A report on

Simulation of a Satellite performing orbital transfer

Presented by:

Anirudh Addagada	N19904251
Amogh Anantha Murthy	N12619144
Mustafa Bhadsorawala	N12220235

NEW YORK UNIVERSITY
TANDON SCHOOL OF ENGINEERING
Brooklyn, NY-11201

Year 2021-22

Content

Table of Figures

1. Abstract

Satellites are an extremely expensive undertaking, financially as well in terms of the time it takes to build it. This automatically makes it a high-stake endeavor and the chances for failures must be minimized to as little as possible. Therefore, we cannot just launch a satellite and hope for the best we need to develop tools to analyze the behavior of the satellite at different states and different initial conditions. The focus of this study is to build a tool that can help us analyze the path a satellite may travel. We incorporate the gravity of the Sun and Earth to model the system and use a State Space system to set our initial conditions. Lastly, we also perform orbital transfer using the Hohmann transfer.

2. Methodology

The path of a celestial body can be traced by means of geometric equations; however, this will not account of the physical properties of the bodies in the planetary system like the mass of the Sun and planet of interest, their radii, distance between the Sun and the planet etc. To accurately depict the system, it needs to be solved numerically with the physical properties.

We break the problem in 3 different parts:

1. Orbit Calculation – We conduct the numerical analysis and develop the trajectories of the motion of the Earth around the Sun and the satellite around Earth
2. Simulation – We create 3D plot the different heavenly bodies to display in an animation
3. Orbital Transfer – Lastly, we use the Hohmann transfer to perform orbital transfer by the satellite

2.1 Orbit Calculation

1. To simulate the Earth around the Sun, the known data i.e., the gravitational constant, radius of the Earth, radius of the Sun, mass of the Earth, mass of the Sun and the distance between them.
2. We create the objects for the Sun, Earth, and the satellite with the above-mentioned parameters in each.
3. Using this known data along with the position and velocity vectors, the angular momentum of the Earth around the Sun can be found by the equation:

$$\vec{h} = \vec{r} \times \vec{v}$$

4. Using the angular momentum of Earth, the eccentricity can be found to determine the type of orbit by the following equation where μ is the gravitational parameter:

$$\vec{e} = \frac{\vec{r} \times \vec{h}}{\mu} - \frac{\vec{r}}{r}$$

5. For each conic section i.e., circular, elliptical, parabolic, and hyperbolic orbits the corresponding value of eccentricity is 0, $0 < e < 1$, $e = 1$ and $e > 1$ respectively. Similarly, the semi major axis of any conic section can be expressed as:

$$a = \frac{h^2}{\mu(1 - e^2)}$$

6. The two-body system of Sun and Earth can be represented using a linear state space representation and integrated to find the position vector of the planet at any given instant in its orbit around the Sun. The state space representation called the state vector is as follows:

$$sys = \begin{bmatrix} x \\ y \\ z \\ X' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} c1 \\ c2 \\ c3 \\ c4 \\ c5 \\ c6 \end{bmatrix}$$

7. For better understanding, the state vector can be expressed as a column vector consisting of [Position_x Position_y Position_z Velocity_x Velcoity_y Velocity_z]. To find the position of the orbiting body, first, the time period of the orbit is needed, which is different based on the type of the orbit. The eccentricity is used to distinguish between the types of orbits, i.e., if an orbit is circular, elliptical, parabolic, or hyperbolic. The time of flight for an orbit with eccentricity less than 1 is given by:

$$T = 2\pi \sqrt{\frac{a^3}{\mu}}$$

8. Having found the time period for one full orbit, the state vector is integrated over the time period. The result of this integration is a matrix composing the position and velocity of the object at each point of time along the orbit. Similarly, the position and velocity vector matrix are calculated for the moon going around the Earth, which in turn is orbiting the Sun. In case of a man-made satellite that is orbiting the Earth, the trajectory of the object depends solely on the mass of the orbited body i.e., Earth since the mass of the satellite is small compared to the mass of the Earth which makes it inconsequential.

We write 2 functions, one to find the time of flight and one for the state matrix over the period of flight, these functions can be used determine the trajectory of any celestial body and man-made satellite given the initial position and velocity vector or the state vector.

The get_TOF() function calculates the time of flight for one complete revolution about the body being revolved around.

