02} \cdot v_{01} \right) \]

\[ + F_2 \left( \frac{\partial G_2}{\partial x_2} \right) \left( -2r_{02} \cdot v_{02} \right) \]

\[ + G_2 \left( \frac{\partial F_2}{\partial x_2} \right) \left( 2r_{02} \cdot v_{02} \right) \]

\[ + F_2 \left( \frac{\partial G_2}{\partial x_2} \right) \left( 2r_{02} \cdot v_{02} \right) \]

\[ + G_1 \left( \frac{\partial^2 G_2}{\partial x_2^2} \right) \left( -2v_{02} \cdot v_{02} \right) \]

\[ + G_2 \left( \frac{\partial^2 G_2}{\partial x_2^2} \right) \left( -2v_{02} \cdot v_{02} \right) \]
\[
\frac{\partial}{\partial x_1} \left( \frac{\partial S}{\partial x_2} \right) = \frac{\partial F_1}{\partial x_1} \left( \frac{\partial F_2}{\partial x_2} \left( -2 \mathbf{r}_{01} \cdot \mathbf{r}_{02} \right) \right)
+ \frac{\partial F_1}{\partial x_1} \left( \frac{\partial F_2}{\partial x_2} \left( -2 \mathbf{r}_{02} \cdot \mathbf{v}_{01} \right) \right)
+ \frac{\partial F_1}{\partial x_1} \left( \frac{\partial G_2}{\partial x_2} \left( -2 \mathbf{r}_{01} \cdot \mathbf{v}_{02} \right) \right)
+ \frac{\partial G_1}{\partial x_1} \left( \frac{\partial F_2}{\partial x_2} \left( -2 \mathbf{r}_{02} \cdot \mathbf{v}_{01} \right) \right)
+ \frac{\partial G_1}{\partial x_1} \left( \frac{\partial G_2}{\partial x_2} \left( -2 \mathbf{v}_{01} \cdot \mathbf{v}_{02} \right) \right)
\]

(\frac{\partial^2 S}{\partial x_1 \partial x_2} = \frac{\partial^2 S}{\partial x_2 \partial x_1})

References