Modeling and Simulation of the Hilda Asteroid Group

a project presented to
The Faculty of the Department of Aerospace Engineering
San José State University

in partial fulfillment of the requirements for the degree
Master of Science in Aerospace Engineering

by

Ian Supelana

May 2018

approved by

Prof. Jeanine Hunter
Faculty Advisor

The Designated Project Advisor(s) Approves the Project Titled

MODELING AND SIMULATION OF THE HILDA ASTEROID GROUP

by

Ian Supelana

APPROVED FOR THE DEPARTMENT OF AEROSPACE ENGINEERING

SAN JOSÉ STATE UNIVERSITY

May 2018

Prof. J. Hunter	Department of Aerospace Engineering	Advisor

1. Introduction

1.1 Motivation

The asteroid belt is an important region in the solar system and has a significant impact on the behavior of major celestial objects around it. Within the asteroid belt are distinct groups of asteroids based either on the composition of the asteroids or their orbital characteristics. One such group is the Hilda asteroid group, unique in the asteroid belt as having a seemingly triangular orbit. Understanding the mechanisms behind such a configuration is important in gaining a complete understanding of the dynamics of the solar system.

1.2 Literature Review

1.2.1 Asteroid Belt

The asteroid belt refers to the region of the solar system between the orbits of Mars and Jupiter where a collection of asteroids has gathered. The asteroids are radially segregated by composition with Tholen spectral class S-type asteroids primarily in the inner belt and C-types in the outer belt. [3] Tholen spectral classes are briefly described in section 1.2.1.2.

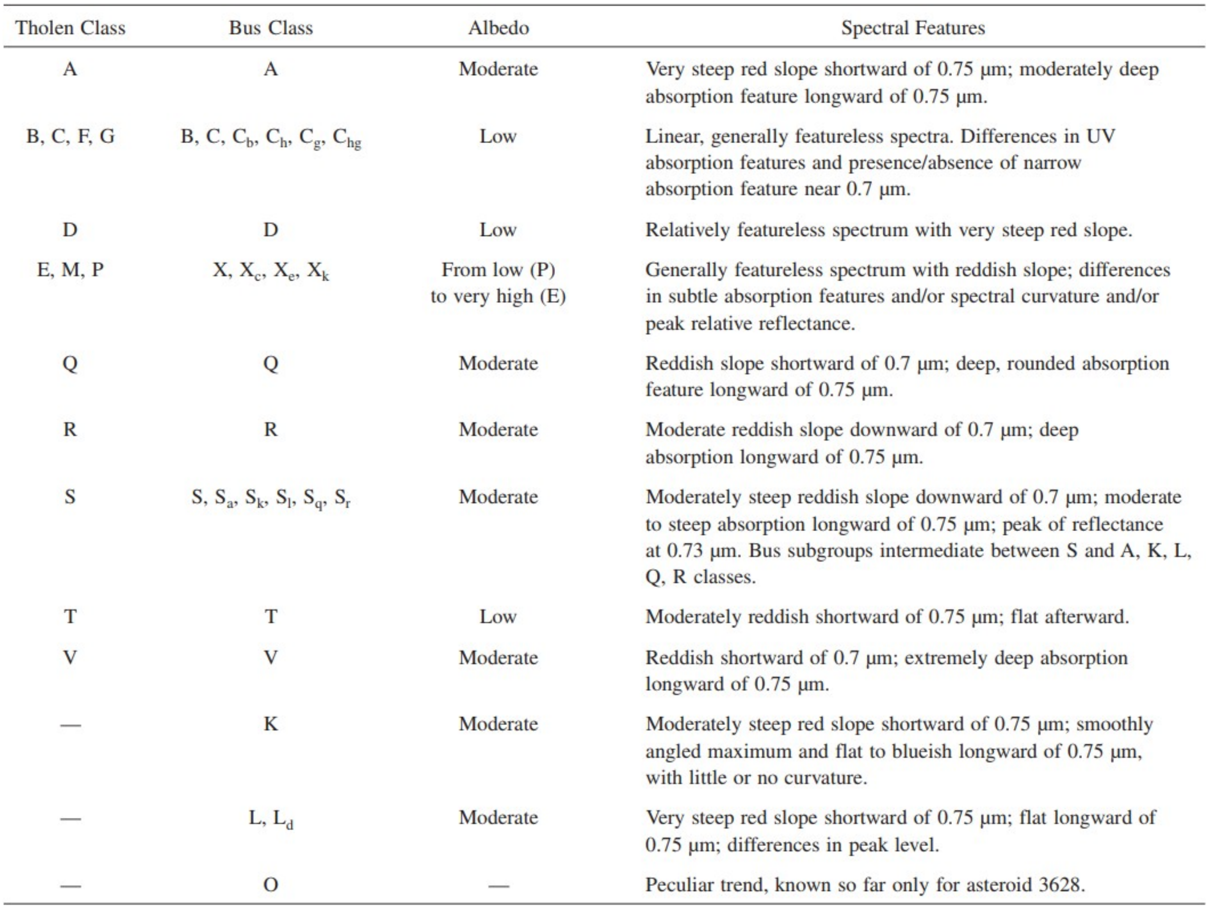
1.2.1.1 Origin

Previous models show that the asteroid belt had a mass that was equal to at least one Earth mass and have just gradually lost mass as time progressed. Currently the asteroid belt contains less than a thousandth of one Earth mass. [3] Most current models explain the loss in mass through various effects including the theory that a portion of the asteroid belt had become part of the inner terrestrial planets and other asteroids had been ejected through dynamic excitation from Jupiter and other planets. [4]

1.2.1.2 Tholen Spectral Taxonomic Classification

The Tholen classification measures the spectral characteristics of asteroids and attempts to classify them based on such measurements. The table below shows the different Tholen classes and the respective features of each one.

Table 1.1 – Summary of Tholen and SMSS Classification [5]



Tholen Class	Bus Class	Albedo	Spectral Features
A	A	Moderate	Very steep red slope shortward of 0.75 μm; moderately deep absorption feature longward of 0.75 μm.
B, C, F, G	B, C, C_b, C_h, C_g, C_{hg}	Low	Linear, generally featureless spectra. Differences in UV absorption features and presence/absence of narrow absorption feature near 0.7 μm.
D	D	Low	Relatively featureless spectrum with very steep red slope.
E, M, P	X, X_c, X_e, X_k	From low (P) to very high (E)	Generally featureless spectrum with reddish slope; differences in subtle absorption features and/or spectral curvature and/or peak relative reflectance.
Q	Q	Moderate	Reddish slope shortward of 0.7 μm; deep, rounded absorption feature longward of 0.75 μm.
R	R	Moderate	Moderate reddish slope downward of 0.7 μm; deep absorption longward of 0.75 μm.
S	S, S_a, S_k, S_l, S_q, S_r	Moderate	Moderately steep reddish slope downward of 0.7 μm; moderate to steep absorption longward of 0.75 μm; peak of reflectance at 0.73 μm. Bus subgroups intermediate between S and A, K, L, Q, R classes.
T	T	Low	Moderately reddish shortward of 0.75 μm; flat afterward.
V	V	Moderate	Reddish shortward of 0.7 μm; extremely deep absorption longward of 0.75 μm.
—	K	Moderate	Moderately steep red slope shortward of 0.75 μm; smoothly angled maximum and flat to blueish longward of 0.75 μm, with little or no curvature.
—	L, L_d	Moderate	Very steep red slope shortward of 0.75 μm; flat longward of 0.75 μm; differences in peak level.
—	O	—	Peculiar trend, known so far only for asteroid 3628.

Tholen classifications are used to approximate standard gravitational parameters for minor celestial objects that do not have an established value. The formula for the approximation is

$$\mu = 6.27 * 10^{-22} * r^3 * T \qquad (1.1)\ [16]$$

where T is the approximate density of the asteroid based the Tholen classification and r is the radius of the asteroid. The values for T are listed below in table 1.2.

Table 1.2 – Approximate density values for Tholen spectral classes [16]

Tholen Class	Density (g/cm^3)
A,E,K,Q,R,S,V	2.4
B,C,D,F,G,P,T	1.8
M,X	5.0

1.2.1.3 Planetary Trojans

1.2.1.3.1 Lagrangian Points

In a system of two orbiting masses, there exists five locations in equilibrium which are referred to as the Lagrangian Points. These points were calculated under the assumption of a restricted three body problem, and as such are only valid when the mass of the third body is much smaller than the other two bodies. An additional stability criteria states that the ratio of the masses between the two large bodies must satisfy the condition

$$M_1 \geq 25M_2\left(\frac{1+\sqrt{1-{}^{4}/_{625}}}{2}\right) \qquad (1.2)\ [1]$$

to have stable L_4 and L_5 points, which are shown below in Figure 1.1. In equation 1.1, M_1 is the mass of the first major body, usually the Sun, and $M_2$is the mass of the secondary body in the system.

The figure below shows the contour plot of the generalized potential with the Lagrangian Points labeled L_1 through L_5.

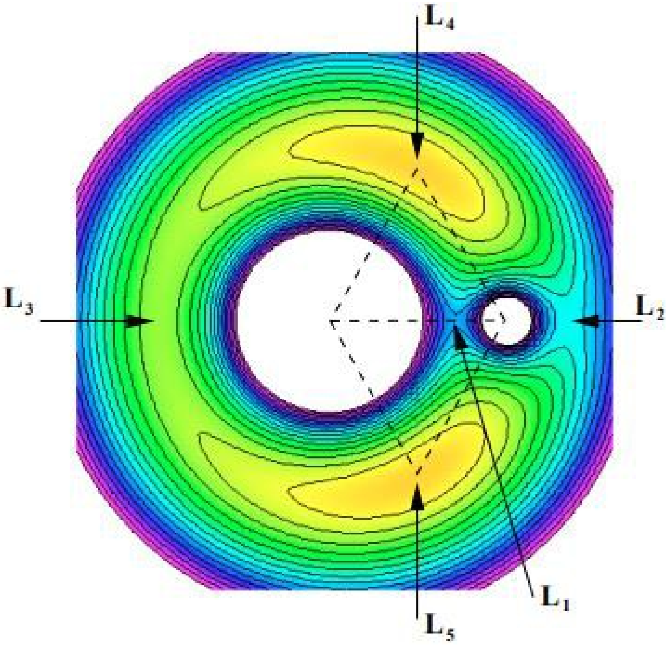


Figure 1.1 – Generalized potential contour plot [1]

L_1, L_2, and $L_3$are points of unstable equilibrium whereas L_4 and $L_5$are stable. While it may seem that L_4 and $L_5$are also unstable due to being “hilltops”, the Coriolis force acts as a restorative force to keep objects within the vicinity of those points. [1] In other words, an object doesn’t stay at a specific point relative to the two main bodies when inside L_4 and L_5, but rather the resulting heliocentric orbit of the object take on characteristics that keeps them within the vicinity of L_4 and L_5. Figure 1.2 below shows an example with the recently discovered 2010 TK_7 asteroid, and asteroid within Earth's L_4 point.

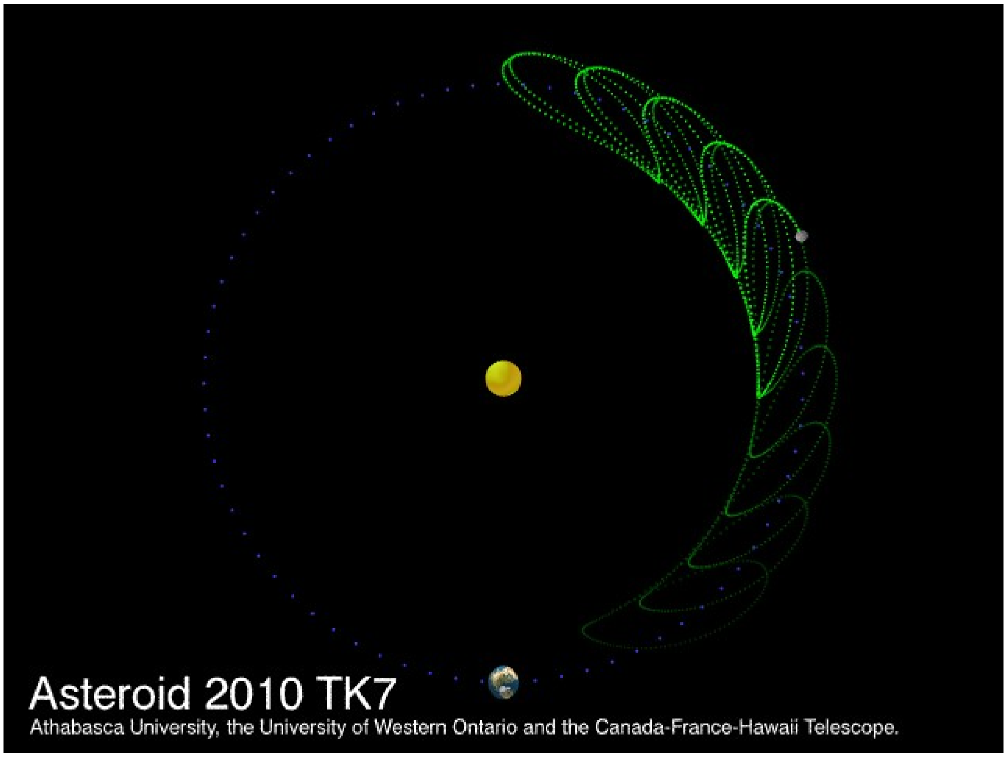

Asteroid 2010 TK7
Athabasca University, the University of Western Ontario and the Canada-France-Hawaii Telescope.

Figure 1.2 – Heliocentric orbit of asteroid 2010 TK_7 [2]

1.2.1.3.2 Jupiter Trojans

Planetary Trojans are asteroids that have been trapped in the L_4 and L_5 points of a planet's orbit. The name comes from 588 Achilles which was one of the first objects discovered to be in L_4. [6] Jupiter Trojans are subdivided into three categories, those in L_4 termed as the "Greek Camp", those in L_5 termed as the "Trojan Camp", and the jumping Trojans known as the Hilda group. Most of the Jupiter Trojans are classified as D-type asteroids with albedo distributions between 3% to 10%. [7][8] Figure 1.3 below depicts the Sun-Jupiter system and the Trojans associated with the system.

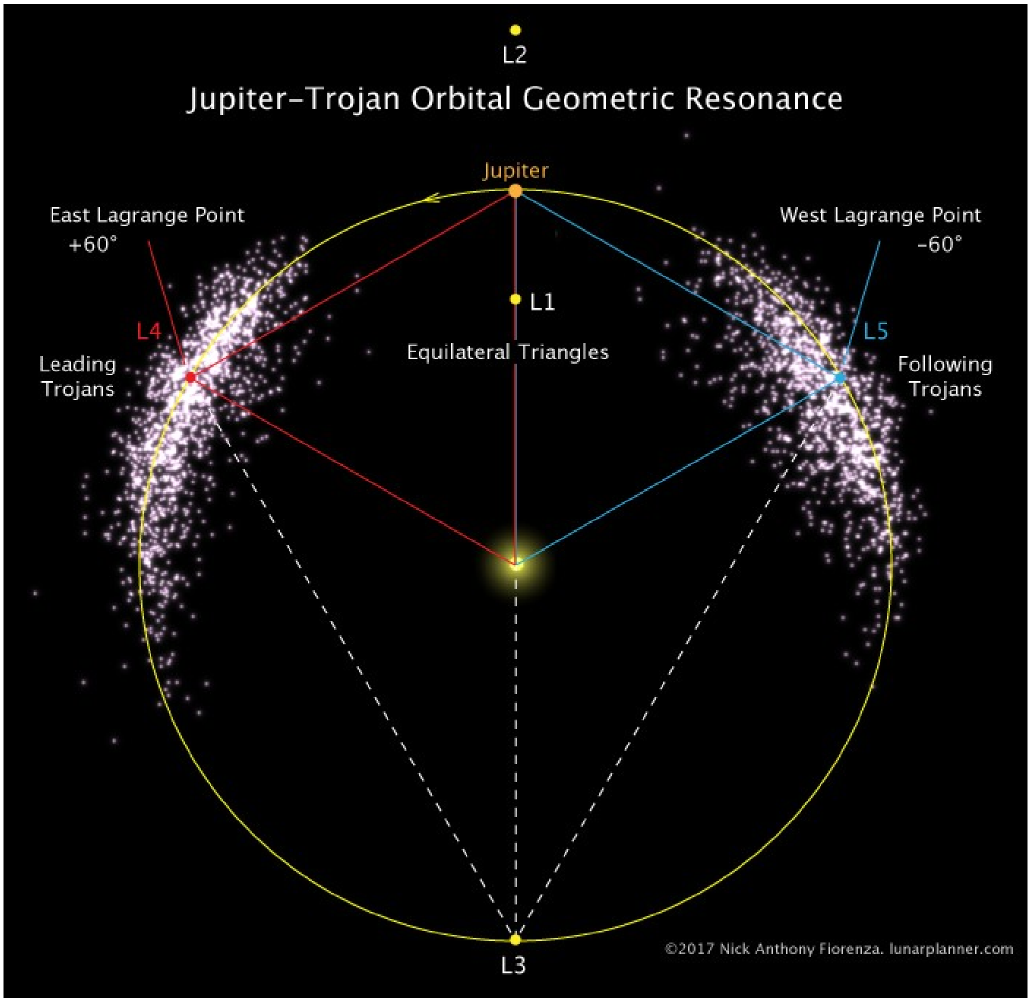


Figure 1.3 – Sun-Jupiter system with associated Trojans [10]

1.2.2 Hilda Asteroid Group

The Hilda asteroid group is not a traditional Trojan as they migrate from one Lagrange point to the next. Asteroids with this orbital pattern are commonly referred to as “Jumping Trojans”. These asteroids typically stay within one Lagrange point for a short while then escape the region, pass by the unstable point $L_{3,}$ and afterwards approach the other stable Lagrange point. For the Hilda group specifically, asteroids approach the three Lagrangian points in order of L_4 - L_3 - L_5 - L_4 and so forth without staying within the points for an extended time. Figure 1.4 illustrate the triangular pattern of the Hilda asteroid group.

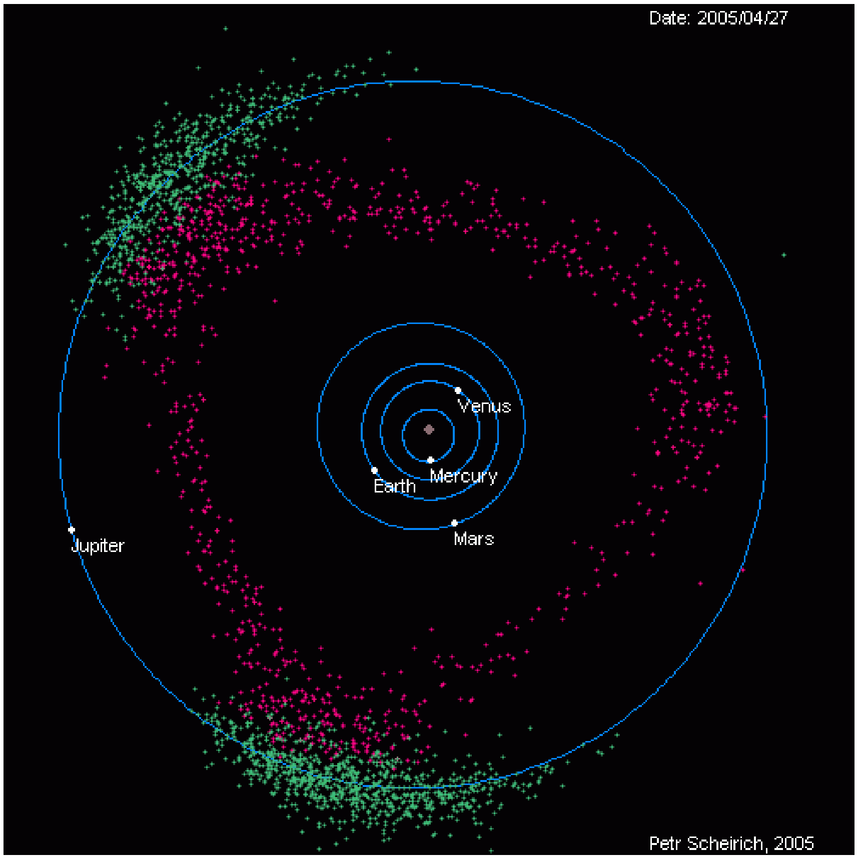


Figure 1.4 – Hilda triangle formation shown in red [11]

The Hilda group is also not a typical asteroid group as the asteroid members do not share a common origin. They are instead a dynamical group which consists of asteroids unrelated in origin but with common orbital characteristics. Those in the Hilda group are in a 3:2 orbital resonance with Jupiter with other orbital characteristics listed below. [12]

Table 1.2 – Summary of Hilda group orbital characteristics [12]

Semi-major axis	3.7AU-4.2AU
Eccentricity	>0.3
Inclination	>20^{o}

1.3 Project Proposal

The objective of this project is to study and model the behavior of Jupiter Trojans, with specific attention to the Hilda group to observe the behavior and mechanisms behind the Hilda triangle formation.

