A STUDY INTO SATELLITE COVERAGE AND ORBITAL EQUATIONS

Daniel Bielich
daniel.bielich@ucdenver.edu

Gregory Matesi
gregory.matesi@ucdenver.edu

Tuan-Anh (Ricky) Le
tuan-anh.le@ucdenver.edu

Sponsor: Ball Aerospace

Contact: David Osterman

ABSTRACT

Ball is developing a number of small satellite cameras for their evapotranspiration mission that will be placed on satellites within a fixed near polar orbit. The purpose is to provide full camera coverage of the Earth within a 24-hour time period. It is the intention of this report to explore the orbits and provide tools to answer the question of how many satellites are necessary to achieve the full coverage of the Earth in a given time-frame. We will do this by creating an Earth centered Earth fixed model of a satellite's orbit imposing a swath projection onto the orbit to evaluate coverage. We are using the assumption that when the equator has been fully covered we have achieved full Earth coverage.

1 INTRODUCTION

The problem being posed by Ball Aerospace, part of an initiative for the company is to send low-cost satellites outfitted with cameras into space for the purposes of recording photographic data of the Earth's surface. The satellite, called a CubeSat spacecraft, is much smaller than the traditional satellite. They are usually only 10 cm x 10 cm x 11.35 cm. Because of their size they are "cheap" to place in orbit and multiple can be launched at once, as opposed to the traditional one satellite per launch.

For Ball's project specifically, these CubeSat satellites will be carrying CIRiS (Compact Infrared Radiometer in Space) cameras. The cameras are built with multiple calibration settings that can be changed mid-orbit. They will be used to capture images of the Earth's surface in a varying image spectrum that can be changed depending on the data needed.

One of the possible applications of this technology is evapotranspiration, that is how much water is lost into the atmosphere as a result from evaporation of water from the grounds and the release of water from plant material. Scientific organizations use the data to gauge weather patterns and changes in the environment. Another, more practical, use of the data is in the agricultural industry. Knowing how much water is lost from the ground will allow farmers to more accurately predict how much watering needs to be done. This will help increase crop irrigation efficiency and management of water supplies.

In order for the data to be useful, we will need full Earth coverage once every 24 hours. We have been told that we are able to have a maximum of 12 satellites in orbit. These satellites will sit on the same orbital path in a near polar orbit. To accomplish this, we construct a discretized Earth centered Earth fixed model of the orbit numerically. We then impose the camera swath and calculate coverage based on height of the satellite and the angle that a camera points toward the Earth's surface. According to Ball, each camera has a 30 degree angle for coverage. To simplify the problem we will be tracking coverage of the equator and assume that the rest of the globe has been captured once the equator is fully recorded.

2 METHODS

For this project, we will be using an Earth-fixed model, that is a model in which the Earth remains stationary. The orbit itself will be a Sun-synchronous orbit, which is a type of near-polar orbit. We define the calculations using various equations derived from Kepler's Law of Planetary Motion. With the model for a satellites orbit, we convert each orbit to an equi-rectangular model. This model is used then for evaluation of the ground-tracking of each satellite where we impose the swath on the graph and track coverage across the equator.

2.1 SATELLITE COVERAGE

We have k satellites, $k \in [1, \ldots, 12] =: A$. In addition we have to take into account the swath angle for inclination or altitude (400-600 km). We can eventually build on only capturing land masses and avoid any ocean coverage (if desired). With these factors how can we determine an (optimal - improved) plan for measuring evapotranspiration on the Earth's surface.

- How many times can we image the Earth in 24 hours with a varying number of satellites
- How quickly can we image the Earth once (Sunny side)
- The minimum k required to accomplish this
- Given the camera coverage, what varying paths can accomplish the above tasks and what can we infer about their differences

2.2 VISUAL AND NUMERICAL REPRESENTATION

The main contribution we provide is to model the satellite behavior numerically. Given $k \in A$ we want to visually represent the trajectory of each k satellites and capture the coverage of each camera on the Earth's surface. Because of the difficulty of this goal we start with a 2-dimensional model and then implement it within a 3-dimensional framework. We do not to stop there. We have written a function/program that will take flight paths, camera specs (area of coverage) and other factors to provide answers to what data this configuration can collect.

The coding language we use is Python.

