Preliminary Orbital Determination of an Earth Orbiting Satellite Using Gauss' Angles Only Method

Amanda Kerlee, Blake Iwaisako, & Miles Puchner
Department of Mechanical and Aerospace Engineering, University of California - San Diego
MAE 180A: Spacecraft Guidance
Professor Rosengren
17 March 2023

Abstract

As space becomes more filled with defunct satellites and debris, collisions become increasingly likely. The collisions cause damage to active satellites and can set research behind. This software aims to solve this problem by propagating the orbits of various satellites and pieces of debris so that the resulting data can be used to plan orbits of new satellites that ensure they are not in the same place as existing satellites at the same time. This software can also be used in advance to determine whether an existing satellite is at risk of collision and provide ample time to determine an appropriate course of action. This is accomplished by calculating the orbital precession due to secular changes in the longitude of the ascending node and the argument of periapsis and the secular changes to the mean anomaly between spherical and oblate Earth models.

Introduction

As space becomes increasingly cluttered, the need to avoid collision becomes more critical than ever. Collisions can cause irreparable damage to probes and end their usability even if they were previously functional. According to NASA, the Hubble Telescope was originally only intended to be in use for 15 years, but combined with upgrades and a lack of notable collision damage, it has remained invaluable to research for over 30 years. Orbit propagation is used to determine the location of objects at a certain time. Orbit propagation determines the future location of an orbiting object based on data from previous observations of the object. Orbit propagation can be achieved using various methods. Two of these methods are the Laplace and Gauss methods. These methods can be used for unperturbed cases such as solving the Kepler problem between two spherical bodies, or made more accurate by accounting for perturbations from other bodies such as the sun and moon, or even account for oblateness (when a body like Earth isn't perfectly spherical). This software works to resolve orbits using angles only Gauss methods and accounting for the changes in the mean anomaly, longitude of the ascending node, and the argument of periapsis of an object's orbit due to the Earth's oblateness. Knowing the fraction of the orbit's period that has elapsed since passing the point of closest approach, angle between the reference frame and the line where the orbit crosses a body's equatorial plane from north to south, and the angle from the ascending node to the point of closest approach are crucial to increasing the accuracy of orbit propagation from solving Kepler's problem as the changes in these metrics help to define how an object's orbit changes due to perturbations and oblateness. This software is accurate to ten digits after the decimal.

Orbit Determination

This software uses the Gibbs variant of Gauss' method of preliminary orbit determination to determine the orbital elements and state vectors using an initial observation and a propagated position made of three sets of angular measurements. Gauss' method uses an analytical approximation utilizing the expected geometry of conic sections when modeling. Typical Gauss propagation algorithms are rather robust, however, may fail unexpectedly. The initial observation is made from a given site latitude (lat), 3 local sidereal times (lst, 3x1 vector), altitude of observation site (alt), and three sets each of right ascension (ra, 3x1 vector), declination (dec, 3x1 vector), and julian dates (JD, 3x1 vector) corresponding to each observation.

First we define the site vectors at each observation using the local sidereal times and the site latitude, this will change depending on Earth's flatness (spherical or oblate model). Next we define the LOS matrix based on the right ascension (ra) and declination (dec) angles for each observation. Using the time

between each observation we can then determine matrix M from the inverse LOS matrix and r_site vectors. Using the intermediate variables establish the 8th degree polynomial and solve for its real positive root. Then define c values from the polynomial solution and use it to determine the initial slant vectors. Lastly, we use an iterative refinement method that uses the current set of slant values to calculate the positions vectors of each observation to calculate the velocity vector at the second observation, the refined orbital elements, the refined lagrange coefficients, new c values, and refined slant vectors. Each new and refined slant vector is compared with its predecessor until they are equal up to 10 decimal places or until 100 iterations have been achieved.