1.4 Methodology

1.4.1 Perturbation Models

1.4.1.1 Cowell's Formulation

1.4.1.1.1 Overview

Cowell's formulation was devised as a method of predicting the perturbed rectangular coordinates by summing up the perturbing gravitational forces present upon a body. Originally devised as a way of checking the path of Comet Halley, it has since been used as a simplified perturbation model of the solar system. [15]

The main advantage to this method is the ease of implementation and speed of calculation, as the equation only involves simple algebraic summation. The major downside is that it is particularly sensitive to large perturbations and require a large number of significant digits to maintain sufficient accuracy.

1.4.1.1.2 Governing Equation

$$\ddot{r}_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{Gm_j(r_j - r_i)}{r_{ij}^3} \qquad (1.3)\ [15]$$

The above equation is the summation of all gravitational forces upon a body. The subscript i represents the body being observed, and j are all the other bodies in the system being considered. Gm_j is simply the gravitational parameter of each perturbing body, r is the rectangular position vector from a chosen reference point, and r_{ij} corresponds to the magnitude of the vector r_j - r_i.

1.4.1.2 JPL Developmental Ephemeris Point Mass Equations

1.4.1.2.1 Overview

Published by the United States Naval Observatory and the United Kingdom Hydrographic Office, the Astronomical Almanac collates information from several sources about astronomical data. The almanac is mainly composed of planetary ephemeris and data from several selected stellar and extragalactic objects. The ephemerides that JPL Horizons uses is also included in this almanac as fundamental ephemerides.

Another resource, the Explanatory Supplement to the Astronomical Almanac, is usually published alongside the Astronomical Almanac, and details among other things the methods and equations used to calculate the final data in the almanac. The point mass equation described below originates from the third edition of the book.

1.4.1.2.2 Governing Equation

$$r_{i_{point\,mass}} = \sum_{j\neq i}\frac{\mu_j(r_j-r_i)}{r_{ij}^3}\Big\{1-\frac{2(\beta+\gamma)}{c^2}\sum_{k\neq i}\frac{\mu_k}{r_{ik}}-\frac{2\beta-1}{c^2}\sum_{k\neq j}\frac{\mu_k}{r_{jk}}+\gamma\left(\frac{v_i}{c}\right)^2+(1+\gamma)\left(\frac{v_j}{c}\right)^2-\frac{2(1+\gamma)}{c^2}\dot{r}_i\cdot\dot{r}_j-\frac{3}{2c^2}\left[\frac{(r_i-r_j)\cdot r_j}{r_{ij}}\right]^2+\frac{1}{2c^2}(r_j-r_i)\cdot\ddot{r}_j\Big\}+\frac{1}{c^2}\sum_{j\neq i}\frac{\mu_j}{r_{ij}^3}\{[r_i-r_j]\cdot[(2+2\gamma)\dot{r}_i-(1+2\gamma)\dot{r}_j]\}(\dot{r}_i-\dot{r}_j)+\frac{(3+4\gamma)}{2c^2}\sum_{j\neq i}\frac{\mu_j\ddot{r}_j}{r_{ij}}+\sum_{m=1}^{3}\frac{\mu_m(r_m-r_i)}{r_{im}^3}+\sum_{c,s,m}F \qquad (1.4)\ [16]$$

The point mass equation considers the perturbing gravitational forces of other bodies as well as the accompanying effects of relativity. The equation was derived by Estrabrook in 1971 from the variation of a time-independent Lagrangian action integral formulated in a non-rotating solar-system-barycentric Cartesian coordinate frame. [16] The variables used are identical to the ones used in equation 2.1. γ represents the amount of space curvature by the unit rest mass of an object, and β represents the amount of nonlinearity in the superposition law for gravity g_{00}. [16] The γ and β parameters are equated as 1 for general relativity. The c variable is the speed of light.

The last two summation terms sum the gravitational perturbations from select asteroids in the asteroid belt. The second to last term sums the three most significant asteroids: Ceres, Pallas, and Vesta. The last term sums the total contribution from 297 other select asteroids classified into three taxonomic classes (C, S, M) depending on the asteroid density. The last summation is only included during calculations for the four inner planets.

The equation is iterative as it depends on the previous acceleration results due to the two acceleration terms present in the right-hand side. The terms are divided by the speed of light however, and as such even simple Newtonian accelerations offer sufficient accuracy.

1.4.2 Numerical Integrators

1.4.2.1 Runge-Kutta Verner 89

Runge-Kutta methods are approximate solutions to differential equations by taking the weighted average of slopes of different intervals. Runge-Kutta Verner 89 is an explicit variable step-size implementation of the basic Runge-Kutta scheme, taking 16 steps to evaluate the next step. The 16 steps and the resulting weighted averages are represented in appendix A as a Butcher tableau. There are two weighted averages for this scheme, an eighth-order and a ninth-order average. The differences between the two are evaluated and taken to be the error of the current step-size. If the error is past a certain set tolerance, then the subsequent step-size is made smaller to reduce the error to within tolerance. Conversely, if the error is too small, the step-size is made larger to reduce the amount of calculations needed to find the solution.

1.4.2.2 MATLAB ODE Suite ODE113

The built-in ODE113 function is based on an Adams-Bashforth-Moulton variable order variable step-size numerical integration method. The implementation is derived from the ODE/STEP, INTRP ODE solver written by Shampine and Gordon in FORTRAN90. [17] The solver contains a family of formulas of orders from 1-12 and relies on both relative tolerance and absolute tolerance to determine order and step-size. Relative tolerance, the error between two values, is better suited to large values and is also what RKV-89 uses for error evaluations. Absolute tolerance is used when values are close to zero as relative tolerance would quickly approach infinity in that situation.

2. Solar System Model Method I – Cowell's Formulation

2.1 Simulation Setup

The above method is numerically integrated using an RKV-89 scheme. All the constants are taken either from the Explanatory Supplement to the Astronomical Almanac or from JPL Horizon. The integration tolerance used is 10^{-12}. Figure 2.10 is simulated with a fixed step RKV 9 scheme and left to propagate for nine years. Each time step for that simulation consists of one hour.

As of the current state of the simulation program for this run, the goal is simply to propagate bodies when given initial state vectors and confirm the model matches with data provided by JPL Horizons.

2.2 Results

The following results represent the simulation method error compared to DE430, the set of ephemerides that the JPL Horizons system currently uses. The error is calculated in blocks of three months from February 02, 2006 00:00:00 TDB for one calendar year. The errors shown in the plots below are ICRF position components. The reference frame used is ICRF with the origin located at the sun.

The full set of rectangular coordinates generated every three months by the model and their DE430 counterparts are included in appendix C and D.