- Use Python to simulate and model 2D orbital equations
- Use Python to simulate and model 3D orbital equations

2.3 Defining a Polar Orbit

In polar coordinates, we have the orbit equation defined in Wikipedia (2019e) as

$$r_1(\theta) = \frac{\ell^2}{m^2\mu} \cdot \frac{1}{1 + e\cos(\theta)} \tag{1}$$

$$e = \sqrt{1 + \frac{2E\ell^2}{m^3\mu^2}} \tag{2}$$

where r_1 can be seen as the distance of a satellite from Earth at any given moment and E is the energy of the orbit. See the table below for the explanation the other parameters.

Parameter	Description
ℓ	Angular momentum of satellite and Earth
m	mass of satellite
μ	See * below
θ	Angle of r with the axis of periapsis

** According to Wiki, "μ is the constant for which μ/r^2 equals the acceleration of the smaller body". (Usually $\mu = GM$)*

A more explicit way of defining the orbital equations and the distance from a satellite from Earth is to use Keplerian motion. This result is described in Wikipedia (2019d), but in a different way than described above. Equation 3 present r_2 under these circumstances.

$$r_2(\theta) = \frac{a\left(1 - e^2\right)}{1 + e\cos(\theta)} \tag{3}$$

The notation used to describe these parameters are as follows, e is called the eccentricity of the orbit and a is the semi-major axis (this defines the orbit size). Note the similarity between the equations. In fact they are identical if $\ell^2/m^2\mu = a(1 - e^2)$.

In addition to looking at the Wiki pages for Orbital Equations and Kepler Orbits, I found their Ellipse page to provide some additional useful information. The equation for an ellipse with origin $(0, 0)$,

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1. \tag{4}$$

2.4 Keplerian Elements

Any Keplerian trajectory can be characterized by the **semi-major axis**, **eccentricity**, **inclination**, **longitude of ascending node**, **argument of periapsis**, and **true anomaly**. To be clear I will describe each of these 6 elements below.

The semi-major axis is exactly half of the major axis. It is the longest line that can be drawn from the origin to the edge within any ellipse, refer to 1. The red line would be considered the semi-major axis. The equation of an ellipse, not in polar coordinates (Wikipedia (2019g)) is

$$1 = \frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2}. \tag{5}$$

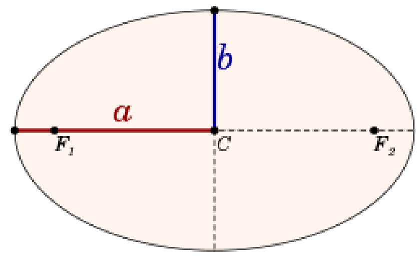

Figure 1: Semi-Major Axis

Some useful information I have found in this context. The parameters h, k are the coordinates for the center of the ellipse. h is the x-coordinate and k is the y-coordinate. “The semi-major axis is the mean value of the maximum and minimum distances of the ellipse from a focus” Wikipedia (2019g) states (but right after they make this claim they say “citation needed”). They then provide some useful relations to the Cartesian coordinate equation 5 to simplify things. Though they are all defined in relation to another. So we need to define at least one value to determine the others. The way we do this is by defining the eccentricity within our code. It is considered user-input and thus can be conformed to whatever ellipse the satellites orbit will have.

$$a = \frac{r_{\text{min}} + r_{\text{max}}}{2}, \quad b = \sqrt{r_{\text{min}} r_{\text{max}}}$$

$$r_{\text{min}} = a\left(1 - e\right), \quad r_{\text{max}} = a\left(1 + e\right)$$

$$e = \sqrt{1 - \frac{b^2}{a^2}}$$

To plot an ellipse using the equation above we only need to know the linear space in which we plot x, then infer what y is based on the representativity. So,

$$y = \sqrt{\left(1 - \frac{(x-h)^2}{a^2}\right) b^2 + k} \tag{6}$$

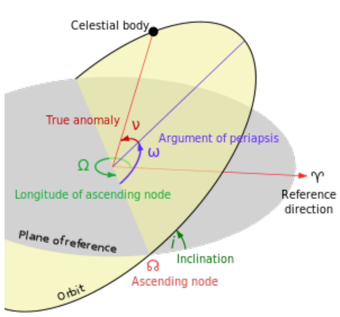

Figure 2: keplerian Orbital Elements

The most clear definition I have found for the eccentricity of an orbit is “the amount by which its orbit around another body deviates from a perfect circle” Wikipedia (2019b). A circular orbit has an eccentricity value $e = 0$. The orbital inclination is the angle at which a satellite is tilted from the Earths equator. If you refer to 2, the plane of reference for the Earth is the equator. From the introduction, we specify that the satellites are all to be near polar orbits.