ANGLES-ONLY GAUSS

$$[M] = [L^{-1}][r_{site}]$$

$$d_1 = M_{21}a_1 - M_{22} + M_{23}a_{23};\ d_2 = M_{21}a_{1u} + M_{23}a_{3u}$$

$$C = \hat{L}_2 \cdot \vec{r}_{site_2}$$

$$r_2^{\,8} - (d_1^{\,2} + 2Cd_1 + r_{site_2}^{\,2})r_2^{\,6} - 2\mu(Cd_2 + d_1d_2)r_2^{\,3} - \mu^2 d_2^{\,2} = 0 \rightarrow u = \mu/r_2^{\,3}$$

$$c_1 = a_1 + a_{1u}u;\ c_2 = -1;\ c_3 = a_3 + a_{3u}u$$

$$[c_1\rho_1,\ c_2\rho_2,\ c_3\rho_3] = [M][-c_1, -c_2, -c_3]$$

Iteration method to refine the initial estimate of slant ranges

$$\vec{r}_i = \rho_i \hat{L}_i + \vec{r}_{site_i} \qquad i = 1, 2, 3.....$$

Once the refined state vector has been achieved a standard ECI orbital elements algorithm is used to find the initial orbital elements. First the 5 propagated orbital elements are found and since we use the oblate Earth model, the first 3 elements do not change and the 4th and 5th elements change based on their secular changes due to J2 (Earth's oblateness). Next we find the 6th orbital element (true anomaly) using a Kepler's problem algorithm. The mean anomaly changes based on both the mean motion n and its secular change based on J2, then using the standard Kepler algorithm, we find the corresponding eccentric anomaly and then convert that into the true anomaly. Lastly we use a standard orbital elements to ECI algorithm to find the propagated state's corresponding state vector. As a result, the program has found the initial state vector (r0, v0) and the initial orbital elements (oe0) as well as the propagated state vector (rf, vf) and the propagated orbital elements (oef).

$$\textbf{\textit{GIBBS}}\,(\overline{r_1}\,\overline{r_2}\,\overline{r_3}\ in\ IJK \Rightarrow \overline{v_2})$$

$$\overline{N} = \overline{r_1}\,\overline{Z_{23}} + \overline{r_2}\,\overline{Z_{31}} + \overline{r_3}\,\overline{Z_{12}}$$

$$\overline{D} = \overline{Z_{12}} + \overline{Z_{23}} + \overline{Z_{31}}$$

$$\overline{S} = (r_2 - r_3)\,\overline{r_1} + (r_3 - r_1)\,\overline{r_2} + (r_1 - r_2)\,\overline{r_3}$$

$$\overline{B} = \overline{D} \times \overline{r_2}$$

$$L_g = \sqrt{\frac{\mu}{ND}}$$

$$\overline{v_2} = \frac{L_g}{r_2}\overline{B} + L_g\overline{S}$$

HERRICK-GIBBS $(\overline{r_1}, \overline{r_2}, \overline{r_3}, JD_1, JD_2, JD_3)$

$$\Delta t_{31} = JD_3 - JD_1$$
$$\Delta t_{32} = JD_3 - JD_2$$
$$\Delta t_{21} = JD_2 - JD_1$$
$$\overline{v_2} = -\Delta t_{32}\left(\frac{1}{\Delta t_{21}\Delta t_{31}} + \frac{\mu}{12r_1^3}\right)\overline{r_1} + \Delta t_{21}\left(\frac{1}{\Delta t_{32}\Delta t_{31}} + \frac{\mu}{12r_3^3}\right)\overline{r_3}$$

The Gauss Gibbs and Herrick-Gibbs methods are used depending on the angle between the input position vectors and coplanarity to find r2 (r0) and velocity at r2 (v0). If the angles between all combinations of position vectors are greater than 1 and they are coplanar, Gibbs method will be used. If the angles are not greater than 1, Herrick-Gibbs method will be used. In the case that the orbits are not coplanar an error will occur and no output will be returned. Once determined, the observation state vector will be converted into the initial orbital elements using the above equations (oe0, a 6x1 array containing a, e, i, cap omega, low omega, f). The changes in the longitude of the ascending node and argument of periapsis due to the Earth's oblateness are determined and used to find their propagated counterparts. Kepler's algorithm is used to determine propagated true anomaly from the change in mean motion and secular mean anomaly due to Earth's oblateness (oef, a 6x1 array). Finally the propagated orbital elements are used to find the corresponding propagated state vectors (rf) and velocity (vf).

Using the data given to us from our corresponding group, we calculated the approach distance between our two propagated positions. This is done by taking the magnitude of the difference between the two position vectors. Since the other group used the spherical Earth model, we calculated the closest approach using a modified spherical model of our own.