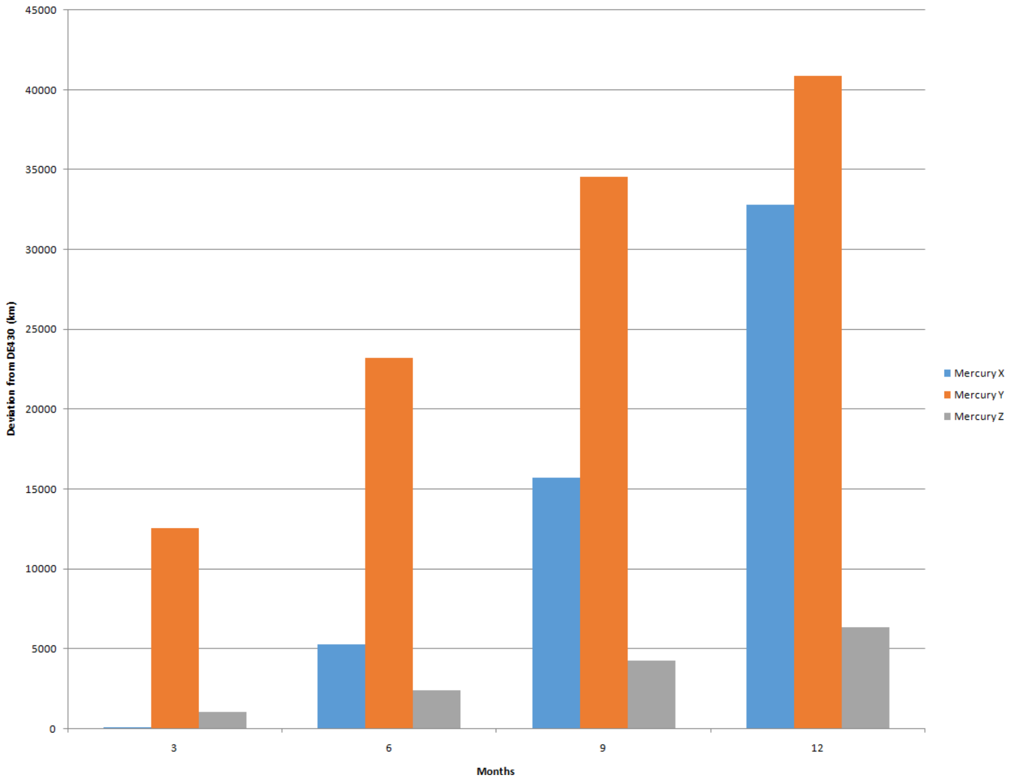


Figure 2.1 – Deviation results for Mercury

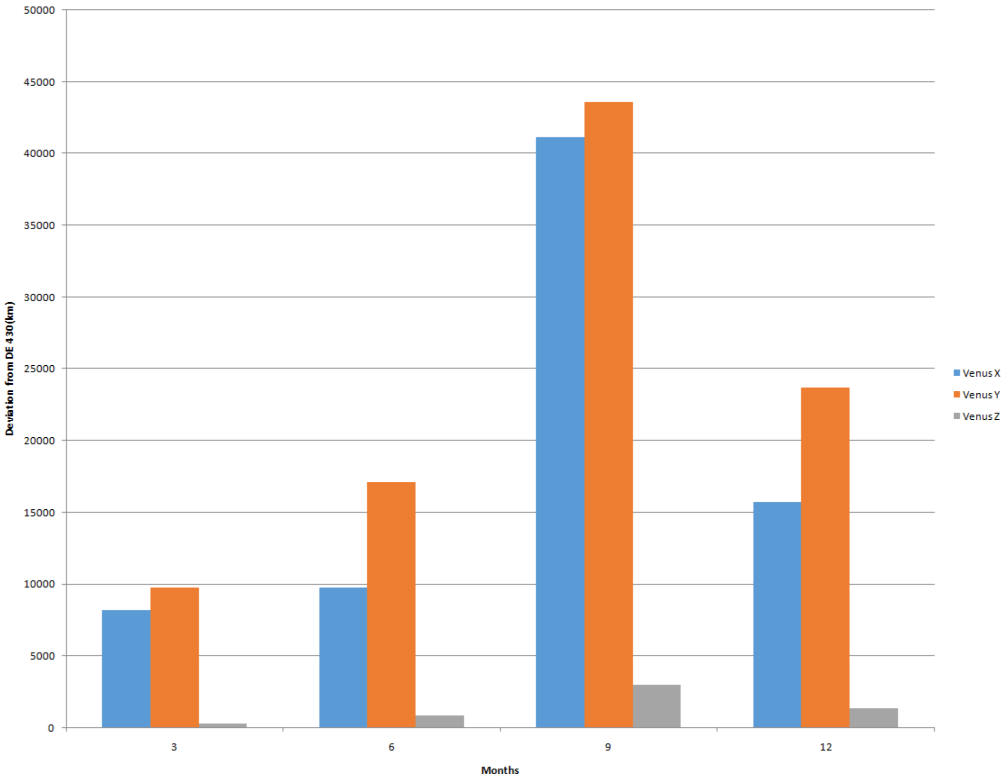


Figure 2.2 – Deviation results for Venus

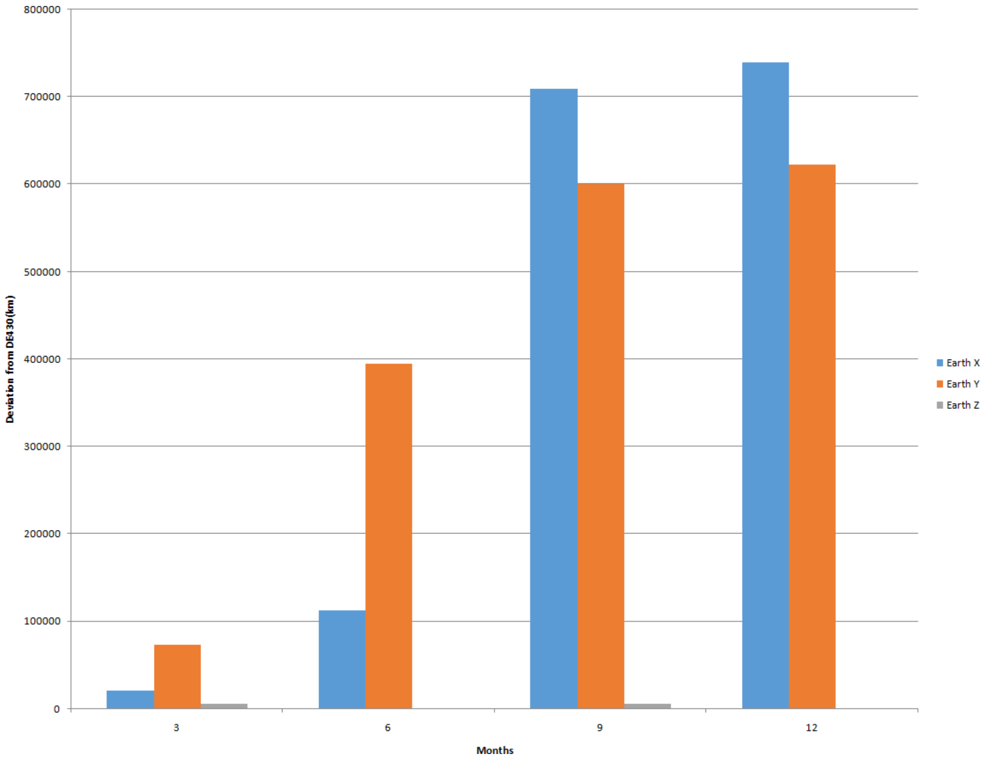


Figure 2.3 – Deviation results for Earth

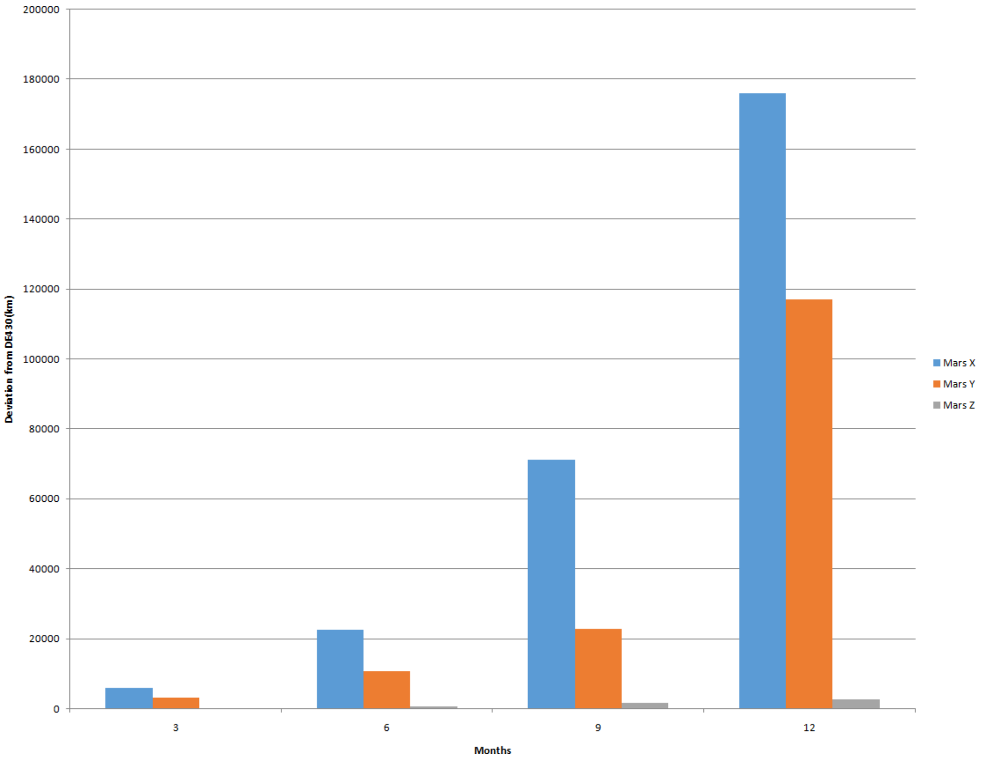


Figure 2.4 – Deviation results for Mars

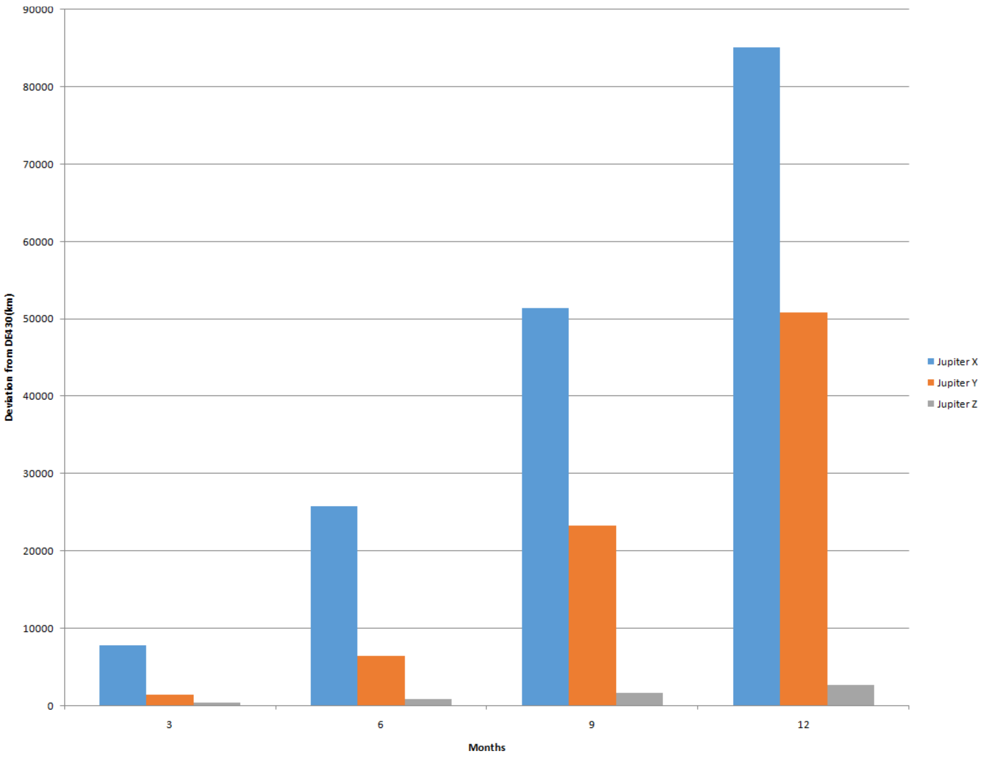


Figure 2.5 – Deviation results for Jupiter

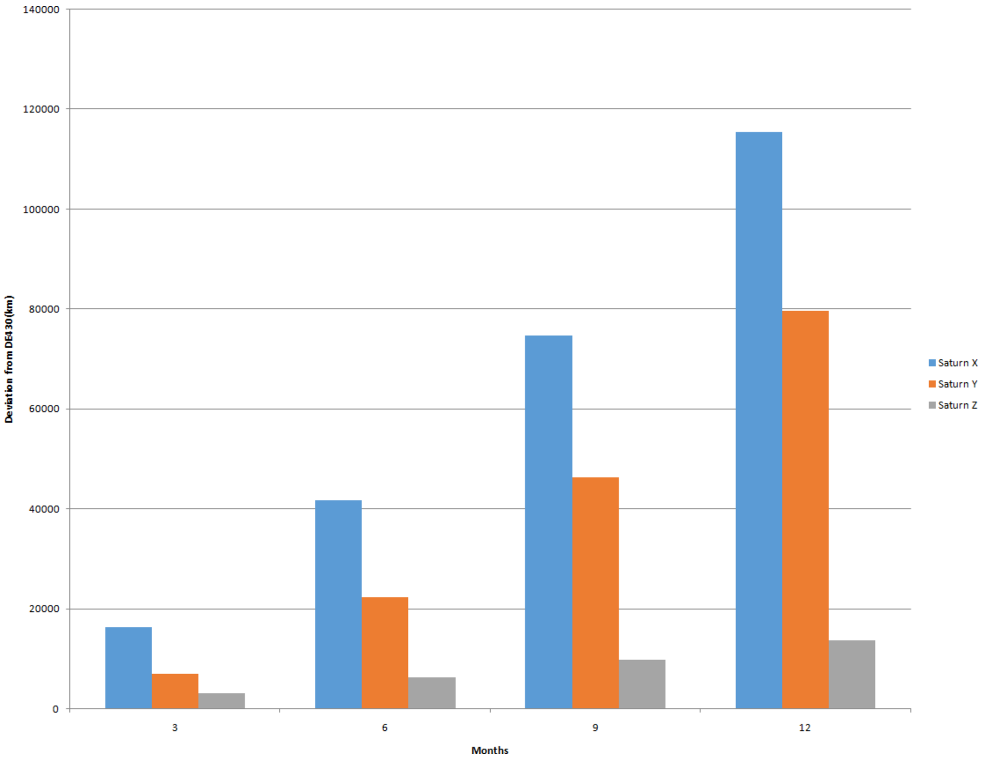


Figure 2.6 – Deviation results for Saturn

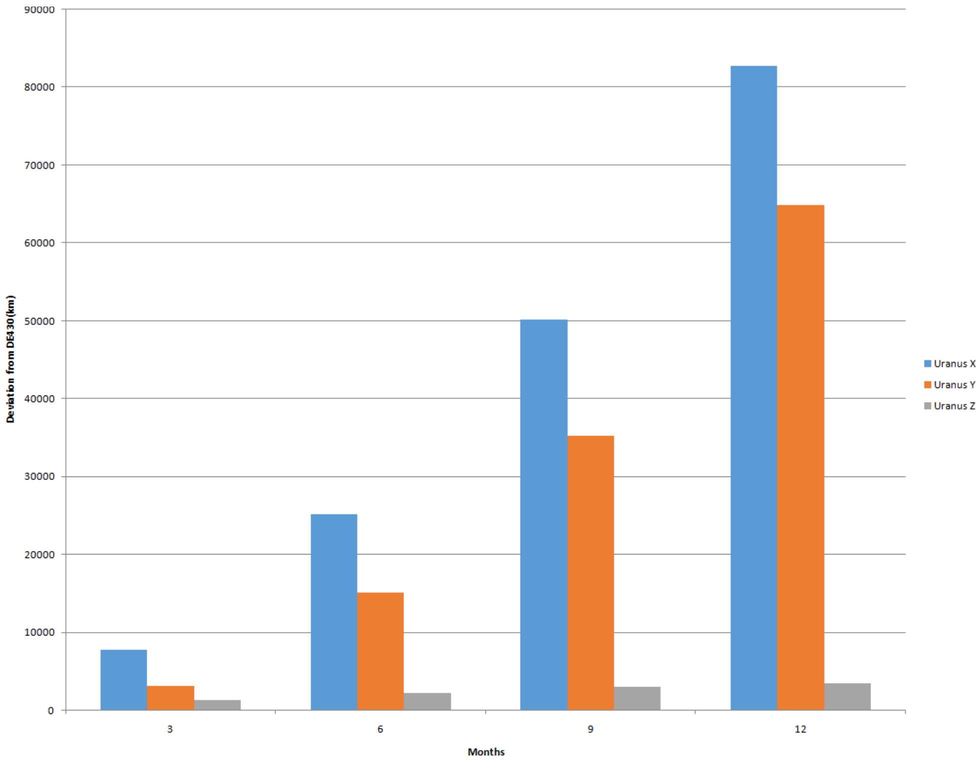


Figure 2.7 – Deviation results for Uranus

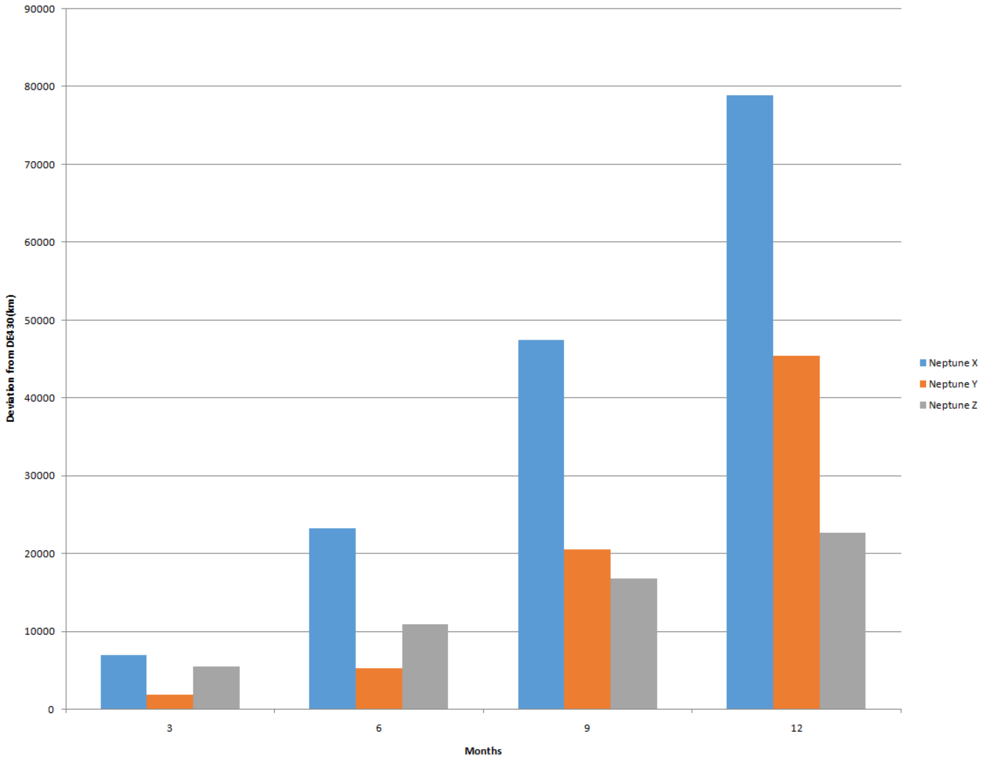


Figure 2.8 – Deviation results for Neptune

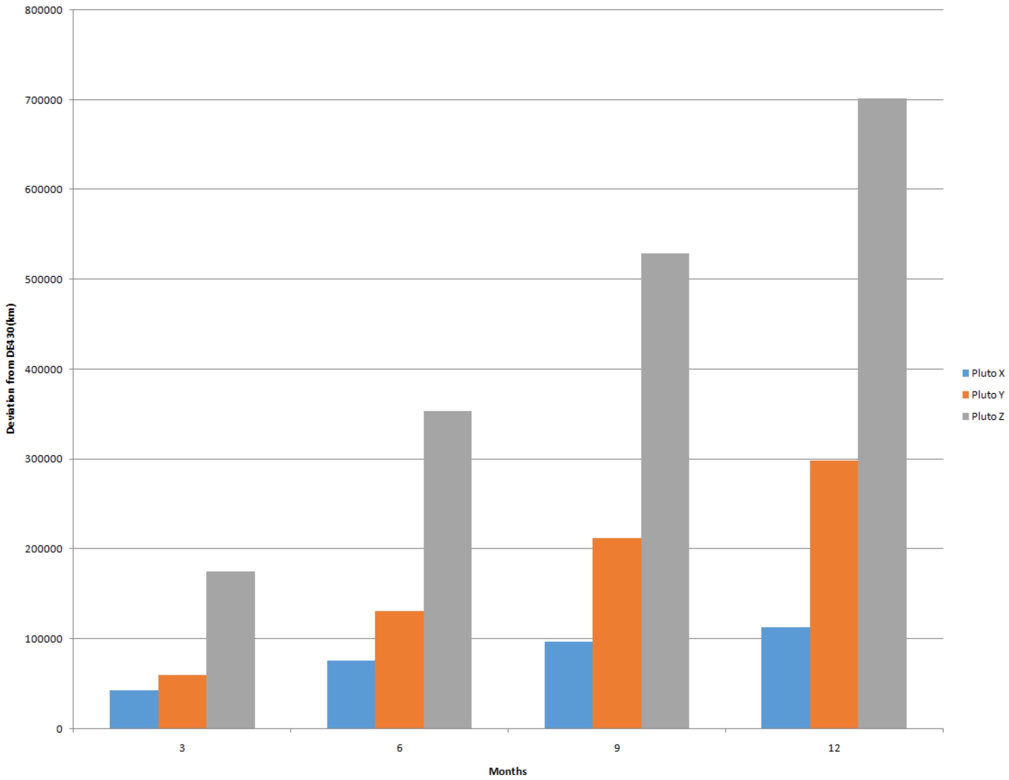


Figure 2.9 – Deviation results for Pluto

2.3 Discussion

The results show the error growth trend of the simulation method after one year. On average, the error growth for the X component is 17738km/month, 14559km/month for the Y component, and 6985km/month for the Z component. Such rates are too high to be used for trajectory design, as the deviations would grow to nearly two million kilometers for both X and Y directions in under ten years of propagation.

Another interesting note in the results is that the rate of growth of the Z direction becomes significantly worse the farther the observed body is from the sun. Neptune's Z component error had a final value of over 20,000km, significantly higher than any other Z component errors from the bodies closer to the sun. Following such a trend, Pluto's Z component is 700,000km off from DE430. The high inclination of the orbit of Pluto and its position as an object within the Kuiper belt may be a contributing factor to this. Likewise, Neptune would also be the object most significantly affected by the high inclination of Kuiper belt objects, leading to higher Z component errors.

2.4 Recommendation

The clearest path forward to improving this method is to include more perturbing objects. However, that comes with the demerit of longer computational times. The numerical integrator also requires more work, as the current implementation requires a runtime of approximately two to three hours to complete a one-year propagation at the current tolerance of 10^{-12}. Other implementations of the same scheme can complete this simulation at tighter tolerances with faster runtimes.

3. Solar System Model Method II – JPL Developmental Ephemeris Point Mass Equations

3.1 Simulation Setup

The simulation setup is identical to the setup for Cowell's formulation. The same integration method, constants, integration limits and tolerances are used. The initial acceleration values used for the first time-step are generated from Cowell's formulation results for the same initial time step. Subsequent time steps use the acceleration from previous time steps.

The last term summing contribution from 297 asteroids are not included. The Earth is also modeled as a single object instead of being modeled as the Earth-Moon system.

Currently, the model is setup to perform the point mass equations on celestial bodies but revert to Cowell's formulation for New Horizons and the three major asteroids. According to the Explanatory Supplement to the Astronomical Almanac, performing simple calculations for the asteroids produce sufficient accuracy.

3.2 Results

The results below are formatted the same as results from section 2.2. The X, Y and Z position deviations have been calculated for each body. The solar system barycenter is the origin of the positional coordinates. The graphs below are split into two halves, the left half of the graph represents the error from the JPL point mass equations and the right half represents the error from Cowell's formulation. The full positional ephemeris results and the DE430 counterparts are included in appendix E and F.