The argument of periapsis - “is the angle from the body’s ascending node to its periapsis, measured in the direction of motion” Wikipedia (2019f). Something that is not completely clear to me currently, if the argument of periapsis is of $0°$ then ”the orbiting body will be at its closest approach to the central body at the same moment that it crosses the plane of reference from North to South”. Figure 2 provides an image explaining the concept. The explicit formula to calculate the argument of periapsis ω

$$\omega = \arccos \frac{\boldsymbol{n} \cdot \boldsymbol{e}}{|\boldsymbol{n}||\boldsymbol{e}|}.$$

Again, e is the eccentricity and $\boldsymbol{n}$ is a vector pointing in the direction of the ascending node and Wiki notes that the z component is always zero for this $\boldsymbol{n}$ vector.

The true anomaly ν is "the angle between the direction of periapsis and the current position of the body" Wikipedia (2019a). Below I include the representativity of the true anomaly for any elliptic orbit. Beforehand, according to Wikipedia (2019c) "An inclination of exactly $90°$ is a polar orbit, in which the spacecraft passes over the north and south poles of the planet". Thus justifying taking this parameter so that i is $90°$, after communicating with David from Ball we know the orbit of the satellites have an inclination roughly $83°$. To compute i explicitly we have,

$$i = \arccos \frac{h_z}{|h|}$$

$$\nu = \arccos \frac{\boldsymbol{e} \cdot \boldsymbol{r}}{|\boldsymbol{e}||\boldsymbol{r}|}.$$

where h is the orbital momentum vector, h_z is the z component, e is eccentricity and r is the orbital position vector.

The true anomaly seems difficult to capture since it is dependent on the position a satellite will be at any given moment. In practice we find the general consensus is to substitute the true anomaly with the mean anomaly. Which is what we do.

2.5 Swath

The swath representative of the camera coverage will be evaluated as a triangle that is perpendicular to the path of an orbit. In order to evaluate the swath we need to construct a tangent at time t to our orbital path. Using the tangent and the radius r from a satellites orbit (radius to the center of the Earth) we can construct the legs of the triangle. This along with our height $h(t)$ at time t we are able to construct the base, which will then be used in within our ground-tracking model.

The swath approaches the equator at an angle so we incorporate this thought when calculating the swath projection. To do this, we first take two points straddling the equator and interpolate a line crossing the equator. The swath is imposed onto the equator its center at a point on the interpolated line crosses the equator. The line that intersects the equator at angle $theta$ allows us to impose the additional measurement across the equator. From this, we can construct a triangle on both sides that give us the two endpoints of the swath coverage. Figure 6 gives an example of how the swath across the equator is calculated.

2.6 Python Modules

To run the Jupyter notebooks that are included with this write-up, there are two packages (not usually contained with a python/anaconda install) that will need to be installed in order to use.

The basic modules needed that should be fine on all hardware with a Python install are: Numpy, Scipy, Matplotlib and MPL Toolkits. The two additional files that one will need are Cartopy and Interval. Cartopy is only used for the ground tracking plot. It provides the Earth backdrop within the plotting. Interval is a package used when tracking coverage across the equator. We use Interval to combine the union of overlapping coverage.

3 RESULTS

Below are the three images in the results. **Figure 3** is the one-dimensional displacement for the three Earth centered Earth fixed models for one satellite for four periods. Note the scale for each axes is represented in meters which is why they are so large. **Figure 4** is the interpolation of points straddling the equator. Here we are modelling three satellites in a half day orbit. **Figure 5** and 6 are a zoomed in image of **Figure 4**. **Figure 5** shows how we numerically interpolate and anchor a point at the equator. Figure 6 represents the swath projection as we cross the equator. The black line represents the length of the swath for that given altitude and we impose the angle of approach to guaranty we don't underestimate coverage. The final zoomed in image, **Figure 7** shows the exact same calculation but we include the calculated swath length using the anchored point at the equator.