Test Results

The main file contains the main function 'OrbitComp' and the master live script which outlines the data and calls the above function. Within the master live script there are a total of 8 defined input variables: lat, lst, alt, ra, dec, JD, JD_prop, and model. The order listed is the order for which they should be inputted into the 'OrbitComp' function.

The variable lat is type 1x1 double and contains the site latitude; it should have units of degrees. The variable lst is type 3x1 double that contains the 3 local sidereal times that correspond to each respective observation; it should have units of degrees. The variable alt is type 1x1 double and is the altitude the site is from sea level; it should have units of meters. The variable ra is type 3x1 double that contains the 3 right ascension angles corresponding to each observation; it should have units of degrees. The variable dec is type 3x1 double that contains the 3 declination angles corresponding to each observation; it should have units of degrees. The variable JD is type 3x1 double that contains the 3 julian dates corresponding to each observation; julian dates have their own continuous time unit. The variable JD_prop is type 1x1 double and contains the julian date corresponding to the desired propagated position; julian dates have their own continuous time unit. Lastly, the model is type char and contains the desired Earth model to be used when calculating specific values; it should be set to 'oblate' if an oblate spheroid

Earth model is desired or it should be set to 'sphere' if a spherical Earth model is desired. It should be noted that all input variables that are type 3x1 double have their respective components listed in chronological order, that is, position 1 in the array corresponds to the first observation and position 3 corresponds to the last observation. A summary of the input variables is listed below in Table 1.

Table 1: Input Variables Summary

NAME	lat	lst	alt	ra	dec	JD	JD_prop	model
TYPE	1x1 double	3x1 double	1x1 double	3x1 double	3x1 double	3x1 double	1x1 double	char
DESCRIPTION	Site Latitude	Site Local Sidereal Times	Site Altitude	Right Ascension Angles	Declination Angles	Initial Julian Dates	Propagated Julian Date	Earth Model Desired
UNITS	degrees	degrees	meters	degrees	degrees	N/A	N/A	N/A

After the 'OrbitComp' function is run according to the above input variables, it will output the following 6 variables: r0, v0, oe0, rf, vf, and oef. This order is also the order specified by the master live script file and should not be deviated from.

The variable r0 is type 3x1 double that contains the position vector in the Earth-Centered Inertial (ECI) frame corresponding to the second observation; it will have units of kilometers. The variable v0 is type 3x1 double that contains the velocity vector in the ECI frame corresponding to the second observation; it will have units of kilometer per second. The variable oe0 is type 6x1 double that contains the 6 keplerian orbital elements corresponding to the second observation; units of its components vary. The variable rf is type 3x1 double that contains the position vector in the ECI frame corresponding to the propagated position; it will have units of kilometers. The variable vf is type 3x1 double that contains the velocity vector in the ECI frame corresponding to the propagated position; it will have units of kilometer per second. The variable oef is type 6x1 double that contains the 6 keplerian orbital elements corresponding to the propagated position; units of its components vary. The components of oe0 and oef are in the following order: semimajor axis in units of kilometers, eccentricity which is unitless, inclination in units of degrees, longitude of ascending node in units of degrees, argument of periapsis in units of degrees, and the true anomaly in units of degrees. A summary of the output variables is listed below in Table 2.

Table 2: Output Variables Summary

NAME	r0	v0	oe0	rf	vf	oef
TYPE	3x1 double	3x1 double	6x1 double	3x1 double	3x1 double	6x1 double
DESCRIPTION	Obs. 2 ECI Position Vector	Obs. 2 ECI Velocity Vector	Obs. 2 Orbital Elements	Propagated ECI Position Vector	Propagated ECI Velocity Vector	Propagated Orbital Elements
UNITS	kilometers	kilometers per second	Units Vary	kilometers	kilometers per second	Units Vary

The input data for our specific satellite can be seen in Table 3 below and follows the variable type and unit conventions outlined above. It should be noted that Table 3 is split up into two sections, Table 3-A corresponds to variable type 1x1 doubles and Table 3-B corresponds to variable type 3x1 doubles.