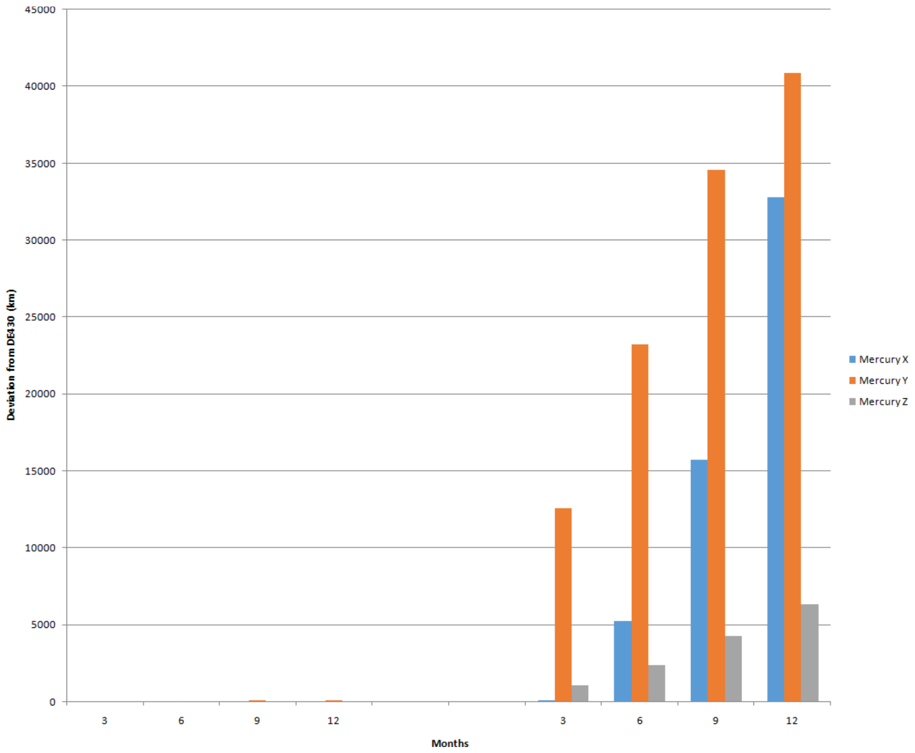


Figure 3.1 – Deviation results for Mercury

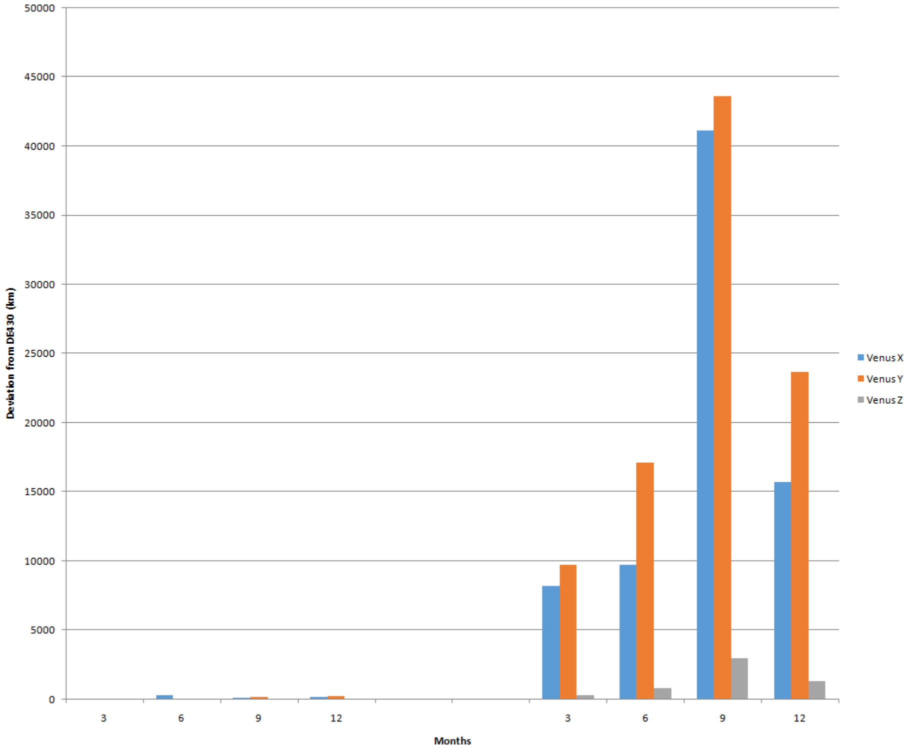


Figure 3.2 – Deviation results for Venus

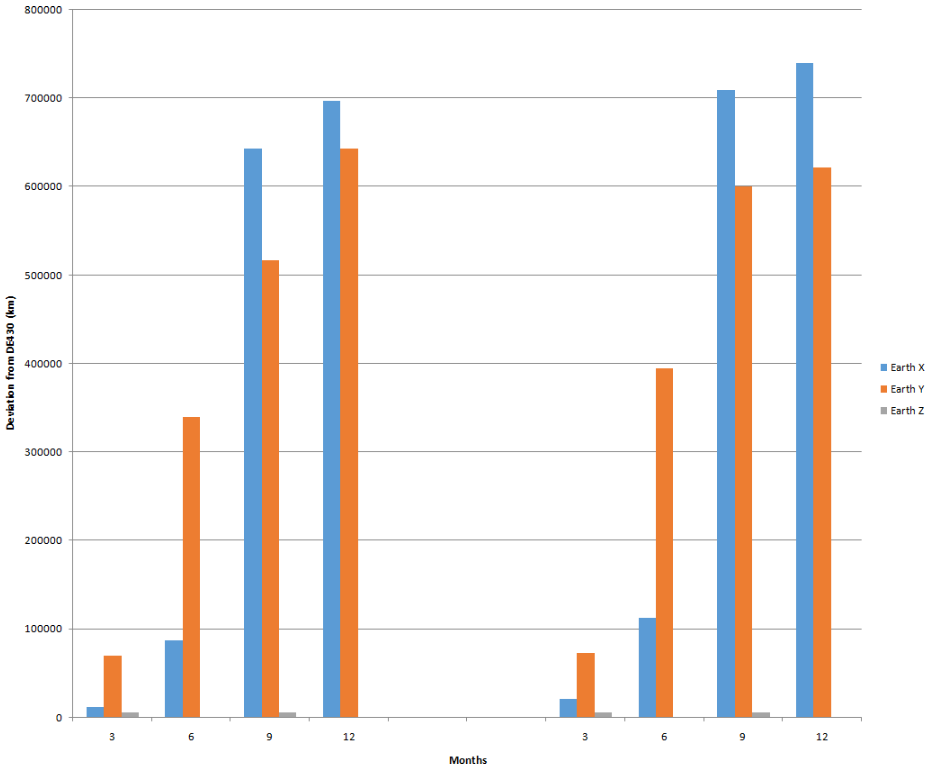


Figure 3.3 – Deviation results for Earth

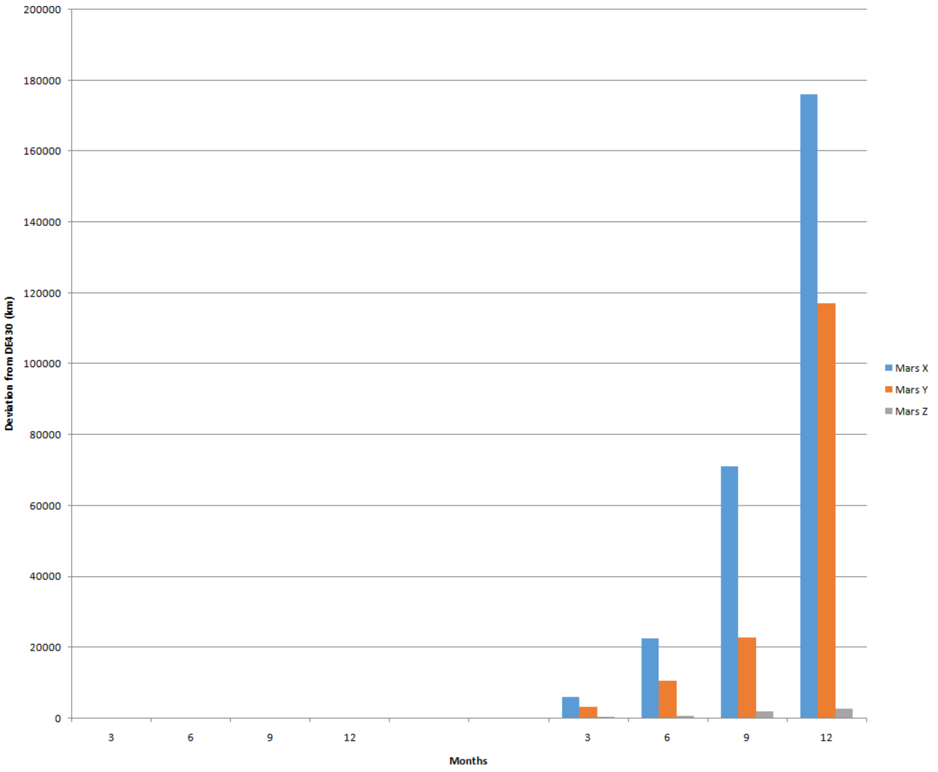


Figure 3.4 – Deviation results for Mars

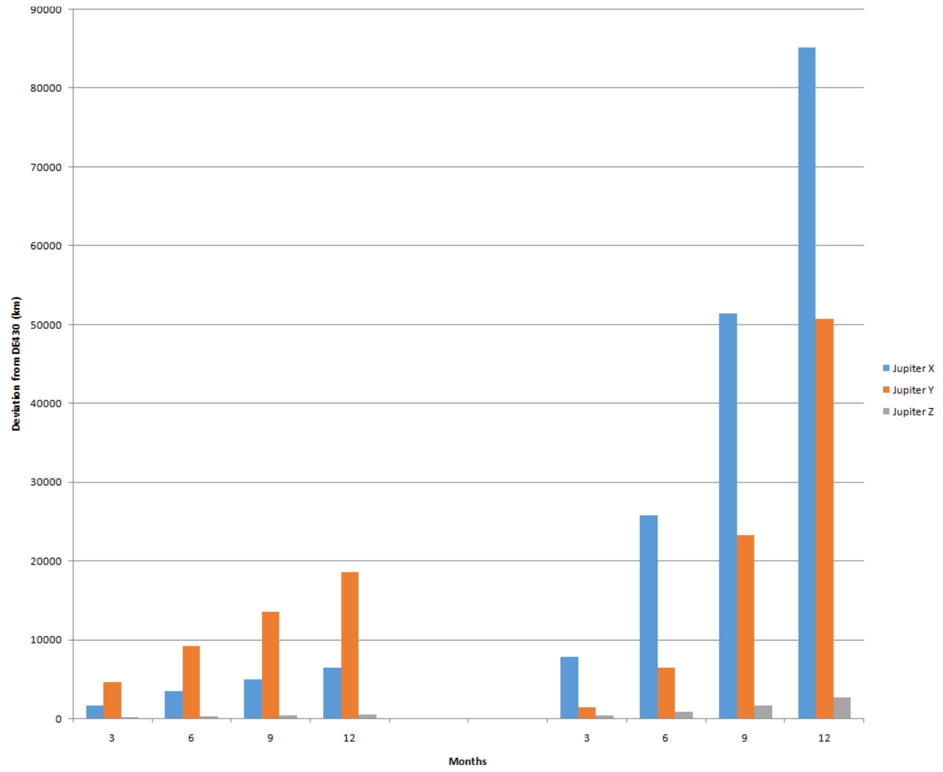


Figure 3.5 – Deviation results for Jupiter

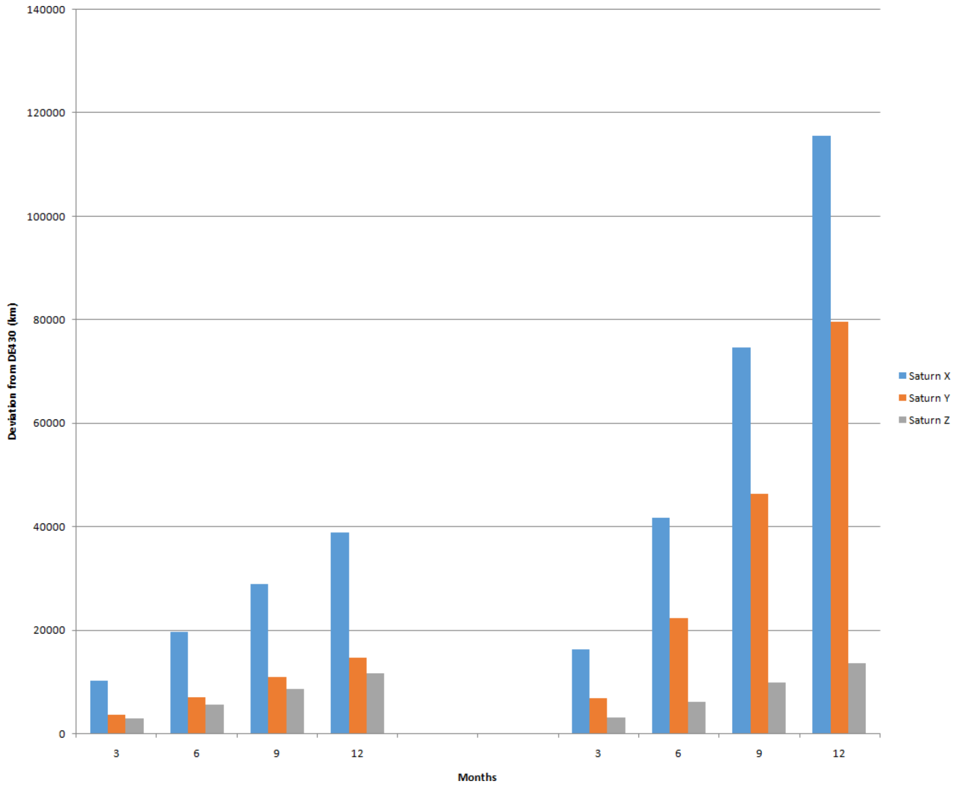


Figure 3.6 – Deviation results for Saturn

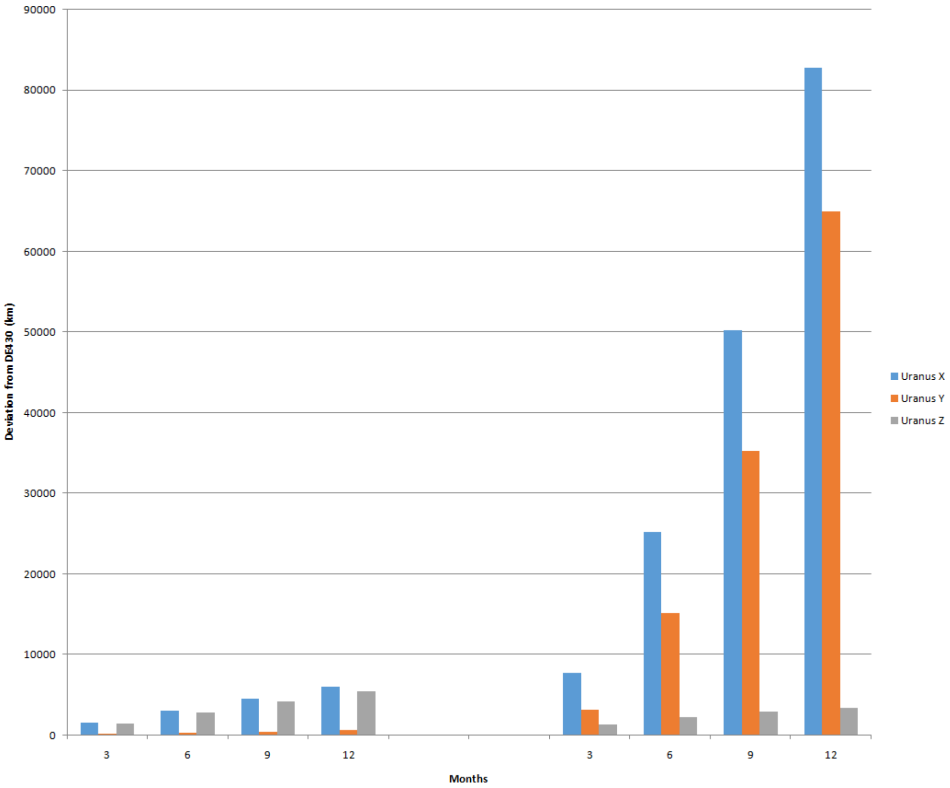


Figure 3.7 – Deviation results for Uranus

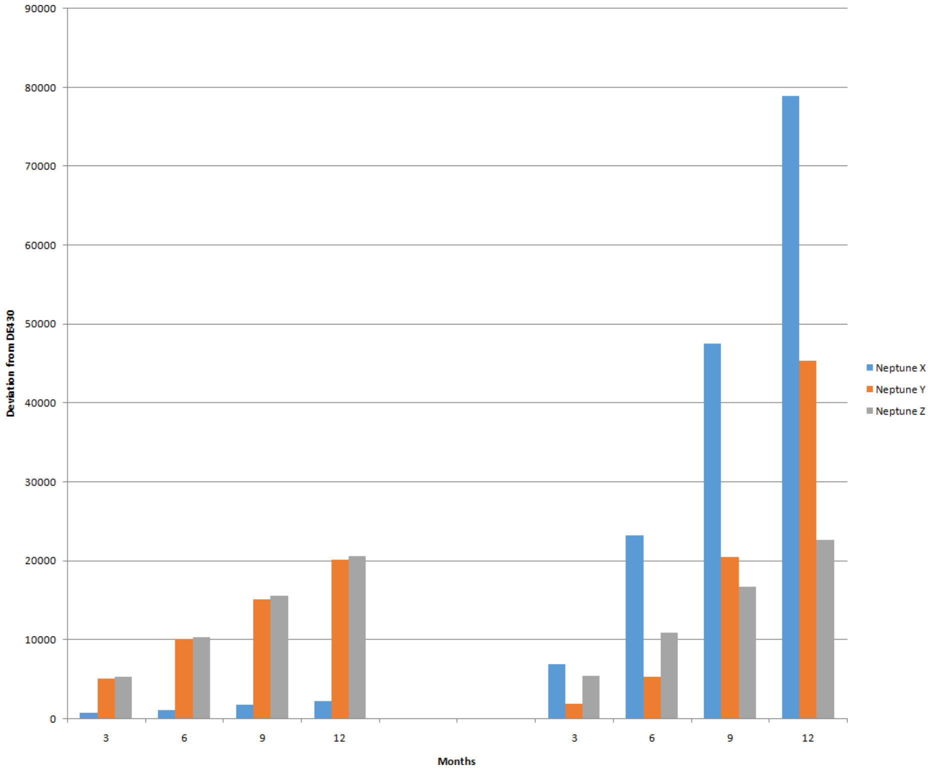


Figure 3.8 – Deviation results for Neptune

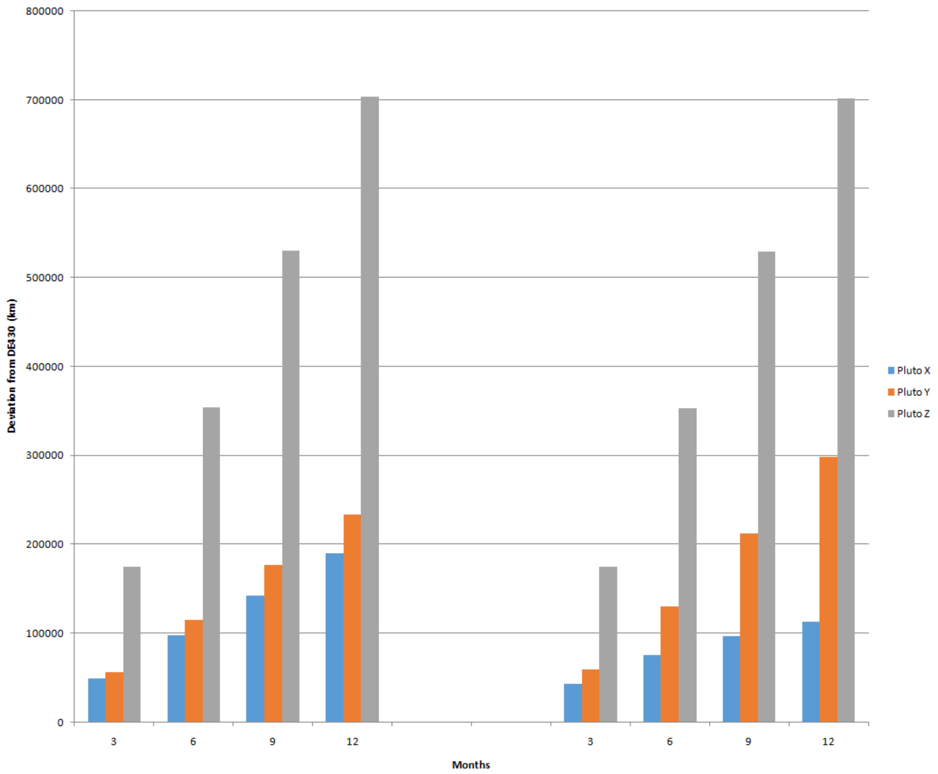


Figure 3.9 – Deviation results for Pluto

3.3 Discussion

The results show a marked improvement in comparison to the results from Cowell's formulation. The inner planets, apart from Earth, have an average error growth of 6.301km/month on the X component, 8.627 km/month on the Y component, and 0.328 km/month on the Z component. The outer planets have worse error growth rates, but still well under what Cowell's formulation produced, at 1097 km/month on the X component, 1124 km/month on the Y component, and 792 km/month on the Z component. With such error rates, the largest error after nine years of propagation would be 931km for inner planets and 120,000 km for outer planets, both on the Y component.

Earth and Pluto are outliers however, as the error growth rates for both bodies mirror those produced by Cowell's formulation. Pluto may require additional refinement due to its nature as a Kuiper belt object as none of the models produce a satisfactory result. Earth on the other hand appears to require being modeled as the Earth-Moon system, as that is the only celestial body within the solar system with a satellite of significant mass compared to its own.

3.4 Recommendation

As mentioned before, Earth must be modeled as a system with the Moon. In the Explanatory Supplement to the Astronomical Almanac, extensive detail is shown in modeling the system, and the explanation for the point mass equation details that the system is treated as one point. The calculations within the point mass equation take the Earth-Moon barycenter as the origin of the point mass, and the resulting positions of the Earth and Moon are derived from it.

Pluto requires additional study to determine the most effective model without having to resort to adding extraneous bodies. The point mass equations may not be suitable for Pluto as those are designed for large bodies in the solar system, and the irregular nature of the orbit of Pluto may require additional factors to model accurately.

4. Simulation Comparisons Between RKV-89 and ODE113

4.1 Simulation Setup

Results taken from Chapter 3 will be compared to results generated by MATLAB's built-in ODE113 function. The same initial vectors with a propagation length of 1 year is used. Relative and absolute tolerances from 10E-10 up to 10E-17 will be used for ODE113 and be compared to previous generated results.

4.2 Results

Table 4.1 – Comparison of results between ODE113 and RKV89

	RKV89	ODE113
Run-time	40 minutes (10E-12)	12 seconds (10E-10)
		17 seconds (10E-12)
		35 seconds (10E-15)
		49 seconds (10E-17)
Data points	313832 (10E-12)	540 (10E-10)
		771 (10E-12)
		1474 (10E-15)
		2086 (10E-17)
Deviation from Mars DE430 ephemerides (X-direction)	110.879999998957 km	1630266.02724496 km (10E-10)
		1630266.02724146 km (10E-12)
		1630266.02725696 km (10E-15)
		1630266.02722846 km (10E-17)

The resulting list of ephemerides are in Appendix F.

4.3 Discussion

Matlab's ODE113 is more efficient compared to the RKV89 integrator used in the previous chapters, however it comes at the expense of accuracy. The accuracy gains in between tolerances of 10E-10 to 10E-17 is miniscule, while the increases in run time are more significant. RKV89, while extremely inefficient due to the number of steps it takes to finish the propagation, is extremely accurate, deviating only 110 km from DE430 compared to more than 1.5 million km from ODE113.

The orders used by ODE113 in the propagation might be lower than eighth or ninth order, which would help to explain the high error despite the equations of motion and initial vectors being identical between the two. However, the small number of evaluations suggest that is not the case, as that would result in higher errors that ODE113 would have to compensate for by decreasing the step-size. ODE113 takes 99% less steps to finish converging to a solution despite being set at a much higher tolerance than RKV89, suggesting that a large step-size is used. This may be the reason for the large error, and further investigation into how it evaluates and weights relative error would be necessary to remedy the issue. The negligible increases in accuracy between the different tolerances also points to a possible issue with error evaluation.

5. Solar System Model with Hilda Trojans

5.1 Simulation Setup

The model consists of the same objects as previous simulations described so far with the inclusion of 21 Hilda asteroids. The model is propagated for 50 years using ODE113 with a relative and absolute tolerance of 10E-17 to expedite results. The chosen Hilda asteroids have Tholen classifications, allowing for an approximation of their standard gravitational parameter. There are numerous other Hilda asteroids, but those asteroids do not have enough information to be used with the established models.