Figures 8 and **9** are both displaying our results in an equi-rectangular model. Both figures show the ground tracking of simulated orbits, one with 5 satellites for 1/4 of a day while the other represents 3 satellites in orbit for half of a day. **Figure 10** shows the coverage at the equator after evaluating the intervals of coverage. This image shows almost full coverage because we simulate 12 satellites for one full day at 500 km in altitude. Note, with a better input for the parameter *argument of perigee*, the spacing on the satellites *could* be better and we should be able to achieve full coverage. Finally, **Figure 11** cuts out all of the unnecessary portion of the globe that is unused. That way if and when someone uses this image in a presentation, they don't need to say please ignore everywhere except the equator. Note here we barely have equator coverage. This simulation was captured using 3 satellites and half of a day.

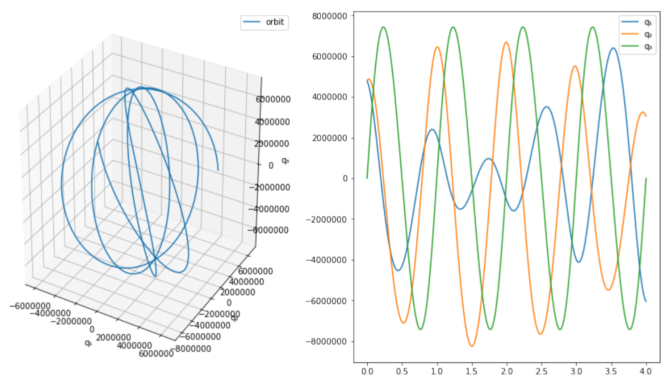

Figure 3: 3-D and 1-D Displacement of ECEF Vector

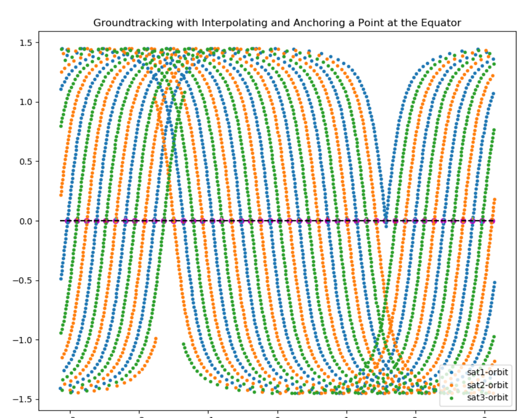

Figure 4: Ground Tracking Plot of Interpolation at Equator

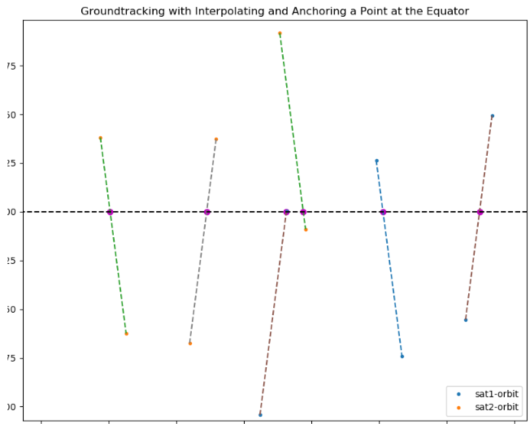

Figure 5: Zoomed in Image of Interpolating Equator Crossing

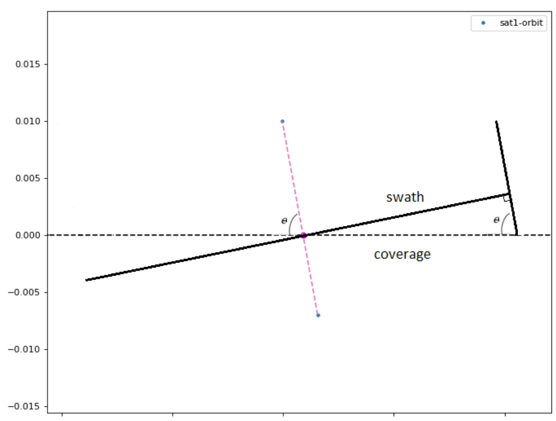

Figure 6: Swath Calculation

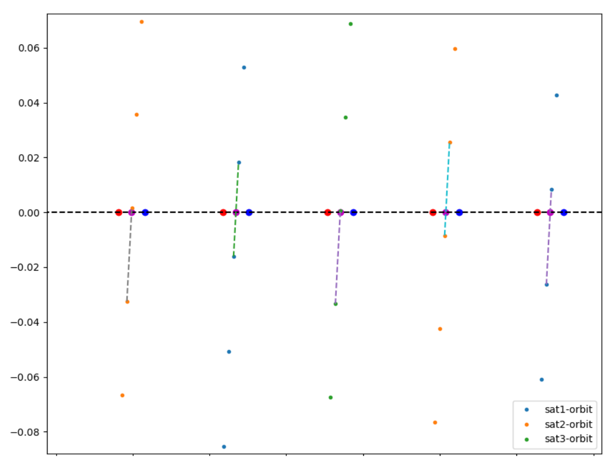