Table 3-A: 1x1 Double Input Variables

Site Latitude (lat) [deg]	Site Altitude (alt) [m]	Propagated Julian Date (JD_prop)	Desired Model (model)
32.881191	111	2454873.205555555	* 'oblate' OR 'sphere' *

* The model variable was defined based on the desired Earth model and was ran both ways for data comparison

Table 3-B: 3x1 Double Input Variables

Observation Number	Local Sidereal Time (lst) [deg]	Right Ascension (ra) [deg]	Declination (dec) [deg]	Julian Dates (JD)
1	33.819170038404	12.106879490506	-21.086486207362	2454870.535518990
2	34.007183385032	14.111774377468	-9.829593672319	2454870.536039823
3	34.195196732076	16.598318338656	6.093565604026	2454870.536560657

The above input data was run twice, once using an oblate-spheroid Earth model and once using a spherical Earth model. Results for the oblate-spheroid Earth model are displayed in Tables 4-A and 4-B.

Table 4-A: ECI Position and Velocity Vectors Using Oblate Spheroid Earth Model

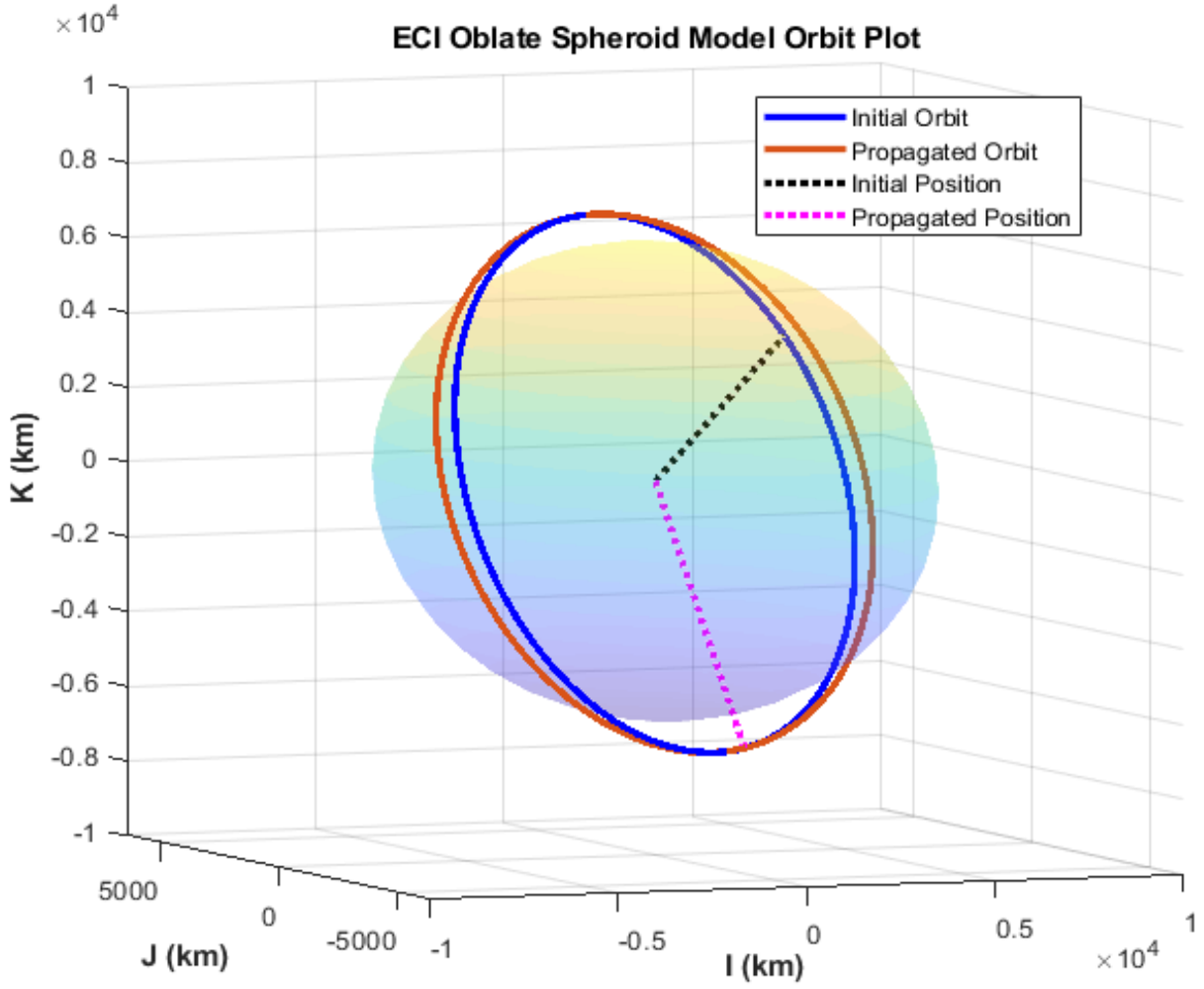
	r0 [km]	v0 [km/s]	rf [km]	vf [km/s]
I	5481.477815518792	-3.996939988497132	1312.734596157754	6.917469551886585
J	3259.446815708850	0.314058554565458	-1671.013044904947	2.430145794978484
K	3257.736058177104	6.331930313751493	-6962.278679095411	0.777444518992604
Magnitude	7161.245479049474	7.494498218396498	7279.346201519646	7.373019297834588