We pass the :

- The Angular Momentum
- Gravitational constant of the orbital body
- Eccentricity of the orbit
- Mass of the orbital body
- Distance from the Sun
- Number of Orbits we want to calculate for (default = 1)

```
Get_TOF(a,u,e,Body_Mass,Dist_to_Sun,no_of_orbits)
```

Inside the function we check if the eccentricity and depending on the condition we define the Time-of-flight formula

- If e < 1 && e >= 0
  - $T = 2\pi N\sqrt{\frac{a^3}{\mu}}$
- If e > 1
  - $T = \sqrt{\frac{-a^3\,(\mathrm{e.sin}^{-1}F - F)}{\mu}}$
- If e == 1
  - $T = \sqrt{\left(\frac{2a(1-e^2)^3}{\mu}\frac{D+D^3}{3}\right)}$

```
%% Propogation of Earth orbit

%Simulating using the ode45 solver
options = odeset('RelTol',1e-8,'AbsTol',1e-9);
[t_S,S_S] = ode45(@(t_S,S_S) OrbitState(t_S,S_S,u_earth,earth_Radius),tspan_S,sys_S,options);
state_S = S_S;
```

Fig 1.1 Integrating the system states at each time step using ode45 solver

The OrbitState() function finds the position at each time step using ode45 to solve to integrate over each time step, the first three columns of the resulting matrix are the positions along the x, y and z directions which are used to create a 3D simulation. A total of 6 column vectors are stored for the simulation, 3 determining the position of Earth around the Sun and 3 for the moon or satellites around the Earth.

```
%% Capturing the Orbit Trajectory

States_X_S = state_S(:,1); % all x
States_Y_S = state_S(:,2); % all y
States_Z_S = state_S(:,3); % all z
States_Xdot_S = state_S(:,4); % all xdot
States_Ydot_S = state_S(:,5); % all ydot
States_Zdot_S = state_S(:,6); % all zdot
```

Fig 1.2 Capturing the orbit trajectory

The MATLAB code follows the following structure:

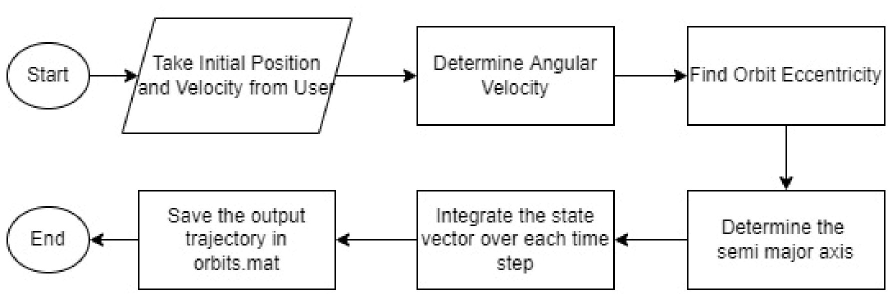


Fig 1.3 Flowchart for Orbit calculation

2.2 3D Simulation

For 3D simulations, a function is written for each object that is the Sun, Earth, satellite, and a circle (the name circle is just a legacy, but it can create shapes that are elliptical as well as hyperbolic, depending on the requisite) depicting the trajectory of the satellite.

We consider the Sun is the origin, the Earths object is placed at the projected position in 3D space and similarly the 3rd object is carefully created at its designated 3D coordinated from the center of Sun. Each of these objects are rotated about their z axis to simulate the rotation.
We showcase some of our results below