5.2 Results

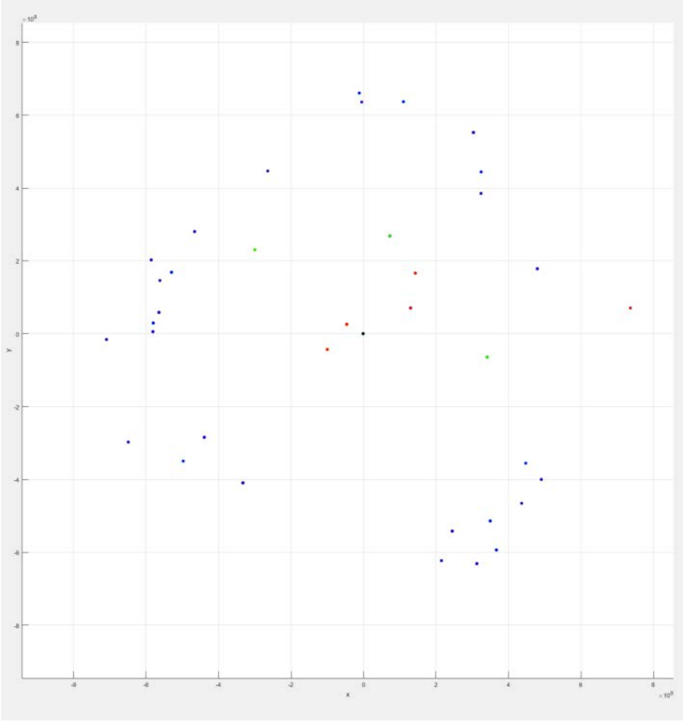


Figure 5.1 – XY-Plot of Hilda asteroids (blue) with Jupiter and the inner planets (red), the sun (black), and Pallas, Ceres and Vesta (green)

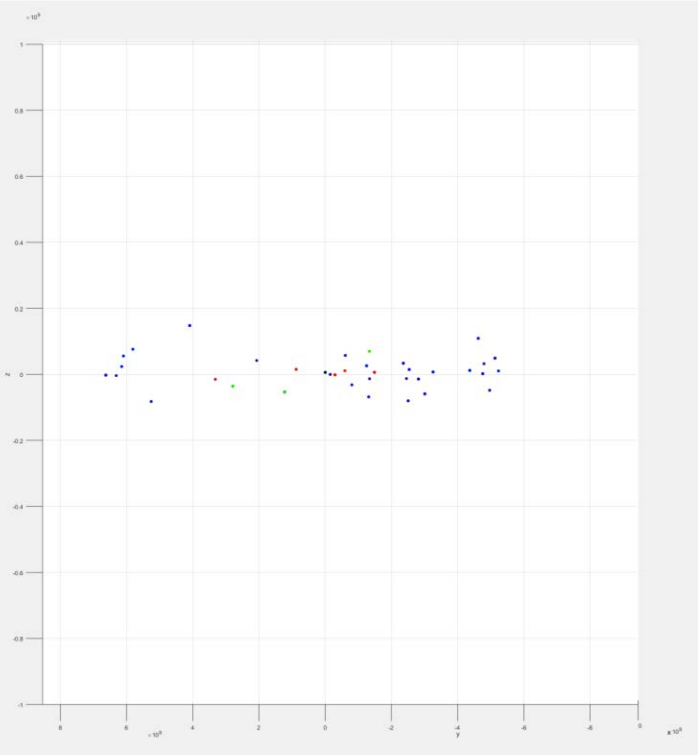


Figure 5.2 – YZ-Plot of Hilda asteroids (blue) with Jupiter and the inner planets (red), the sun (black), and Pallas, Ceres and Vesta (green)

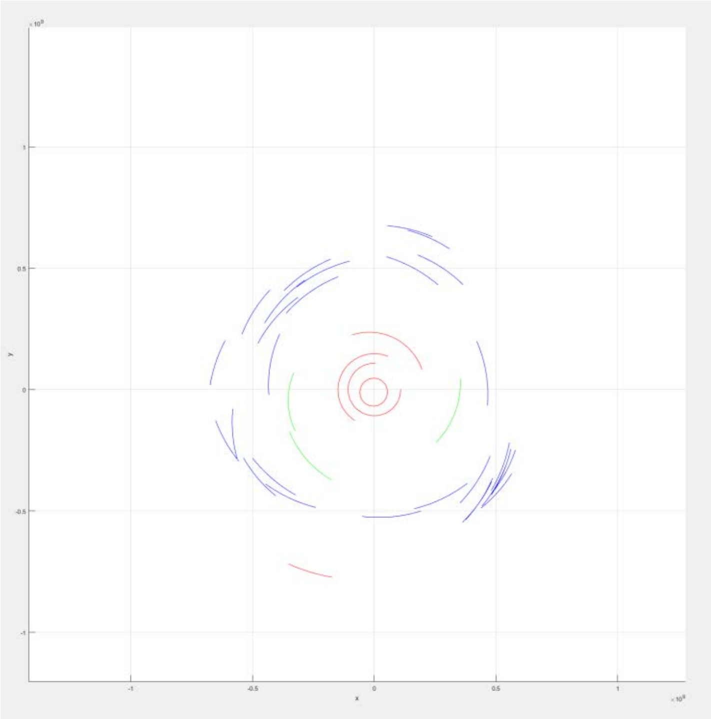


Figure 5.3 – XY-Plot with trailing lines of Hilda asteroids (blue) with Jupiter and the inner planets (red), the sun (black), and Pallas, Ceres and Vesta (green)

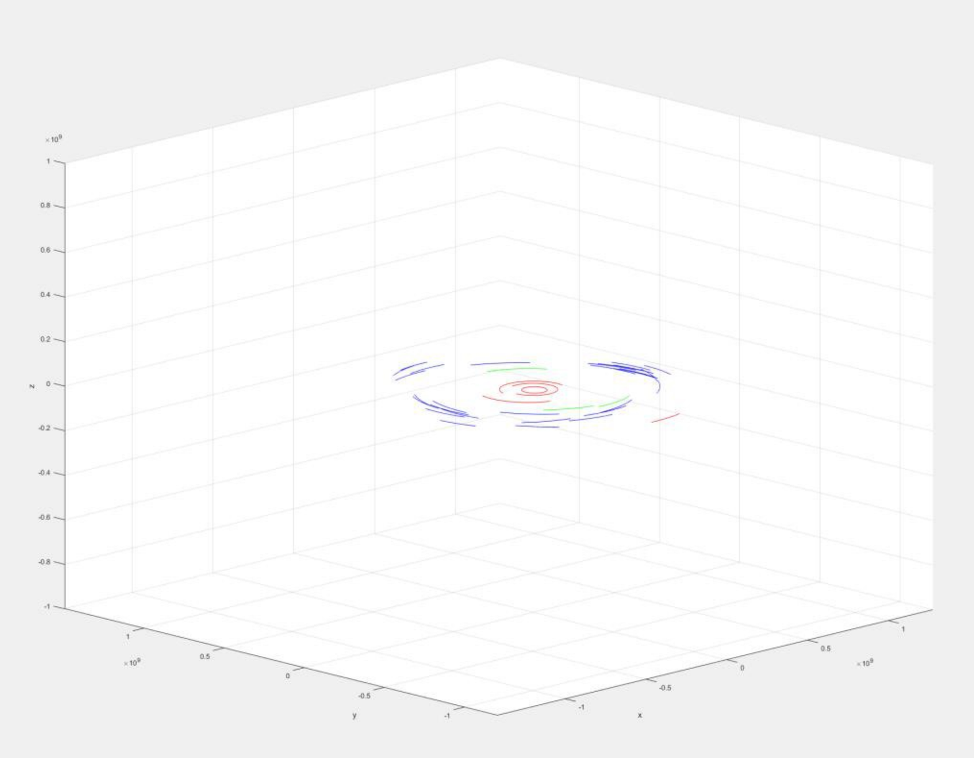


Figure 5.4 – XYZ-Plot with trailing lines of Hilda asteroids (blue) with Jupiter and the inner planets (red), the sun (black), and Pallas, Ceres and Vesta (green)

5.3 Discussion

Figures 5.1-5.4 show the simulation results with the Hilda asteroid positions highlighted in blue. The triangular formation of the Hilda asteroids is more pronounced with Figures 5.3 and 5.4. Separately, the Hilda asteroids follow a simple elliptical orbit, but grouped together they appear to be travelling along a triangle. Jupiter is always at the base of the triangle, which helps the Hilda asteroids stay in formation as they never approach Jupiter close enough to be ejected from their established orbits.

The reason for the triangular shape appears to be due to the orbital characteristics of the asteroid group. Those in the group have orbits in which a particular point in their orbit comes close to that of the orbit of Jupiter. Due to this overlapping of orbits, the asteroids will intersect with L3, L4, and L5. The intersection with the Lagrange points are also due to the 3:2 orbital resonance, in which the most stable configuration would be an approach to one Lagrange point every orbit. If the aphelion corresponded to the points in between Lagrange points, then the asteroid would approach Jupiter instead, which would result in an unstable orbit due to the large perturbation experienced with approaching a gas giant. Indeed, if one were to observe a single Hildian asteroid for a certain number of orbits, one would notice that the aphelion of the asteroid always corresponds to a Lagrangian point approach. This also explains the perceived clumping of Hilda asteroids around the Lagrangian points, as the aphelion of a body is also the point in which orbital velocity is the lowest. This configuration also results in the perihelion of a Hildian asteroid as being in between Lagrangian points, which explains why Jupiter is always located at one of the bases of the triangle as it is between L4 and L5, as the basses of the triangle represent the asteroids that are at perihelion. All these factors combined give the appearance of a triangular cluster of asteroids.

6. Solar System Model with Hilda Trojans and Jupiter Trojans

6.1 Simulation Setup

The setup for this simulation remains the same as that in Chapter 5 with the addition of Jupiter Trojans. There are 8 asteroids in L4 and 14 asteroids in L5 included. ODE113 with tolerances of 10E-17 was used and the model was propagated for 50 years.

6.2 Results

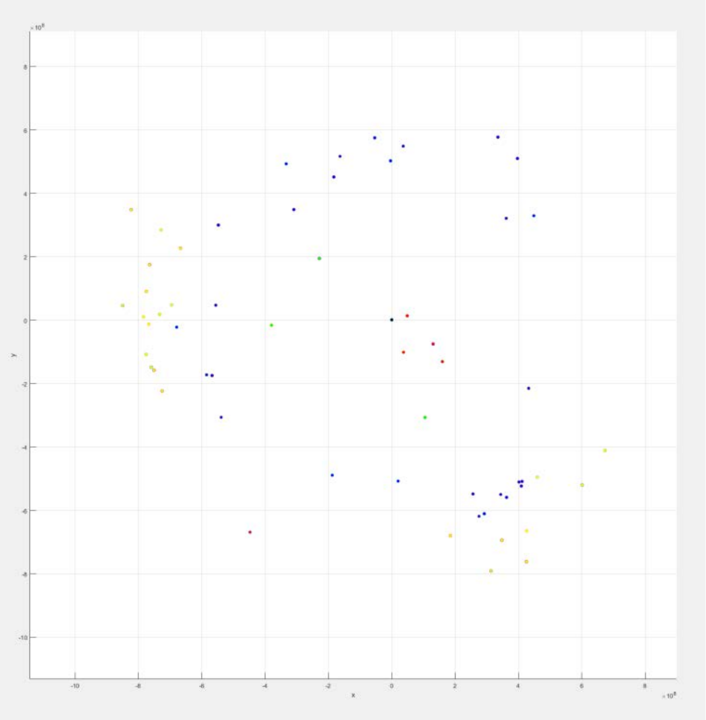


Figure 6.1 – XY-Plot of Hilda asteroids (blue) with Jupiter and the inner planets (red), the sun (black), Jupiter Trojans (yellow), and Pallas, Ceres and Vesta (green)

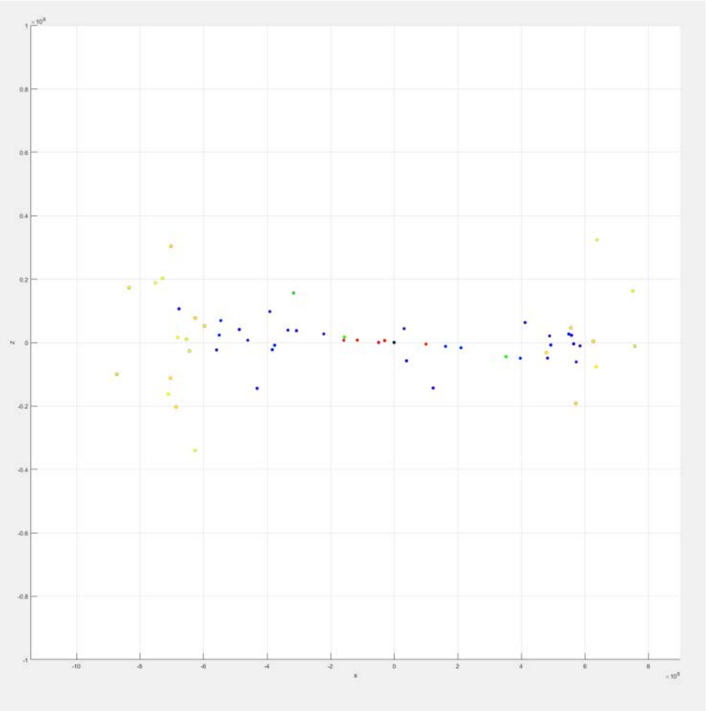


Figure 6.2 – XZ-Plot of Hilda asteroids (blue) with Jupiter and the inner planets (red), the sun (black), Jupiter Trojans (yellow), and Pallas, Ceres and Vesta (green)

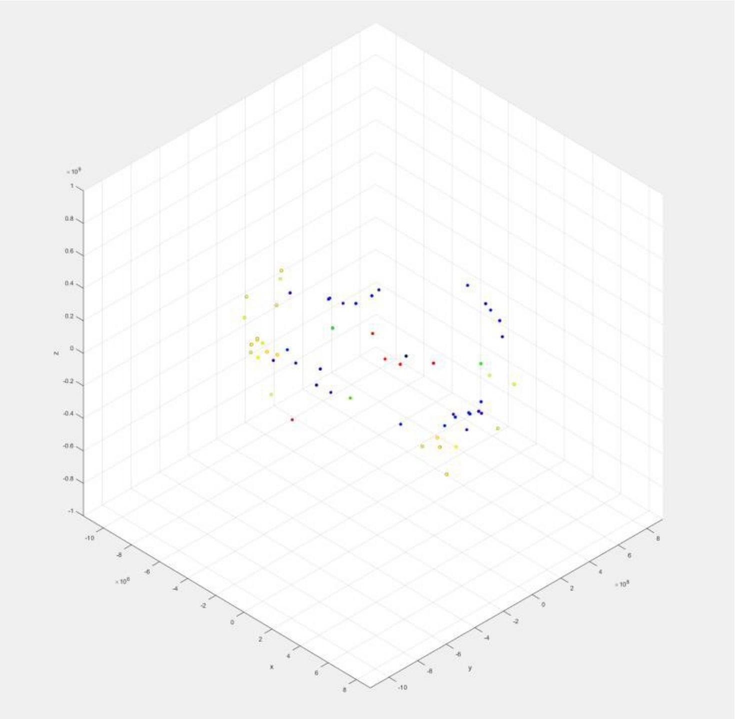


Figure 6.3 – XYZ-Plot of Hilda asteroids (blue) with Jupiter and the inner planets (red), the sun (black), Jupiter Trojans (yellow), and Pallas, Ceres and Vesta (green)

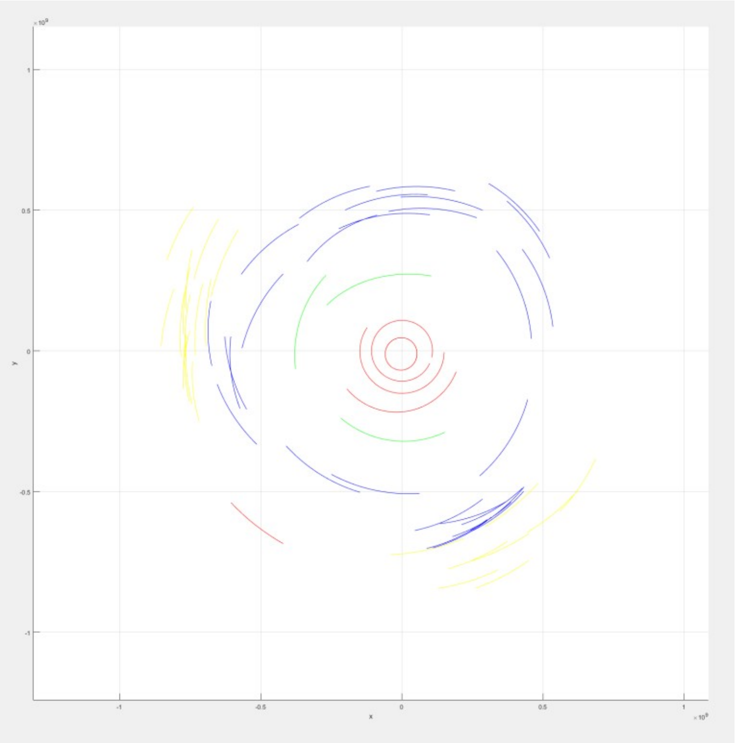


Figure 6.4 – XY-Plot with trailing lines of Hilda asteroids (blue) with Jupiter and the inner planets (red), the sun (black), Jupiter Trojans (yellow), and Pallas, Ceres and Vesta (green)

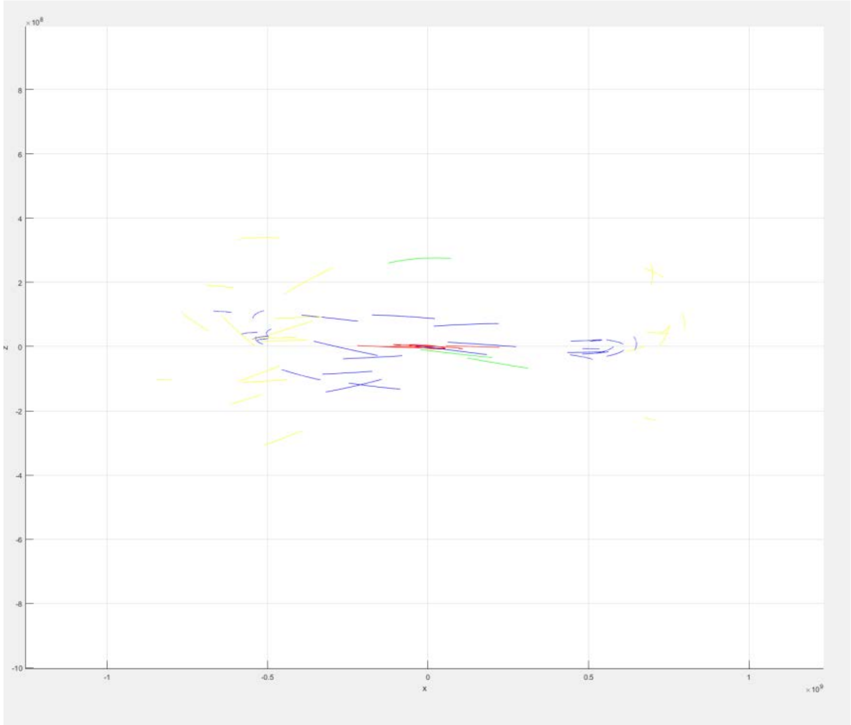


Figure 6.5 – XZ-Plot with trailing lines of Hilda asteroids (blue) with Jupiter and the inner planets (red), the sun (black), Jupiter Trojans (yellow), and Pallas, Ceres and Vesta (green)

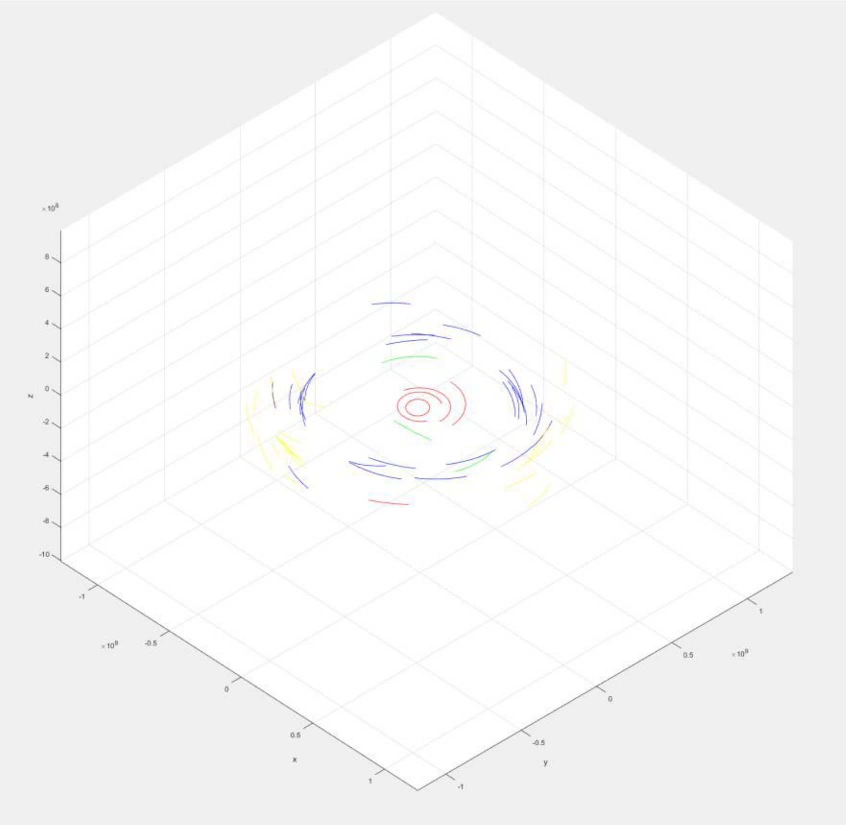


Figure 6.6 – XYZ-Plot with trailing lines of Hilda asteroids (blue) with Jupiter and the inner planets (red), the sun (black), Jupiter Trojans (yellow), and Pallas, Ceres and Vesta (green)

6.3 Discussion

As expected, the Jupiter Trojans keep pace with the orbit of Jupiter, staying in L4 and L5 during their orbit. The Trojans are on the outskirts of the Hildas and are at higher inclinations, and therefore do not interact which each other. The Hilda asteroids also must keep a substantial distance from the Lagrangian points to avoid being captured, further reducing the possibility of any interactions with the Trojans.