Table 4-B: Keplerian Orbital Elements Using Oblate Spheroid Earth Model

	oe0	oef
a [km]	7227.040218788220	7227.040218788220
e	0.010292692130	0.010292692130
***i* [deg]**	74.048806685722	74.048806685722
Ω [deg]	22.341351289173	17.619902692679
ω [deg]	56.322977606921	50.976706124031
***f* [deg]**	331.914636974993	224.900098904865

Figure 1 shows the graphical representation of the initial and propagated orbits and positions for the ECI oblate-spheroid Earth model.

Figure 1



Tables 5-A and 5-B display the results for the spherical Earth model in the ECI reference frame.

Table 5-A: ECI Position and Velocity Vectors Using Spherical Earth Model

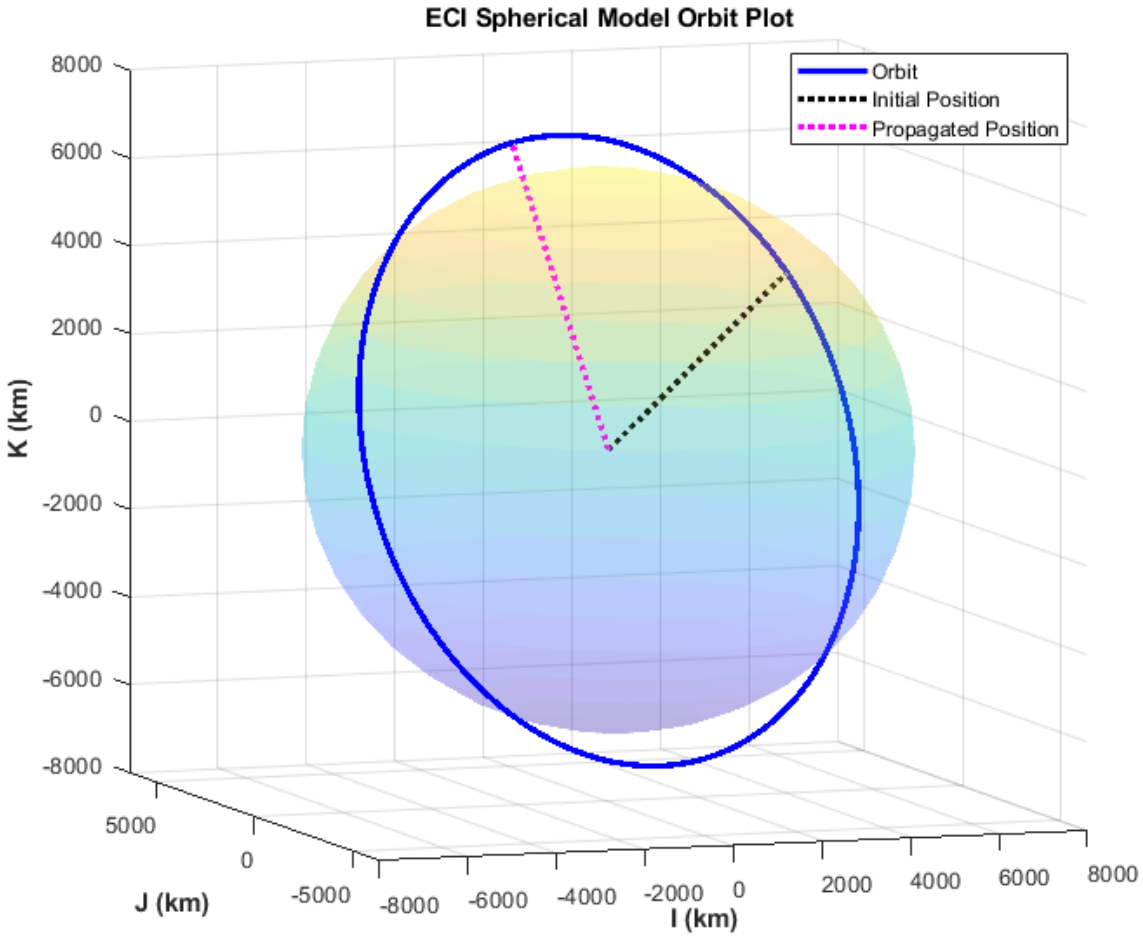
	r0 [km]	v0 [km/s]	rf [km]	vf [km/s]
I	5472.430831133358	-3.983284413790064	-1527.48947216074 5	-6.773763769352308
J	3255.315316649736	0.312859863432937	1481.286361606157	-3.040981245129802
K	3278.351902869471	6.303884933444033	6827.698310557311	-0.855630682249662
Magnitude	7161.855081709011	7.463471127390580	7151.566093704894	7.474192023264335