Figure 7: Interpolation of Equator Crossing with Swath Projection included

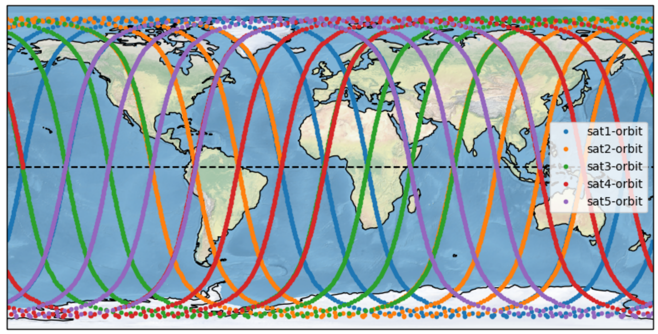

Figure 8: 2-D Ground Tracking

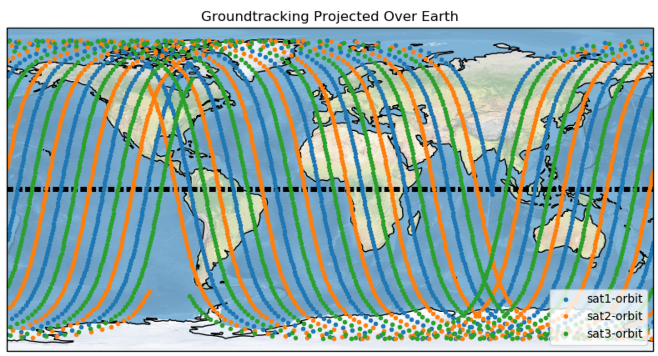

Figure 9: 2-D Ground Tracking

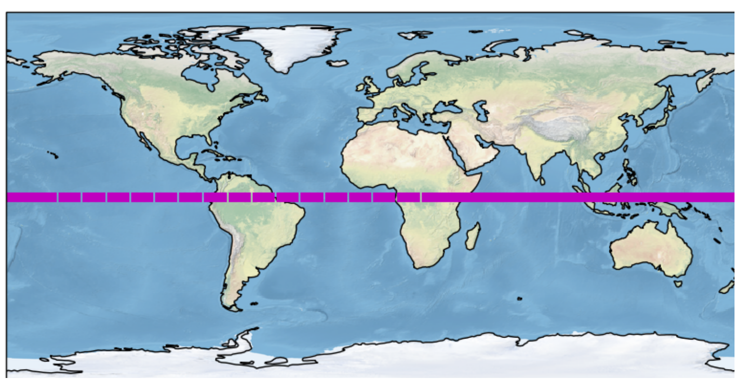

Figure 10: Equatorial Coverage After Simulation

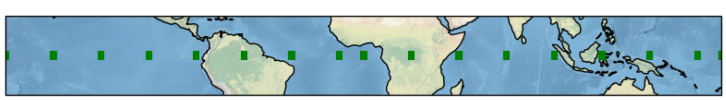

Figure 11: Equatorial Coverage After Simulation

4 DISCUSSION

Both graphs in **Figure 3** display the Earth centered Earth fixed (ECEF) displacement of a single satellite over four orbits. The graph on the left is the 3 dimensional ECEF displacement over time.

The graph on the right displays all three ECEF displacement components individually. The third, polar, component of the ECEF position retains a constant magnitude. This is due to the fact that the inclination of the orbital plane remains constant. So each orbit achieves a particular maximum and minimum latitude. The first and second, equitorial, components of the ECEF displacement have variable magnitude. This is due to the argument of perigee remaining constant in the orbital plane and the longitude of the ascending node precessing once per day. When the projection of the semi major axis into the equitorial plane is aligned with the x-axis the first component of the ECEF vector will achieve its maximum magnitude. Conversely, the second component of the ECEF position will then have it's smallest magnitude.