The model itself has demonstrated the capability to simulate any number of objects. The current simulation contains 61 total bodies and displays no odd behavior when even when propagated for as long as 50 years. All the orbits are stable and behave in accordance to theory.

7. Conclusion

The project aim was to study and explain the peculiar shape of the Hilda asteroid group. To achieve this goal, two perturbation models are presented and compared to each other to determine which model is most suitable to use in simulating the Hilda asteroid orbits. Cowell’s formulation presented a simple way to model orbits at the expense of accuracy, while the JPL point mass equations included a more comprehensive model at the expense of more complex calculations. Ultimately, a combination of Cowell’s formulation and JPL equations was chosen to model the system. JPL equations were used for major bodies to retain accuracy as those bodies had the largest effect on the rest of the bodies in the model, and Cowell’s formulation was used for the minor bodies to save computational resources and time.

The reason for the triangular shape of the Hilda asteroids is due to the orbital characteristics of the group. The most stable long-term orbit given the orbital resonance of the Hilda asteroids were that which located each aphelion at L3, L4, and L5, giving the appearance of a triangle. This resulted in clumps of asteroids along each Lagrangian point while spreading out the asteroids along the base of the triangle as those points correspond to perihelion.

Recommendations for future work include expanding the model to include yet more objects, as well as refining the code further to achieve even higher accuracy. As it stands, depending on the integrator used, the results can either be accurate with a long run-time or approximate with a short run-time. Finding an integrator that serves as a middle ground between the two or further improving the implementation of the current integrators would be the next step. Moreover, exotic objects such as comets that originate from the Oort Cloud can be included to further stress test the model.

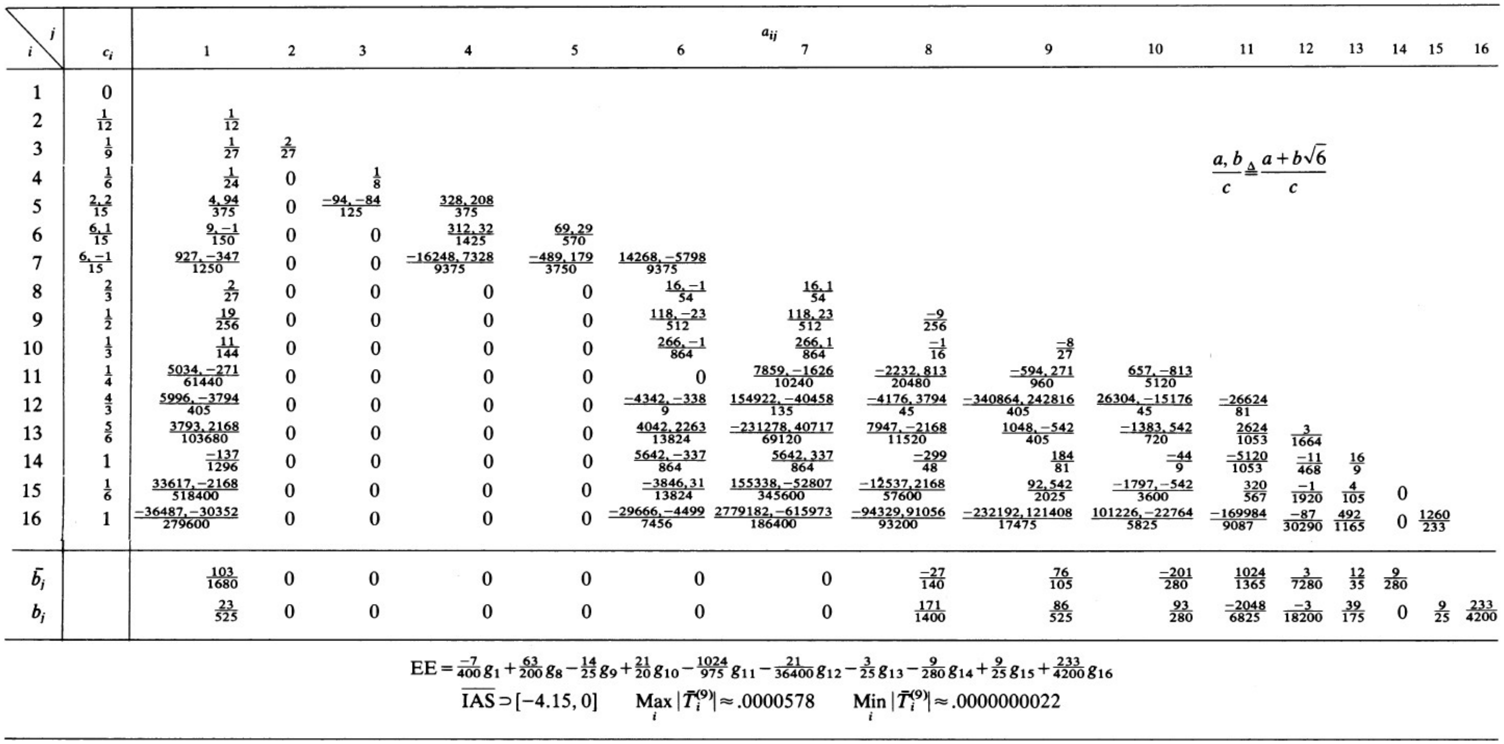
8. References

1. Cornish, N.J., "The Lagrange Points", https://map.gsfc.nasa.gov/ContentMedia/lagrange.pdf
2. Connors, M., Wiegert, P., Veillet, C., "Earth's first Trojan asteroid: 2010 TK7", http://www.astro.uwo.ca/~wiegert/2010TK7/
3. Raymond, S.N., Izidoro, A., "The empty primordial asteroid belt", *Science Advances*, Vol. 3, No. 9, 2017
4. Petit, J., Morbidelli, A., Chambers, J., "The Primordial Excitation and Clearing of the Asteroid Belt", *Icarus*, Vol. 153, No. 2, 2001, pp.338-347
5. Cellino, A., Bus, S.J., Doressoundiram, A., Lazzaro, D., "Spectroscopic Properties of Asteroid Families", *Asteroids III*, 2002, pp.633-643
6. Nicholson, S.B., "The Trojan Asteroids", *Astronomical Society of the Pacific Leaflets*, Vol. 8, No. 381, 1961, pp.239
7. Jewitt, D.C., Sheppard, S.S., Porco, C., *Jupiter: The Planet, Satellites, and Magnetosphere*, Cambridge University Press, 2004
8. Fernandez, Y.R., Sheppard, S.S., Jewitt, D.C., "The Albedo Distribution of Jovian Trojan Asteroids", *The Astronomical Journal*, Vol. 126, No. 3, 2003, pp. 1563-1574
9. Oshima, K., Yanao, T., "Jumping mechanisms of Trojan asteroids in the planar restricted three- and four-body problems", *Celestial Mechanics and Dynamical Astronomy*, Vol. 122, No. 1, 2015, pp.53-74.
10. Fiorenza, N.A., "Jupiter's Trojans", http://www.lunarplanner.com/HCpages/Jupiter-Trojans/index.html
11. Scheirich P., "Asteroid (and Comet) Groups", http://sajri.astronomy.cz/asteroidgroups/groups.htm
12. Ohtsuka, K., Ito, T., Yoshikawa, M., Asher D.J., Arakida, H., "Quasi-Hilda comet 147P/Kushida-Muramatsu. Another long temporary satellite capture by Jupiter", *Astronomy and Astrophysics*, Vol. 489, No. 3, 2008, pp.1355-1362
13. Verner, J.H., "Explicit Runge-Kutta Methods with Estimates of the Local Truncation Error", SIAM Journal on Numerical Analysis, Vol.15, 1978
14. JPL Horizons On-Line Ephemeris System, URL: https://ssd.jpl.nasa.gov/?horizons
15. Cowell, P.H., and Crommelin, A.C.D., "Investigation of the Motion of Halley's Comet from 1758 to 1910", Royal Observatory, Greenwich, 1910
16. Standish, E.M., and Williams, J.G., *Explanatory Supplement to the Astronomical Almanac*, 3rd ed., 2012, Chap. 8
17. Shampine, L.F., and Reichelt, M.W. "The MATLAB ODE Suite", *SIAM Journal on Scientific Computing*, Vol. 18, 1997, pp.1-22.

9. Appendix

Appendix A

Table A – Butcher’s tableau for the Runge-Kutta Verner 89 scheme [13]



$i \backslash j$	c_i	a_{ij}															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0																
2	$\frac{1}{12}$	$\frac{1}{12}$															
3	$\frac{1}{9}$	$\frac{1}{27}$	$\frac{2}{27}$														
4	$\frac{1}{6}$	$\frac{1}{24}$	0	$\frac{1}{8}$													
5	$\frac{2, 2}{15}$	$\frac{4, 94}{375}$	0	$\frac{-94, -84}{125}$	$\frac{328, 208}{375}$												
6	$\frac{6, 1}{15}$	$\frac{9, -1}{150}$	0	0	$\frac{312, 32}{1425}$	$\frac{69, 29}{570}$											
7	$\frac{6, -1}{15}$	$\frac{927, -347}{1250}$	0	0	$\frac{-16248, 7328}{9375}$	$\frac{-489, 179}{3750}$	$\frac{14268, -5798}{9375}$										
8	$\frac{2}{3}$	$\frac{2}{27}$	0	0	0	0	$\frac{16, -1}{54}$	$\frac{16, 1}{54}$									
9	$\frac{1}{2}$	$\frac{19}{256}$	0	0	0	0	$\frac{118, -23}{512}$	$\frac{118, 23}{512}$	$\frac{-9}{256}$								
10	$\frac{1}{3}$	$\frac{11}{144}$	0	0	0	0	$\frac{266, -1}{864}$	$\frac{266, 1}{864}$	$\frac{-1}{16}$	$\frac{-8}{27}$							
11	$\frac{1}{4}$	$\frac{5034, -271}{61440}$	0	0	0	0	0	$\frac{7859, -1626}{10240}$	$\frac{-2232, 813}{20480}$	$\frac{-594, 271}{960}$	$\frac{657, -813}{5120}$						
12	$\frac{4}{3}$	$\frac{5996, -3794}{405}$	0	0	0	0	$\frac{-4342, -338}{9}$	$\frac{154922, -40458}{135}$	$\frac{-4176, 3794}{45}$	$\frac{-340864, 242816}{405}$	$\frac{26304, -15176}{45}$	$\frac{-26624}{81}$					
13	$\frac{5}{6}$	$\frac{3793, 2168}{103680}$	0	0	0	0	$\frac{4042, 2263}{13824}$	$\frac{-231278, 40717}{69120}$	$\frac{7947, -2168}{11520}$	$\frac{1048, -542}{405}$	$\frac{-1383, 542}{720}$	$\frac{2624}{1053}$	$\frac{3}{1664}$				
14	1	$\frac{-137}{1296}$	0	0	0	0	$\frac{5642, -337}{864}$	$\frac{5642, 337}{864}$	$\frac{-299}{48}$	$\frac{184}{81}$	$\frac{-44}{9}$	$\frac{-5120}{1053}$	$\frac{-11}{468}$	$\frac{16}{9}$			
15	$\frac{1}{6}$	$\frac{33617, -2168}{518400}$	0	0	0	0	$\frac{-3846, 31}{13824}$	$\frac{155338, -52807}{345600}$	$\frac{-12537, 2168}{57600}$	$\frac{92, 542}{2025}$	$\frac{-1797, -542}{3600}$	$\frac{320}{567}$	$\frac{-1}{1920}$	$\frac{4}{105}$	0		
16	1	$\frac{-36487, -30352}{279600}$	0	0	0	0	$\frac{-29666, -4499}{7456}$	$\frac{2779182, -615973}{186400}$	$\frac{-94329, 91056}{93200}$	$\frac{-232192, 121408}{17475}$	$\frac{101226, -22764}{5825}$	$\frac{-169984}{9087}$	$\frac{-87}{30290}$	$\frac{492}{1165}$	0	$\frac{1260}{233}$	
$\bar{b}_j$		$\frac{103}{1680}$	0	0	0	0	0	0	$\frac{-27}{140}$	$\frac{76}{105}$	$\frac{-201}{280}$	$\frac{1024}{1365}$	$\frac{3}{7280}$	$\frac{12}{35}$	$\frac{9}{280}$		
b_j		$\frac{23}{525}$	0	0	0	0	0	0	$\frac{171}{1400}$	$\frac{86}{525}$	$\frac{93}{280}$	$\frac{-2048}{6825}$	$\frac{-3}{18200}$	$\frac{39}{175}$	0	$\frac{9}{25}$	$\frac{233}{4200}$

$$\frac{a, b}{c} \triangleq \frac{a+b\sqrt{6}}{c}$$

$$\mathrm{EE} = \frac{-7}{400} g_1 + \frac{63}{200} g_8 - \frac{14}{25} g_9 + \frac{21}{20} g_{10} - \frac{1024}{975} g_{11} - \frac{21}{36400} g_{12} - \frac{3}{25} g_{13} - \frac{9}{280} g_{14} + \frac{9}{25} g_{15} + \frac{233}{4200} g_{16}$$

$$\overline{\mathrm{IAS}} \supset [-4.15, 0] \qquad \operatorname*{Max}_i |\bar{T}_i^{(9)}| \approx .0000578 \qquad \operatorname*{Min}_i |\bar{T}_i^{(9)}| \approx .0000000022$$

Appendix B

Table B – Positional ephemeris results from DE 430 with the sun as origin [14]

Timestamp		2006-05-04 07:27:35.97	2006-08-03 14:54:23.04	2006-11-02 22:21:36.02	2007-02-02 07:51:25.02
Mercury	X	53463653.72	53411252.28	49573425.7	41381757.75
	Y	-17631837.55	-3234980.003	11366082.96	25316447.81
	Z	-6346812.14	-5165783.124	-3620696.86	-1729214.639
Venus	X	38705882.22	25584539.9	-77804828.95	107099063.8
	Y	-101718302.5	104704359.3	-75167354.74	16956521.78
	Z	-3625861.822	-43885.71892	3462020.799	-5949339.795
Earth	X	-109192841.1	99688163.85	113249062.4	-100509511.4
	Y	-104065343.5	-114470005.5	95925915.26	107842936.8
	Z	1377.973027	1987.208151	-1695.698427	-1818.671219
Mars	X	-168408587.5	-246200439.2	-195214977.8	-30746567.14
	Y	180762786.3	30165097.61	-135267012.4	-218133994.9
	Z	7924608.089	6680118.483	1961399.665	-3814850.4
Jupiter	X	-584438352.3	-509852834.9	-427295644	-337891186.5
	Y	-562062047.3	-627024983.9	-682180006.6	-726572351.7
	Z	15411427.68	14012125.76	12393706.8	10577339.48
Saturn	X	-901226897.2	-961020255.2	-1017707774	-1071174194
	Y	1025892045	973736177.6	918447098.6	860166757.9
	Z	18003300.48	21290843.06	24509248.93	27651696.02
Uranus	X	2845386120	2861810061	2877369181	2892072139
	Y	-961845572	-913377868.9	-864623542.9	-815550567.1
	Z	-40426570.6	-40458685.37	-40478662.21	-40486584.83
Neptune	X	3326994211	3355455793	3383619995	3411509196
	Y	-3024402432	-2992353601	-2960024152	-2927385128
	Z	-14395753.2	-15712237.96	-17027693.83	-18342865.94
Pluto	X	-396653057	-353067914.1	-309442220.9	-265745372.9
	Y	-4596977745	-4608367174	-4619369315	-4629999729
	Z	606458006.6	595080136.1	583650350.6	572158968.3

Appendix C

Table C – Positional ephemeris results from Cowell's formulation with RKV-89

Timestamp		2006-05-04 07:27:35.97	2006-08-03 14:54:23.04	2006-11-02 22:21:36.02	2007-02-02 07:51:25.02
Mercury	X	53463735	53416502	49589113	41414562
	Y	-17644382	-3258175	11331514	25275563
	Z	-6347840	-5168147	-3624936	-1735533
Venus	X	38697697	25594266	-77763662	107114779
	Y	-101728030	104687269	-75210973	16932870
	Z	-3625602	-44677	3459060	-5950629
Earth	X	-109212952	99576340	113957945	-99769611
	Y	-104137877	-114864815	95325372	108464955
	Z	-3897	1575	4138	-1528
Mars	X	-168414513	-246222947	-195286053	-30922628
	Y	180759736	30154583	-135289662	-218250980
	Z	7924769	6680722	1963193	-3812270
Jupiter	X	-584446134	-509878557	-427347072	-337976328
	Y	-562060730	-627031375	-682203274	-726623107
	Z	15411737	14012953	12395279	10579923
Saturn	X	-901243171	-961061941	-1017782365	-1071289708
	Y	1025885247	973713950	918400831	860087142
	Z	18006339	21296976	24518973	27665302
Uranus	X	2845378447	2861784950	2877318987	2891989355
	Y	-961848668	-913392909	-864658733	-815615475
	Z	-40427798	-40460850	-40481564	-40489939
Neptune	X	3326987357	3355432582	3383572525	3411430260
	Y	-3024400657	-2992358854	-2960044642	-2927430508
	Z	-14390369	-15701386	-17010978	-18320240
Pluto	X	-396610620	-352992585	-309345587	-265632708
	Y	-4597036987	-4608497257	-4619580848	-4630298162
	Z	606283634	594726870	583121442	571457645

Appendix D

Table D – Positional ephemeris results from DE 430 with solar system barycenter as origin [14]

Timestamp		2006-05-04 07:27:07.97	2006-08-03 14:54:58.98	2006-11-02 22:21:10.97	2007-02-02 05:07:51.02
Mercury	X	53983122.36	53873845.11	49972293.36	42048987.99
	Y	-17192082.99	-2719397.466	11943034.19	25519330.23
	Z	-6364236.89	-5182484.999	-3637031.949	-1810883.522
Venus	X	39224603	26046138.27	-77407073.42	107480131.9
	Y	-101277536.6	105218408.5	-74588524.46	17251564.46
	Z	-3643142.802	-60683.28959	3445877.547	-5972373.006
Earth	X	-108673772.6	100151778.7	113647915.7	-99964239.78
	Y	-103623625.3	-113955549.7	96503544.23	108676105.4
	Z	-15950.81523	-14885.17587	-17882.42408	-17091.55149
Mars	X	-167888486.1	-245737687.3	-194816987.4	-30664455.12
	Y	181204302.4	30678064.24	-134688370.1	-217488040.3
	Z	7907276.272	6663231.38	1945231.358	-3823836.634
Jupiter	X	-583918971	-509389650.7	-426897558.6	-337679433.8
	Y	-561620689.3	-626511501.8	-681601652.9	-725891239.4
	Z	15394103.47	13995246.48	12377525.51	10564446.72
Saturn	X	-900707049.2	-960557695.5	-1017309241	-1070783289
	Y	1026333336	974249689.3	919025474.2	860873974
	Z	17985959.98	21273985.51	24493052.08	27632582.43
Uranus	X	2845905691	2862272960	2877767492	2892380949
	Y	-961404632.1	-912863890.9	-864045502.9	-814978674.7
	Z	-40443899.23	-40475557.88	-40494848.94	-40501843.07
Neptune	X	3327513740	3355918747	3384018265	3411801255
	Y	-3023961434	-2991839697	-2959446060	-2926792826
	Z	-14413077.27	-15729116.32	-17043876.41	-18356494.41
Pluto	X	-396133580.6	-352604890.2	-309044001.2	-265473029.9
	Y	-4596536592	-4607853468	-4618791085	-4629353527
	Z	606440717.9	595063212.4	583634200.7	572158086.5

Appendix E

Appendix E – Positional ephemeris results from JPL point mass equation with RKV-89