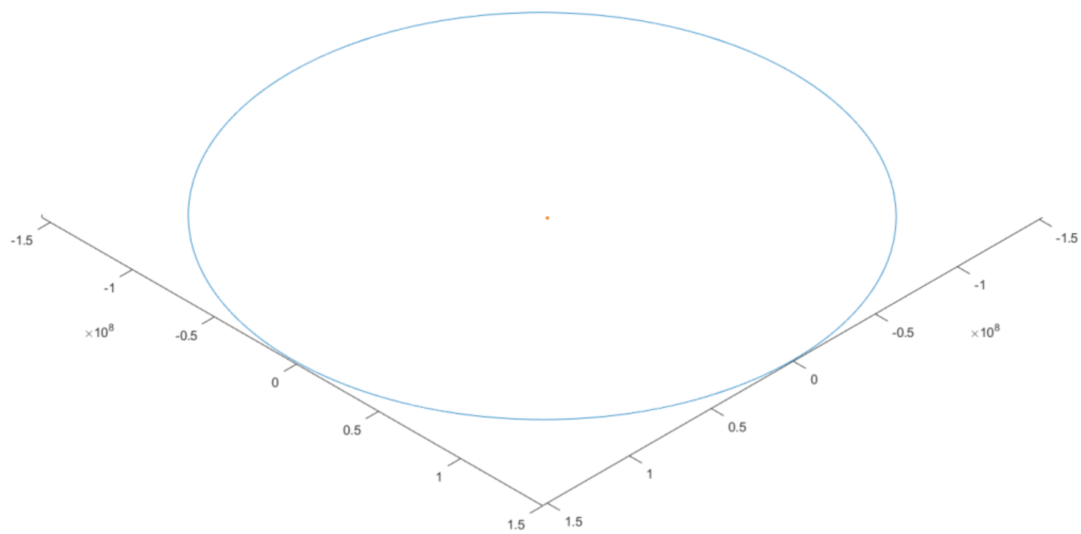


Fig 1.4 To scale Model of the orbit of Earth around the Sun

From fig 1.4, we can see that when the model is to scale, we can barely perceive the individual objects, but if you look closely, you can observe the Sun in the center at 0,0,0. The Earth and the satellite however is invisible for all practical purposes. To get around it we scale the models by a factor so that they are visible in relation to each other.

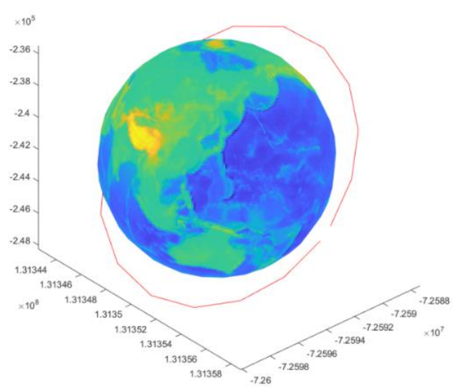


Fig 1.5 To scale model of a Low Earth Orbit around the Earth

Upon launch, a satellite is often placed in one of several types of orbits around the Earth. The type of orbit a satellite would assume depends on what it is built to achieve. Following are the some of the different types of orbits.

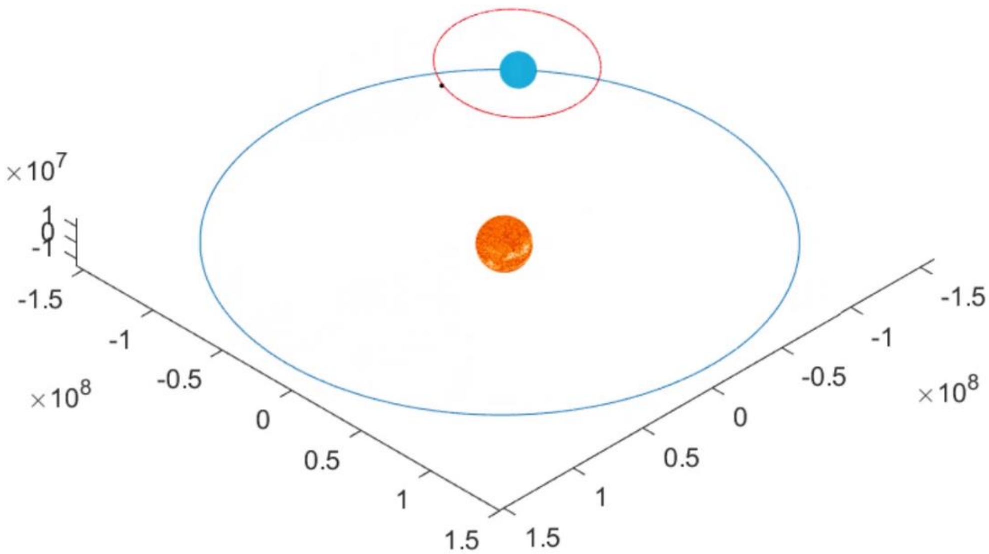


Fig 1.6 The Sun Earth and Moon Model

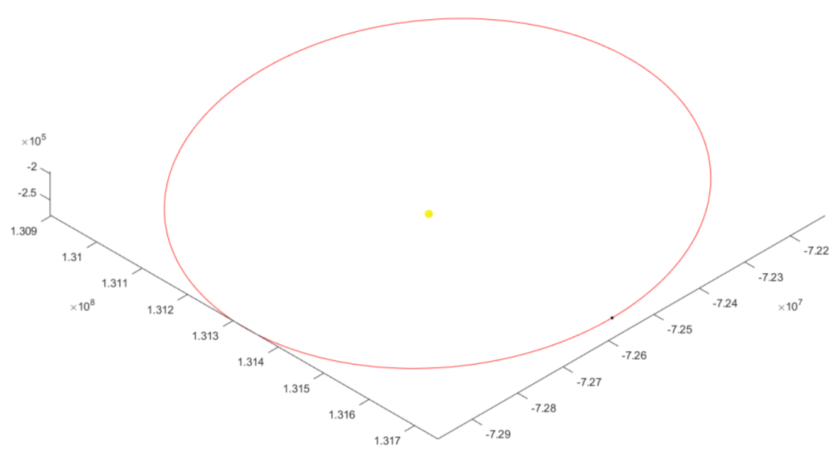


Fig 1.7 To Scale model of the Earth and the Moon

2.2.1 Geostationary Orbit

Satellites in this orbit circle the Earth above the equator following the Earth's rotation by travelling at the same rate as the Earth which is 23 hours 56 minutes and 4 seconds. To achieve this orbit, the satellite must have a speed of 3 km/s at an altitude of 35,786 km from the surface of Earth. These orbits cater to telecommunications such that an antenna on the ground can always point at the same direction.