Table 5-B: Keplerian Orbital Elements Using Spherical Earth Model

	oe0	oef
a [km]	7167.955786767737	7167.955786767737
e	0.002286529540	0.002286529540
***i* [deg]**	74.060726458607	74.060726458607
Ω [deg]	22.290982181372	22.290982181372
ω [deg]	96.696648198697	96.696648198697
***f* [deg]**	291.731382748123	0.141258509996

Figure 2 displays the initial and propagated positions relative to a spherical Earth model. It should be noted that only one orbit is displayed as changes in the orbital elements due to Earth's oblateness are not considered for a spherical model.

Figure 2



Finally, results are concluded with the position and velocity vectors of a 'compare' satellite provided by a separate group. The values are in Table 6 below.

Table 6: State Vector of the 'Compare' Satellite

Position Vector r [km]	Velocity Vector v [km/s]
-1329.016	3.73
1315.305	-6.185
6900.261	1.903

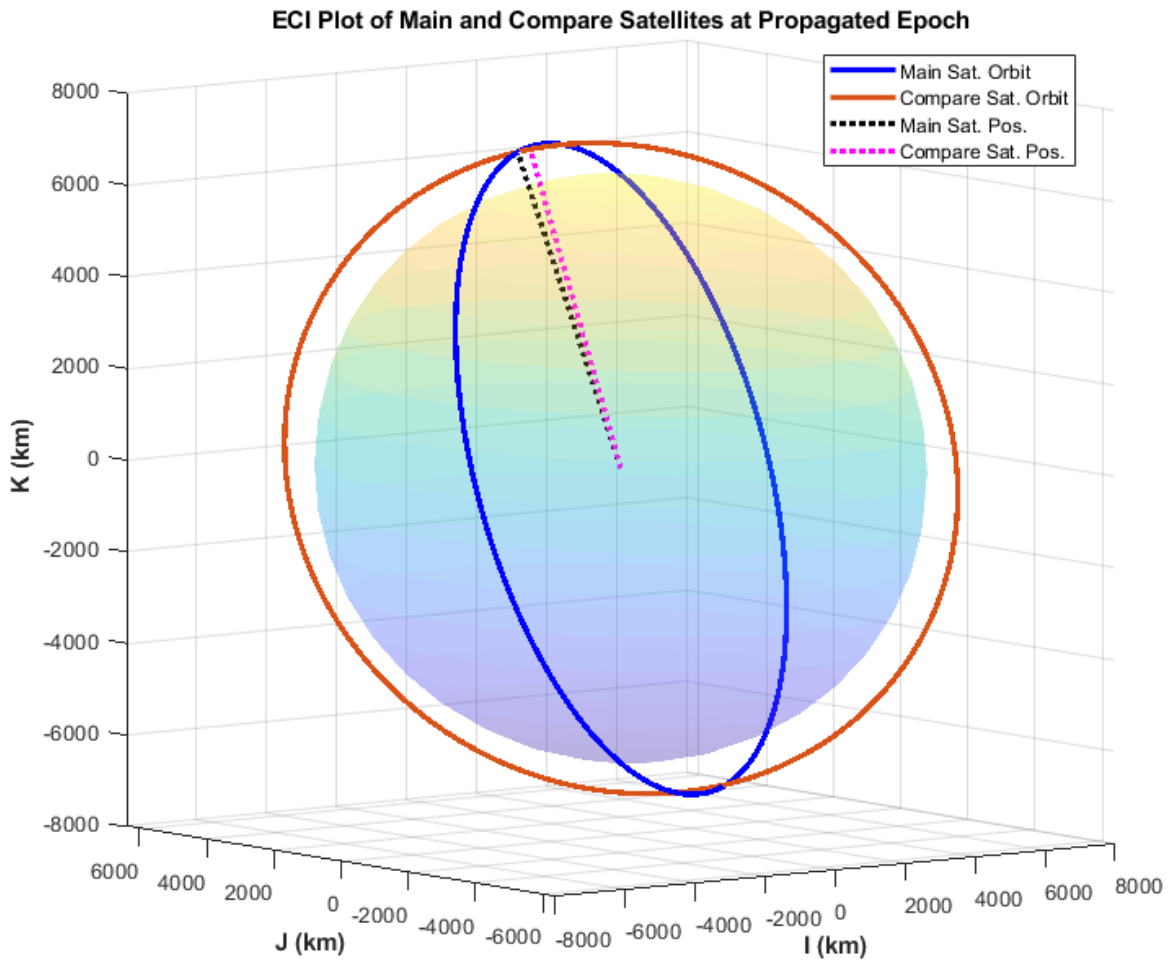
* Data above was supplied by Ryan Walsh's group using a spherical Earth model as a basis for calculation

Using the above position vector of the 'compare' satellite and the spherical Earth model position vector of the main satellite, the approach distance can be calculated.

Approach Distance: 268.7133704 km

Figure 3 displays the main satellite's position and orbit compared to the 'compare' satellite's position and orbit at the propagated epoch for a spherical Earth model.

Figure 3



Conclusion

As shown in the preceding analysis, the two satellites being analyzed do not collide. The calculations to determine this have increased accuracy compared to a spherical earth model as they account for the effect of Earth's oblateness. The orbit propagation is precise to ten digits beyond the decimal point as further precision becomes far too computationally intensive for some computers. Resolving the orbit to this precision also results in a resolution of less than a micrometer, far more precision than needed in