Timestamp		2006-05-04 07:27:07.97	2006-08-03 14:54:58.98	2006-11-02 22:21:10.97	2007-02-02 05:07:51.02
Mercury	X	53983131	53873867	49972321	42049013
	Y	-17192081	-2719387	11943075	25519364
	Z	-6364237	-5182485	-3637031	-1810882
Venus	X	39224604	26046377	-77407150	107480237
	Y	-101277572	105218396	-74588362	17251380
	Z	-3643142	-60698	3445887	-5972381
Earth	X	-108685054	100064914	114291154	-99267176
	Y	-103692676	-114295182	95986611	109319398
	Z	-21361	-15481	-12116	-16791
Mars	X	-167888484	-245737691	-194817027	-30664566
	Y	181204308	30678088	-134688341	-217488062
	Z	7907276	6663232	1945233	-3823834
Jupiter	X	-583920544	-509393058	-426902481	-337685798
	Y	-561616145	-626502368	-681588153	-725872663
	Z	15394244	13995520	12377917	10564961
Saturn	X	-900717132	-960577244	-1017338128	-1070822202
	Y	1026329751	974242733	919014656	860859413
	Z	17988830	21279570	24501619	27644180
Uranus	X	2845904213	2862270001	2877763079	2892374999
	Y	-961404509	-912863633	-864045117	-814978141
	Z	-40445294	-40478272	-40498909	-40507206
Neptune	X	3327513082	3355917687	3384016562	3411799118
	Y	-3023956441	-2991829650	-2959430964	-2926772729
	Z	-14407861	-15718813	-17028319	-18335884
Pluto	X	-396084948	-352507412	-308901602	-265283765
	Y	-4596592618	-4607968250	-4618967023	-4629586572
	Z	606266178	594709396	583104132	571454897

Appendix F

Appendix F – Positional ephemeris results from JPL point mass equation with ODE113

Timestamp		2007-02-02 00:00:00.00	2007-02-02 00:00:00.00	2007-02-02 00:00:00.00	2007-02-02 00:00:00.00
Tolerance		10E-10	10E-12	10E-15	10E-17
Mercury	X	41963312.19900	41963312.33876	41963312.33947	41963312.33945
	Y	25628386.94468	25628386.76654	25628386.76562	25628386.76565
	Z	-1794114.75206	-1794114.779446	-1794114.77958	-1794114.77958
Venus	X	107466532.1575	107466532.1575	107466532.1575	107466532.1575
	Y	17336864.82902	17336864.82901	17336864.82898	17336864.82902
	Z	-5970421.48896	-5970421.488965	-5970421.48896	-5970421.48896
Earth	X	-99322886.9881	-99322886.98810	-99322886.9880	-99322886.9881
	Y	109269155.5371	109269155.5371	109269155.537	109269155.5371
	Z	-16792.4932677	-16792.49326774	-16792.4932677	-16792.4932677
Mars	X	-30602795.3719	-30602795.37192	-30602795.3719	-30602795.3719
	Y	-217491331.770	-217491331.7708	-217491331.770	-217491331.770
	Z	-3825420.64053	-3825420.640534	-3825420.64053	-3825420.64053
Jupiter	X	-337656807.736	-337656807.7368	-337656807.736	-337656807.736
	Y	-725884803.776	-725884803.7763	-725884803.776	-725884803.776
	Z	10564362.08322	10564362.08322	10564362.08322	10564362.08322
Saturn	X	-1070838498.28	-1070838498.281	-1070838498.28	-1070838498.28
	Y	860840676.5565	860840676.5565	860840676.556	860840676.5565
	Z	27645155.46437	27645155.46437	27645155.46437	27645155.46437
Uranus	X	2892379457.726	2892379457.726	2892379457.726	2892379457.726
	Y	-814962665.450	-814962665.4505	-814962665.450	-814962665.450
	Z	-40507206.9272	-40507206.92723	-40507206.9272	-40507206.9272
Neptune	X	3411807809.073	3411807809.073	3411807809.073	3411807809.073
	Y	-2926762413.80	-2926762413.803	-2926762413.80	-2926762413.80
	Z	-18336294.8907	-18336294.89076	-18336294.8907	-18336294.8907