The ECEF positions over time were projected onto a rectangular plane representing the surface of the earth as displayed in **Figure 4**. The y-axis runs from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$ with 0 on the y-axis representing the equator. The x-axis runs from $-\pi$ to π. The graph in **Figure 4** shows the equi-rectangular projection of three satellites completing seven orbits.

The equi-rectangular projection of a satellite's path allows for a visual and a numerical way to identify when a satellite crosses the equator. We can then interpolate a line between the two points immediately preceding and following the crossing as displayed in **Figure 5**. This interpolated line serves as an approximation to the projected satellite path. Here, for each satellite, two passes are in the same north/south direction and their projected paths are parallel to each other. The third crossing is in the opposite north/south direction and its interpolated path crosses the equator at a complimentary angle.

Each projected interpolated crossing has a swath associated with it. This swath covers a small portion of the equator. **Figure 6** provides a visual interpretation of the trigonometry used to calculate this small portion for any crossing. The two knowns here are the length of the swath and the angle from the projected interpolated path of a satellites crossing past the equator. The unknown coverage interval can then be calculated using the sine function.

Now that we can calculate a coverage interval for any projected equator crossing we are able to plot the interval of the equator covered by the swath for each crossing as shown in **Figure 7**.

Figures 8 and **9** are the projected equi-rectangular paths of five satellites over 3 orbits. The equi-rectangular map of the earth is now included. Additionally, **Figure 9** features the interval of the equator covered by the swath for each pass.

The projected paths of the satellites may visually detract from the goal of plotting and accumulating equatorial coverage. **Figure 10** features the interval of the equator covered by the satellites while excluding the projected equi-rectangular paths of the satellites as seen in the preceding figure. This graphic provides a less cluttered visual tool to assess how much of the equator has been covered.

Similarly, if our goal is to cover the equator than the entire rest of the equi-rectangular map of the Earth is an additional source of visual noise. **Figure 11** provides an even less cluttered visual tool to assess equatorial coverage by zooming in on just the equator. The graph in this figure shows the equatorial coverage accumulation for one satellites after just over one full day.

5 CONCLUSION

Our final product is a tool that can be used to model how long it would take to scan the entire Earth for a given number of satellites. It should be noted however, the number of days required in our model may be an overestimate. The reason for the discrepancy is the spacing of the projected satellites are not optimal. Our lack of knowledge within this field of study is shown with this part of the research. We believe the argument of perigee to be the deterministic value which displaces a satellite in an orbit. But we were unable to quantify which set of ω-values would sufficiently distribute the satellites (ω is how it is referenced within the Python code and within literature).

Ultimately, we were successful in creating several models that showed full coverage at the equator. We need to note that in this model, the method of how we determine the total length of the equator is determined by R which is a parameter defined in user input. The default parameter set is defined to be the mean distance from the Earth. The correct calculated distance is provided in a commented block where the parameter is defined. The result provides 40,075 km which is the length of the equator whereas using the Earth's mean radius means we calculate the equator to be 40,030 km. In addition we did not factor in the curvature of the Earth when calculating the swath, our argument was that coverage would be negligible. Overall this tool will provide a good answer to the question Ball Aerospace asks, which is how many satellites are needed to achieve full coverage in one day and at what altitude.

REFERENCES

Wikipedia. True anomaly, 2019a. URL https://en.wikipedia.org/wiki/True_anomaly.

Wikipedia. Eccentricity, 2019b. URL https://en.wikipedia.org/wiki/Orbital_eccentricity.

Wikipedia. Orbital inclination, 2019c. URL https://en.wikipedia.org/wiki/Orbital_inclination.

Wikipedia. Kepler orbit, 2019d. URL https://en.wikipedia.org/wiki/Kepler_orbit.

Wikipedia. Orbit Equation, 2019e. URL https://en.wikipedia.org/wiki/Orbit_equation.

Wikipedia. Argument of periapsis, 2019f. URL https://en.wikipedia.org/wiki/Argument_of_periapsis.

Wikipedia. Major semi-axis, 2019g. URL https://en.wikipedia.org/wiki/Semi-major_and_semi-minor_axes.