most cases as satellites are magnitudes larger in size compared to the resolution. Possible sources of error include perturbations neglected during the analysis such as from celestial bodies such as the moon and sun, atmospheric drag, and from smaller bodies in space like other satellites and pieces of debris. The larger the body and the closer the body, the more of an effect that it will have on the satellites being analyzed, so as these satellites are geocentric, the sun and moon may have a significant impact and their influence will be much larger than that of other satellites and debris. Atmospheric drag, particularly in cases with lower altitude orbits, causes satellite orbits to slow and in some cases lead to deorbit. Having neglected these sources of perturbations will lead to some error. Another possible source of error calculation of the approach. In the most ideal cases, the approach distance used is the distance of closest approach. However, this requires the distance of approach to be calculated at every epoch with sufficiently small time steps to ensure the distance used is the shortest possible. Using an approach distance that is not necessarily the shortest will lead to some error depending on how much the approach distance used and distance of shortest approach differ. Future updates for this software may include accounting for more sources of perturbations and epoch calculations for the distance of shortest approach.

Author Contributions

All code and data/figure inputs under the Test Results section of this report were done by Miles Puchner, with peer review done by Amanda Kerlee and Blake Iwaisako. The Abstract and Introduction were written by Amanda Kerlee. The Orbital Determination section of this report was completed by Blake Iwaisako. Lastly, Amanda Kerlee and Blake Iwaisako both contributed to the completion of the Conclusion Section of this report. All members proofread and edited the report.