Appendix G

JPL Point Mass Equation with ODE113 Integrator MATLAB Code:

```
%% Input parameters
clc; clear variables;

global mu_n beta gamma c_constant a_body_n seed_flag
global a_body_update bodies hilda_bodies trojan_bodies

%% Program Variables and Constants

% Orbit propagator
beta = 1;                   % PPN parameter
gamma = 1;                  % PPN parameter
c_constant = 299792.458;    % Speed of light (km/s)
t_span = [0 1607040000];    % Length of time step taken by ode113 (seconds)
opts = odeset('Reltol',1e-17 ,'AbsTol',1e-17);
h = t_span(2);
A_r = 2.4;          % Various density constants for different Tholen types
B_r = 1.8;
C_r = 1.8;
D_r = 1.8;
E_r = 2.4;
F_r = 1.8;
G_r = 1.8;
K_r = 2.4;
M_r = 5.0;
P_r = 1.8;
Q_r = 2.4;
R_r = 2.4;
S_r = 2.4;
T_r = 1.8;
V_r = 2.4;
X_r = 5.0;
hilda_bodies = 29;
trojan_bodies = 22;

% Mu Constants (km^3/s^2)
mu_sun = 1.3271244004193938E11;
mu_mercury = 22032.09;
mu_venus = 324858.63;
mu_earth = 398600.440;
mu_mars = 42828.3;
mu_jupiter = 126686511;
mu_saturn = 37931207.8;
mu_uranus = 5793966;
mu_neptune = 6835107;
mu_ceres = 62.6284;
mu_pallas = 14.3;
mu_vesta = 17.8;

% Celestial and Spacecraft Initial State Vectors (km;km/d)->km/s
% 2018/01/01 00:00:00.0000 TDB
```

```
sun_r_state = [2.696822729957703E+05;  9.170988258285450E+05; -
1.798365568174730E+04];
sun_v_state = [-8.749385007982006E+02;  7.515130458753841E+02;
2.120786313213856E+01]./(24*60*60);

mercury_r_state = [-5.773063859343297E+07; -2.474809078820562E+05;
5.207789164126894E+06];
mercury_v_state = [-7.919962805974684E+05; -4.026357868397071E+06; -
2.564686044723194E+05]./(24*60*60);

venus_r_state = [1.091042254713612E+07; -1.073453106424018E+08; -
2.117095924141012E+06];
venus_v_state = [2.990013044711243E+06;  2.859017701976050E+05; -
1.686609333649275E+05]./(24*60*60);

earth_r_state = [-2.594286566500337E+07;  1.456625130721959E+08; -
2.366799613461643E+04];
earth_v_state = [-2.574145128774615E+06; -4.675320299287317E+05; -
1.676891043500532E+01]./(24*60*60);

mars_r_state = [-2.366443833882647E+08; -5.728070983468132E+07;
4.576782272909489E+06];
mars_v_state = [5.767010098299264E+05; -1.853403852730346E+06; -
5.300709637187062E+04]./(24*60*60);

jupiter_r_state = [-6.372073092898788E+08; -5.028276068515576E+08;
1.633825735300046E+07];
jupiter_v_state = [6.861006265459799E+05; -8.324934352852211E+05; -
1.188674636430751E+04]./(24*60*60);

saturn_r_state = [7.165340481072909E+06; -1.504508147585481E+09;
2.587354754541469E+07];
saturn_v_state = [7.886224599062497E+05; 1.325792124311489E+03; -
3.142445982684764E+04]./(24*60*60);

uranus_r_state = [2.651366065542918E+09;  1.355805938558842E+09; -
2.931337452471685E+07];
uranus_v_state = [-2.722088277171703E+05;  4.964368402640501E+05;
5.351277571902309E+03]./(24*60*60);

neptune_r_state = [4.290710317482585E+09; -1.285293794759043E+09; -
7.241554287538326E+07];
neptune_v_state = [1.316883503783873E+05;  4.526371662227903E+05; -
1.241014786165766E+04]./(24*60*60);

ceres_r_state = [-2.127972318122728E+08;  3.191129816053626E+08;
4.927982465331382E+07];
ceres_v_state = [-1.315703517030119E+06; -9.816011256051966E+05;
2.114953881483568E+05]./(24*60*60);

pallas_r_state = [1.832611984150499E+08;  2.467418328153097E+08; -
1.851951060659268E+08];
pallas_v_state = [-1.663298914872307E+06;  5.654288605275278E+05; -
2.508321274737805E+05]./(24*60*60);
```

```
vesta_r_state = [-2.717662543931279E+08; -1.792566866007622E+08;
3.846472647458429E+07];
vesta_v_state = [1.063218192384667E+06; -1.444057075890015E+06; -
8.617207072321705E+04]./(24*60*60);

%% Hilda Asteroids
a334_r_state = [-5.660923958034916E+07;5.932523144785926E+08;-
2.750858982085955E+07];
a334_v_state = [-1.267509260926113E+06;-
1.155027491042804E+05;8.469968889220349E+04]./(24*60*60);
mu_334 = (99.385^3)*C_r*(6.27E-22);

a153_r_state =
[5.164456734450716E+08;3.795425100028710E+08;1.810721535376635E+07];
a153_v_state = [-5.794380049776826E+05;1.035806137802320E+06;-
1.541380450061696E+05]./(24*60*60);
mu_153 = (85.315^3)*P_r*(6.27E-22);

a190_r_state = [1.314910887307515E+08;-
6.737952778107764E+08;7.165355740148711E+07];
a190_v_state = [1.065680461414187E+06;2.586151034742122E+05;-
3.693470157560058E+04]./(24*60*60);
mu_190 = (79.5^3)*P_r*(6.27E-22);

a361_r_state = [-
3.707920467471800E+08;3.203721253409100E+08;9.458272793096821E+07];
a361_v_state = [-1.161969842635968E+06;-9.672440605517696E+05;-
1.210348708484102E+05]./(24*60*60);
mu_361 = (77.167^3)*D_r*(6.27E-22);

a499_r_state = [4.099488357569566E+07;-
7.117739171502970E+08;7.620392929561943E+06];
a499_v_state =
[1.044808822909851E+06;1.681510194536707E+05;3.567240852185893E+04]./(24*60*6
0);
mu_499 = (38.664^3)*P_r*(6.27E-22);

a748_r_state = [1.979319639125218E+08;-
6.578397007828205E+08;9.807122183297336E+06];
a748_v_state =
[1.015848352238302E+06;4.087148744803442E+05;3.872857446297735E+04]./(24*60*6
0);
mu_748 = (51.8625^3)*P_r*(6.27E-22);

a1038_r_state = [4.596570587453505E+08;-2.956410036061012E+07;-
6.547359932125793E+07];
a1038_v_state =
[8.822414828928080E+04;1.605148098424734E+06;1.261878297833073E+05]./(24*60*6
0);
mu_1038 = (29.15^3)*D_r*(6.27E-22);

a1162_r_state = [-4.507618447632506E+08;-
2.875205111878963E+08;2.196995683731511E+06];
a1162_v_state = [8.342173262941178E+05;-1.148904160388616E+06;-
4.659901322327492E+04]./(24*60*60);
```

```
mu_1162 = (21.1215^3)*P_r*(6.27E-22);

a1180_r_state = [-
6.280746250609601E+08;1.052283766488016E+08;7.968817370793879E+07];
a1180_v_state = [-3.067040515129521E+04;-1.192887198961898E+06;-
6.276625663810677E+02]./(24*60*60);
mu_1180 = (48.5^3)*P_r*(6.27E-22);

a1212_r_state = [3.232316147978092E+08;4.497193453424651E+08;-
7.337611261278240E+07];
a1212_v_state = [-
1.240572619878425E+06;5.840356800016183E+05;1.666011738444269E+04]./(24*60*60
);
mu_1212 = (38.1975^3)*P_r*(6.27E-22);

a1268_r_state = [1.606762055671854E+08;-6.320208325157750E+08;-
4.647607360842216E+07];
a1268_v_state =
[1.133215711820595E+06;2.707986053393988E+05;3.441208969245316E+04]./(24*60*6
0);
mu_1268 = (48.354^3)*P_r*(6.27E-22);

a1269_r_state = [1.517896843924856E+08;  5.477715179664397E+08; -
2.373907803504649E+07];
a1269_v_state = [-1.318860034999355E+06;  2.337054962407128E+05;
3.724679388308137E+04]./(24*60*60);
mu_1269 = (52.4465^3)*D_r*(6.27E-22);

a1345_r_state = [1.659562190021329E+08;  4.705648435598646E+08; -
9.235245681046961E+07];
a1345_v_state = [-1.447725572922735E+06;  3.720225688672043E+05;
1.422165682562500E+05]./(24*60*60);
mu_1345 = (36.4875^3)*X_r*(6.27E-22);

a1439_r_state = [3.409589509589228E+08;  4.728642988014538E+08;
1.362631466311350E+07];
a1439_v_state = [-1.153229142292553E+06;  6.433333885789039E+05;
8.769834990859644E+04]./(24*60*60);
mu_1439 = (25.271^3)*X_r*(6.27E-22);

a1512_r_state = [-3.067511431165601E+08; -4.036807552087385E+08; -
3.907572855667594E+07];
a1512_v_state = [1.217473290438057E+06; -8.491765717337778E+05; -
1.192334020633906E+05]./(24*60*60);
mu_1512 = (39.611^3)*P_r*(6.27E-22);

a1529_r_state = [-3.116123171782855E+08;  4.755641775730078E+08;
3.494372177530363E+07];
a1529_v_state = [-1.233779271165484E+06; -5.088635890090343E+05;
2.081759889260863E+05]./(24*60*60);
mu_1529 = (28.1635^3)*P_r*(6.27E-22);

a1578_r_state = [6.471077649717303E+07; -7.107013397302471E+08; -
3.659527541986108E+06];
```

```
a1578_v_state = [1.021407602010900E+06;  2.002285724043255E+05; -
1.306817878253321E+04]./(24*60*60);
mu_1578 = (23.5385^3)*D_r*(6.27E-22);

a1746_r_state = [2.252623976948957E+08; -4.844556948036578E+08; -
3.585681656399968E+07];
a1746_v_state = [1.148259558332988E+06;  8.120126077876096E+05;
1.981545661553841E+05]./(24*60*60);
mu_1746 = (31.2615^3)*D_r*(6.27E-22);

a1748_r_state = [-6.702955860542336E+08; -6.939940786928333E+07;
3.359724758960499E+07];
a1748_v_state = [3.051091908969706E+05; -1.076898138104233E+06;
2.213838979388566E+04]./(24*60*60);
mu_1748 = (31.2615^3)*D_r*(6.27E-22);

a1754_r_state = [5.437920282801523E+08;  2.791711247014878E+08; -
9.170021072005786E+07];
a1754_v_state = [-3.971027341648772E+05;  1.152826831908660E+06; -
2.119700722869303E+05]./(24*60*60);
mu_1754 = (39.76^3)*P_r*(6.27E-22);

a1902_r_state = [-6.662390058274922E+08;  2.231239437987160E+08;
1.521368020779454E+08];
a1902_v_state = [-2.737493200728401E+05; -1.003238877696629E+06; -
6.151943706226264E+04]./(24*60*60);
mu_1902 = (41.7215)*X_r*(6.27E-22);

a1911_r_state = [-6.662390058274922E+08;  2.231239437987160E+08;
1.521368020779454E+08];
a1911_v_state = [-2.737493200728401E+05; -1.003238877696629E+06; -
6.151943706226264E+04]./(24*60*60);
mu_1911 = (41.7215^3)*X_r*(6.27E-22);

a2067_r_state = [1.930654365870309E+08; -6.638233610662347E+08;
2.588261143971369E+07];
a2067_v_state = [1.022729721256660E+06;  3.751628357025844E+05; -
4.480688834100836E+04]./(24*60*60);
mu_2067 = (23.0015^3)*P_r*(6.27E-22);

a2246_r_state = [2.439010950751203E+08;  5.529254487754790E+08; -
6.869849186060721E+07];
a2246_v_state = [-1.195044250793985E+06;  4.044560430306507E+05;
1.419194787819907E+04]./(24*60*60);
mu_2246 = (24.212^3)*D_r*(6.27E-22);

a2312_r_state = [-6.358116312821826E+06; -6.470670394949169E+08; -
2.749285225940189E+07];
a2312_v_state = [1.163717239716825E+06;  1.404430069751796E+05; -
8.640413678357804E+04]./(24*60*60);
mu_2312 = (25.061^3)*D_r*(6.27E-22);

a2760_r_state = [8.180204713589877E+07; -6.176387992849405E+08; -
1.445498170079314E+08];
```

```
a2760_v_state = [1.194965475511503E+06;  3.149512083166264E+04;
4.472749261442272E+04]./(24*60*60);
mu_2760 = (28.95^3)*X_r*(6.27E-22);

a466_r_state = [-2.702290010452100E+07;  4.937280662298011E+08;
5.192919974007326E+07];
a466_v_state = [-1.338153394496108E+06; -1.565424072232535E+05; -
4.524289411723662E+05]./(24*60*60);
mu_466 = (47.7475^3)*C_r*(6.27E-22);

a1144_r_state = [-5.968674246865408E+08;  1.048997484080484E+08;
2.408180383494028E+07];
a1144_v_state = [-2.566347119070432E+05; -1.179663559699131E+06;
2.035274470808721E+05]./(24*60*60);
mu_1144 = (28.1735^3)*D_r*(6.27E-22);

a1256_r_state = [5.394350358387843E+08;  2.319830796794252E+07;
3.206623233422379E+07];
a1256_v_state = [-1.311703013037190E+04;  1.401435709246058E+06; -
5.648605134925363E+04]./(24*60*60);
mu_1256 = (34.1265^3)*D_r*(6.27E-22);

%% Jupiter Trojans - L4 Greek Camp
a588_r_state = [2.195319367071073E+08; -8.561806742309635E+08; -
8.581805874557769E+07];
a588_v_state = [9.292220053389821E+05;  2.649920509082507E+05;
1.514176402554973E+05]./(24*60*60);
mu_588 = (65.0495^3)*D_r*(6.27E-22);

a624_r_state = [4.022244598253643E+08; -6.677661454961369E+08; -
1.705850538314853E+08];
a624_v_state = [9.585280056470216E+05;  4.905542831321214E+05;
2.467041106384269E+05]./(24*60*60);
mu_624 = (112.5^3)*D_r*(6.27E-22);

a659_r_state = [-9.291106100906771E+07; -7.251083366152976E+08; -
5.785787256025362E+07];
a659_v_state = [1.189516711550262E+06; -2.475031196685107E+04;
1.309242243527163E+04]./(24*60*60);
mu_659 = (56.16^3)*X_r*(6.27E-22);

a911_r_state = [2.027250940541521E+08; -7.584175975727792E+08; -
2.508343465297483E+08];
a911_v_state = [1.012838139854829E+06;  2.465190583537012E+05;
2.425595437734791E+05]./(24*60*60);
mu_911 = (65.519^3)*D_r*(6.27E-22);

a1143_r_state = [8.186474113570364E+07; -8.496970857794180E+08;
3.796881314169592E+07];
a1143_v_state = [1.021264101530404E+06;  9.596382317667111E+04;
3.306657723792610E+04]./(24*60*60);
mu_1143 = (57.312^3)*D_r*(6.27E-22);

a1437_r_state = [5.274744511932679E+08; -6.019567011278348E+08; -
2.415844826279843E+07];
```

```
a1437_v_state = [7.562533676067480E+05;  6.929992215881000E+05;
3.826520588987247E+05]./(24*60*60);
mu_1437 = (58.893^3)*D_r*(6.27E-22);

a1583_r_state = [2.499256283113326E+08; -6.504355873888571E+08;
3.551401488780757E+08];
a1583_v_state = [9.763525664834852E+05;  5.160115523105981E+05;
1.408882838781204E+05]./(24*60*60);
mu_1583 = (54.421^3)*D_r*(6.27E-22);

a2260_r_state = [1.177555141550063E+08; -7.857808252001264E+08; -
5.270307097159702E+07];
a2260_v_state = [1.028273974224956E+06;  2.153968212561019E+05; -
3.252891727169752E+05]./(24*60*60);
mu_2260 = (38.2175^3)*D_r*(6.27E-22);

%% Jupiter Trojans - L5 Trojan Camp
a617_r_state = [-7.905069409017330E+08;  2.617417756999413E+08;
2.994002297180880E+08];
a617_v_state = [-3.896443556418308E+05; -8.917615697460237E+05; -
1.483632715063775E+05]./(24*60*60);
mu_617 = (70.181^3)*P_r*(6.27E-22);

a884_r_state = [-7.376245092569276E+08;  3.410365708979864E+08; -
7.072280568383938E+07];
a884_v_state = [-3.254315973608705E+05; -1.014141878117213E+06; -
1.266830901289410E+05]./(24*60*60);
mu_884 = (50.5465^3)*D_r*(6.27E-22);

a1172_r_state = [-7.586813941659168E+08;  2.558163659225389E+08; -
2.390503038922518E+08];
a1172_v_state = [-2.669667544302975E+05; -1.013930861676004E+06;
4.360678464993227E+04]./(24*60*60);
mu_1172 = (59.01^3)*D_r*(6.27E-22);

a1173_r_state = [-7.133430748756920E+08;  5.467343688577291E+08; -
6.815890629562208E+07];
a1173_v_state = [-5.879850450880636E+05; -7.709644533758684E+05; -
9.166170149895073E+04]./(24*60*60);
mu_1173 = (49.7745^3)*P_r*(6.27E-22);

a1208_r_state = [-7.397313328743825E+08;  1.144583848891697E+08;
4.174928794663525E+08];
a1208_v_state = [-2.934144378309696E+05; -9.437819280041645E+05; -
2.691555617613096E+05]./(24*60*60);
mu_1208 = (50.2385^3)*F_r*(6.27E-22);

a1867_r_state = [-6.634915311789105E+08;  3.018946817644486E+08; -
2.911733616662771E+08];
a1867_v_state = [-3.013372686565857E+05; -1.032951353301762E+06; -
2.725080036324002E+05]./(24*60*60);
mu_1867 = (59.11^3)*D_r*(6.27E-22);

a2207_r_state = [-7.571525793594217E+08;  1.051353742157169E+08;
2.055492448367196E+07];
```

```
a2207_v_state = [-1.697221001615341E+05; -1.120512507878241E+06;
1.322532917269408E+05]./(24*60*60);
mu_2207 = (48.829^3)*D_r*(6.27E-22);

a2223_r_state = [-7.604454844948108E+08;  1.070236955466223E+08; -
1.653475197566518E+08];
a2223_v_state = [-1.790471268525216E+05; -1.087509752661243E+06;
2.019919389289297E+05]./(24*60*60);
mu_2223 = (38.74^3)*D_r*(6.27E-22);

a2241_r_state = [-5.507199004788246E+08;  4.744758928592425E+08; -
1.694123253691942E+08];
a2241_v_state = [-6.680311201553114E+05; -9.464592482654349E+05; -
1.891688391086984E+05]./(24*60*60);
mu_2241 = (56.841^3)*D_r*(6.27E-22);

a2357_r_state = [-7.379716256753871E+08;  2.495670234861175E+08; -
1.115689240967929E+07];
a2357_v_state = [-3.128847906684666E+05; -1.081924335719941E+06;
5.067616987351819E+04]./(24*60*60);
mu_2357 = (47.3125^3)*D_r*(6.27E-22);

a2363_r_state = [-7.373687462707223E+08;  7.401157588513255E+06; -
2.472673364553505E+08];
a2363_v_state = [-1.325372062225918E+05; -1.005625218443398E+06;
4.941234390256676E+05]./(24*60*60);
mu_2363 = (47.988^3)*D_r*(6.27E-22);

a2674_r_state = [-6.889139178591666E+08;  2.896157834732346E+08; -
9.295247518209442E+06];
a2674_v_state = [-3.898565600438613E+05; -1.104027869513602E+06;
3.584022805886861E+04]./(24*60*60);
mu_2674 = (37.1335^3)*D_r*(6.27E-22);

a2893_r_state = [-6.234394083456531E+08;  5.097486433863056E+08;
1.116548543132261E+08];
a2893_v_state = [-5.942150926520568E+05; -8.663755982037102E+05;
2.200061208450217E+05]./(24*60*60);
mu_2893 = (43.442^3)*D_r*(6.27E-22);

a3317_r_state = [-7.304864623519613E+08;  3.996821979922485E+08;
1.174761040219435E+08];
a3317_v_state = [-2.952849453012048E+05; -8.943377631924625E+05;
4.481691057718208E+05]./(24*60*60);
mu_3317 = (59.395^3)*T_r*(6.27E-22);

mu_n = [mu_sun; mu_mercury; mu_venus; mu_earth; mu_mars; ...
    mu_jupiter; mu_saturn; mu_uranus; mu_neptune; ...
    mu_ceres; mu_pallas; mu_vesta; ...
    mu_334; mu_153; mu_190; mu_361; mu_499; mu_748; ...
    mu_1038; mu_1162; mu_1180; mu_1212; mu_1268; ...
    mu_1269; mu_1345; mu_1439; mu_1512; mu_1529; mu_1578; mu_1746;
mu_1748; ...
    mu_1754; mu_1902; mu_1911; mu_2067; mu_2246; mu_2312; mu_2760; mu_466;  ...
    mu_1144; mu_1256; mu_588; mu_624; mu_659; mu_911; mu_1143; mu_1437;
mu_1583; ...
    mu_2260; mu_617; mu_884; mu_1172; mu_1173; mu_1208; mu_1867; mu_2207;
mu_2223; ...
    mu_2241; mu_2357; mu_2363; mu_2674; mu_2893; mu_3317];

%% Orbit Propagator

a_body_n = zeros(3,9);
a_body_update = zeros(3,9);

% Main bodies
state_init(1:6,1) = [sun_r_state;sun_v_state];
state_init(1:6,2) = [mercury_r_state;mercury_v_state];
state_init(1:6,3) = [venus_r_state;venus_v_state];
state_init(1:6,4) = [earth_r_state;earth_v_state];
state_init(1:6,5) = [mars_r_state;mars_v_state];
state_init(1:6,6) = [jupiter_r_state;jupiter_v_state];
state_init(1:6,7) = [saturn_r_state;saturn_v_state];
state_init(1:6,8) = [uranus_r_state;uranus_v_state];
state_init(1:6,9) = [neptune_r_state;neptune_v_state];
state_init(1:6,10) = [ceres_r_state;ceres_v_state];
state_init(1:6,11) = [pallas_r_state;pallas_v_state];
state_init(1:6,12) = [vesta_r_state;vesta_v_state];

% Hilda asteroids
state_init(1:6,13) = [a334_r_state;a334_v_state];
state_init(1:6,14) = [a153_r_state;a153_v_state];
state_init(1:6,15) = [a190_r_state;a190_v_state];
state_init(1:6,16) = [a361_r_state;a361_v_state];
state_init(1:6,17) = [a499_r_state;a499_v_state];
state_init(1:6,18) = [a748_r_state;a748_v_state];
state_init(1:6,19) = [a1038_r_state;a1038_v_state];
state_init(1:6,20) = [a1162_r_state;a1162_v_state];
state_init(1:6,21) = [a1180_r_state;a1180_v_state];
state_init(1:6,22) = [a1212_r_state;a1212_v_state];
state_init(1:6,23) = [a1268_r_state;a1268_v_state];
state_init(1:6,24) = [a1269_r_state;a1269_v_state];
state_init(1:6,25) = [a1345_r_state;a1345_v_state];
state_init(1:6,26) = [a1439_r_state;a1439_v_state];
state_init(1:6,27) = [a1512_r_state;a1512_v_state];
state_init(1:6,28) = [a1529_r_state;a1529_v_state];
state_init(1:6,29) = [a1578_r_state;a1578_v_state];
state_init(1:6,30) = [a1746_r_state;a1746_v_state];
state_init(1:6,31) = [a1748_r_state;a1748_v_state];
state_init(1:6,32) = [a1754_r_state;a1754_v_state];
state_init(1:6,33) = [a1902_r_state;a1902_v_state];
state_init(1:6,34) = [a1911_r_state;a1911_v_state];
state_init(1:6,35) = [a2067_r_state;a2067_v_state];
state_init(1:6,36) = [a2246_r_state;a2246_v_state];
state_init(1:6,37) = [a2312_r_state;a2312_v_state];
state_init(1:6,38) = [a2760_r_state;a2760_v_state];
state_init(1:6,39) = [a466_r_state;a466_v_state];
state_init(1:6,40) = [a1144_r_state;a1144_v_state];
state_init(1:6,41) = [a1256_r_state;a1256_v_state];
```

```
% Jupiter Trojans - L4 Greek Camp
state_init(1:6,42) = [a588_r_state;a588_v_state];
state_init(1:6,43) = [a624_r_state;a624_v_state];
state_init(1:6,44) = [a659_r_state;a659_v_state];
state_init(1:6,45) = [a911_r_state;a911_v_state];
state_init(1:6,46) = [a1143_r_state;a1143_v_state];
state_init(1:6,47) = [a1437_r_state;a1437_v_state];
state_init(1:6,48) = [a1583_r_state;a1583_v_state];
state_init(1:6,49) = [a2260_r_state;a2260_v_state];

% Jupiter Trojans - L5 Trojan Camp
state_init(1:6,50) = [a617_r_state;a617_v_state];
state_init(1:6,51) = [a884_r_state;a884_v_state];
state_init(1:6,52) = [a1172_r_state;a1172_v_state];
state_init(1:6,53) = [a1173_r_state;a1173_v_state];
state_init(1:6,54) = [a1208_r_state;a1208_v_state];
state_init(1:6,55) = [a1867_r_state;a1867_v_state];
state_init(1:6,56) = [a2207_r_state;a2207_v_state];
state_init(1:6,57) = [a2223_r_state;a2223_v_state];
state_init(1:6,58) = [a2241_r_state;a2241_v_state];
state_init(1:6,59) = [a2357_r_state;a2357_v_state];
state_init(1:6,60) = [a2363_r_state;a2363_v_state];
state_init(1:6,61) = [a2674_r_state;a2674_v_state];
state_init(1:6,62) = [a2893_r_state;a2893_v_state];
state_init(1:6,63) = [a3317_r_state;a3317_v_state];

[~,bodies] = size(state_init);
seed_flag = 1;
[t_step, state_result] = ode113(@EOM, t_span, state_init(:,:),opts);
iterations = size(state_result);

state_result = state_result.';
state_result = reshape(state_result,6,bodies,iterations(1));

figure
hold all
axis tight manual
axis equal
grid on
xlim([-4.495E9 4.495E9])
ylim([-4.495E9 4.495E9])
zlim([-1E9 1E9])
pbaspect([1 1 1])
axis vis3d
xlabel('x');
ylabel('y');
zlabel('z');
set(gcf,'Renderer','OpenGL');

for i = 1:bodies
    if i == 1
        h_line(i) =
plot3(squeeze(state_result(1,i,1)),squeeze(state_result(2,i,1)),squeeze(state
_result(3,i,1)),'o','MarkerSize',5,'MarkerFaceColor','k'); %#ok<SAGROW>
    elseif i>=2 && i < 10
```

```
        h_line(i) = 
plot3(squeeze(state_result(1,i,1)),squeeze(state_result(2,i,1)),squeeze(state
_result(3,i,1)),'o','MarkerSize',5,'MarkerFaceColor','r'); %#ok<SAGROW>
    elseif i>=10 && i<13
        h_line(i) = 
plot3(squeeze(state_result(1,i,1)),squeeze(state_result(2,i,1)),squeeze(state
_result(3,i,1)),'o','MarkerSize',5,'MarkerFaceColor','g'); %#ok<SAGROW>
    elseif i>=13 && i<hilda_bodies+12+1
        h_line(i) = 
plot3(squeeze(state_result(1,i,1)),squeeze(state_result(2,i,1)),squeeze(state
_result(3,i,1)),'o','MarkerSize',5,'MarkerFaceColor','b'); %#ok<SAGROW>
    elseif i>=hilda_bodies+12+1
        h_line(i) = 
plot3(squeeze(state_result(1,i,1)),squeeze(state_result(2,i,1)),squeeze(state
_result(3,i,1)),'o','MarkerSize',5,'MarkerFaceColor','y'); %#ok<SAGROW>
    end
end

an_loop = 1;
while an_loop <= iterations(1)
    for i = 1:bodies
        
set(h_line(i),'XData',state_result(1,i,an_loop),'YData',state_result(2,i,an_l
oop),'ZData',state_result(3,i,an_loop));
    end
    drawnow;
    an_loop = an_loop + 1;
end

figure
hold all
axis tight manual
axis equal
grid on
xlim([-4.495E9 4.495E9])
ylim([-4.495E9 4.495E9])
zlim([-1E9 1E9])
pbaspect([1 1 1])
axis vis3d
xlabel('x');
ylabel('y');
zlabel('z');
set(gcf,'Renderer','OpenGL');

for i = 1:bodies
    if i == 1
        h_animated(i) = 
animatedline('MaximumNumPoints',1000,'Color','k'); %#ok<SAGROW>
    elseif i>=2 && i < 10
        h_animated(i) = 
animatedline('MaximumNumPoints',1000,'Color','r'); %#ok<SAGROW>
    elseif i>=10 && i<13
        h_animated(i) = 
animatedline('MaximumNumPoints',1000,'Color','g'); %#ok<SAGROW>
    elseif i>=13 && i<hilda_bodies+12+1
```

```
        h_animated(i) =
animatedline('MaximumNumPoints',1000,'Color','b'); %#ok<SAGROW>
    elseif i>=hilda_bodies+12+1
        h_animated(i) =
animatedline('MaximumNumPoints',1000,'Color','y'); %#ok<SAGROW>
    end
end

for k = 1:iterations(1)
    for i = 1:bodies

addpoints(h_animated(i),state_result(1,i,k),state_result(2,i,k),state_result(
3,i,k));
    end
    drawnow
end

%% Orbit EOM
function ydot = EOM(t,state) %#ok<INUSL>

    global mu_n beta gamma seed_flag
    global c_constant bodies hilda_bodies trojan_bodies
    global a_body_n a_body_update

    dv_final = zeros(1,1);
    dr = zeros(1,1);
    r_body_n = zeros(3,bodies);
    v_body_n = zeros(3,bodies);

    for i = 1:bodies

        index_i = 1;
        index_j = 3;
        for index_n = 1:bodies
            r_body_n(1:3,index_n) = state(index_i:index_j);
            index_i = index_i + 3;
            index_j = index_j + 3;
            v_body_n(1:3,index_n) = state(index_i:index_j);
            index_i = index_i + 3;
            index_j = index_j + 3;
        end

        if seed_flag == 1
            for seed_i = 1:9
                dv_update = 0;

                for j = 1:9
                    if j~=seed_i
                        dv = (mu_n(j).*(r_body_n(1:3,j)-
r_body_n(1:3,seed_i))./ ...
                            (norm(r_body_n(1:3,j)-r_body_n(1:3,seed_i))^3));
                        dv_update = dv + dv_update;
                    end
                end
                a_body_n(1:3,seed_i) = dv_update;
            end
            seed_flag = 0;
        end

        if i<10
            target = 'planet';
        elseif i>=10 && i<13
            target = 'asteroid';
        elseif i>=13 && i<hilda_bodies+12+1
            target = 'hilda';
        elseif i>=hilda_bodies+12+1 && i<trojan_bodies+hilda_bodies+12+1
            target = 'trojans';
        end

        r_i = r_body_n(1:3,i);
        v_i = v_body_n(1:3,i);
        sum1_update = 0;
        sum2_update = 0;
        sum3_update = 0;
        sum4_update = 0;
        sum5_update = 0;
        dv_update = 0;
        inner_sum1_update = 0;
        inner_sum2_update = 0;
        a = (2*(beta+gamma))/(c_constant^2);
        b = ((2*beta)-1)/(c_constant^2);
        c = (2*(1+gamma))/(c_constant^2);

        % DE405 Point Mass Equations
        dr(1:3,i) = v_i;

        switch target
            case 'planet'
                for j = 1:9
                    if j~=i
                        d = r_i-r_body_n(1:3,j);
                        e = r_body_n(1:3,j)-r_i;

                        for ki = 1:9
                            if ki~=i
                                inner_sum1 = mu_n(ki)/norm(r_body_n(1:3,ki)-
r_i);

                                inner_sum1_update = inner_sum1 +
inner_sum1_update;
                            end
                        end
                        inner_sum1 = inner_sum1_update;

                        for kj = 1:9
                            if kj~=j
                                inner_sum2 = mu_n(kj)/norm(r_body_n(1:3,kj)-
r_body_n(1:3,j));
                                inner_sum2_update = inner_sum2 +
inner_sum2_update;
                            end
                        end
                        inner_sum2 = inner_sum2_update;

                        inner1 = 1-(a*inner_sum1)-
(b*inner_sum2)+(gamma*((norm(v_i)/c_constant)^2)) ...
                            +
((1+gamma)*((norm(v_body_n(1:3,j))/c_constant)^2)) -
((dot(v_i,v_body_n(1:3,j)))*c) ...
                            -
((3/(2*(c_constant^2)))*((dot(d,v_body_n(1:3,j))/norm(r_body_n(1:3,j)-
r_i))^2)) ...
                            +
(dot(e,a_body_n(1:3,j))*(1/(2*(c_constant^2))));

                        sum1 = ((mu_n(j)*(r_body_n(1:3,j)-
r_i))/(norm(r_body_n(1:3,j)-r_i)^3))*inner1;
                        sum1_update = sum1 + sum1_update;
                    end
                end
                sum1 = sum1_update;

                for j = 1:9
                    if j~=i
                        f = r_i-r_body_n(1:3,j);
                        g = ((2+(2*gamma))*v_i)-
((1+(2*gamma))*v_body_n(1:3,j));
                        inner2 = dot(f,g).*(v_i-v_body_n(1:3,j));
                        sum2 = (mu_n(j)/(norm(r_body_n(1:3,j)-
r_i)^3))*inner2;
                        sum2_update = sum2 + sum2_update;
                    end
                end
                sum2 = (1/(c_constant^2))* sum2_update;

                h = (3+(4*gamma))/(2*(c_constant^2));
                for j = 1:9
                    if j~=i
                        sum3 =
(mu_n(j).*a_body_n(1:3,j))./norm(r_body_n(1:3,j)-r_i);
                        sum3_update = sum3 + sum3_update;
                    end
                end
                sum3 = h.*sum3_update;

                for m = 10:12
                    sum4 = (mu_n(m).*(r_body_n(1:3,m)-
r_i))./(norm(r_body_n(1:3,m)-r_i)^3);
                    sum4_update = sum4 + sum4_update;
                end
                sum4 = sum4_update;

                if j == 4 || j == 5
                    for m = 13:bodies
                        sum4 = (mu_n(m).*(r_body_n(1:3,m)-
r_i))./(norm(r_body_n(1:3,m)-r_i)^3);
                        sum5_update = sum5 + sum5_update;
                    end
                else
                    sum5_update = 0;
                end
                sum5 = sum5_update;

                dv = sum1+sum2+sum3+sum4+sum5;
                dv_final(1:3,i) = dv;

                a_body_update(1:3,i) = dv;

            case 'asteroid'
                for j = 1:12
                    if j~=i
                        dv = (mu_n(j).*(r_body_n(j)-r_i))./(norm(r_body_n(j)-
r_i)^3);
                        dv_update = dv + dv_update;
                    end
                end
                dv = dv_update;
                dv_final(1:3,i) = dv;

            case 'hilda'
                for j = 1:bodies
                    if j~=i
                        dv = (mu_n(j).*(r_body_n(j)-r_i))./(norm(r_body_n(j)-
r_i)^3);
                        dv_update = dv + dv_update;
                    end
                end
                dv = dv_update;
                dv_final(1:3,i) = dv;

            case 'trojans'
                for j = 1:bodies
                    if j~=i
                        dv = (mu_n(j).*(r_body_n(j)-r_i))./(norm(r_body_n(j)-
r_i)^3);
                        dv_update = dv + dv_update;
                    end
                end
                dv = dv_update;
                dv_final(1:3,i) = dv;
        end
    end

    a_body_n = a_body_update;
    ydot_end = [dr;dv_final];

    ydot = reshape(ydot_end,bodies*6,1);
end
```