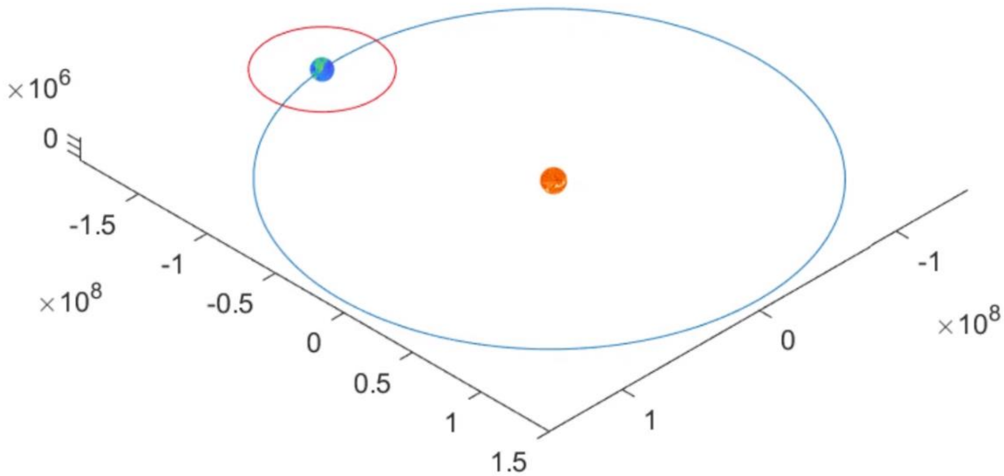


Fig 1.8 Geostationary Orbit

2.2.2 Low Earth orbit

As the name suggests, this type of orbit is closer to the surface of the Earth. Usually at an altitude less than 1000 km and above 160 km. As opposed satellites in geostationary orbits, ones in LEO

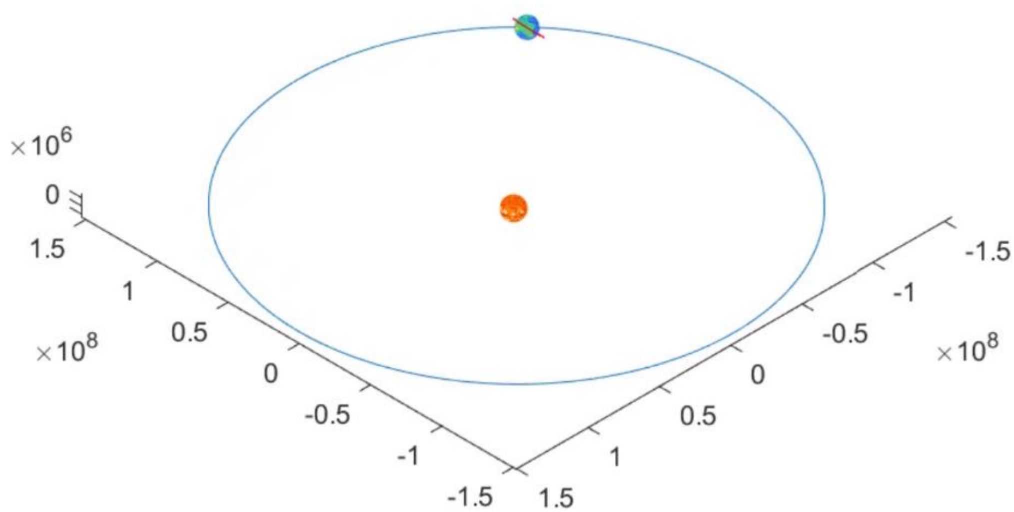


Fig 1.9 Low Earth Orbit

do not have to be along the equator and instead have an inclination, this means they can cover different locations on the planet, which is one of the main reasons why LEO is the most used orbit.

The applications include satellite imaging but not limited to it. This orbit is used by the ISS as it makes it easier for astronauts to travel to and from ground. At this orbit ISS moves approximately at a speed of 7.8 km/s and take around 90 minutes to complete one orbit and orbits Earth about 16 time in a 24-hour period.

2.2.3 Polar Orbit

Satellites in this orbit revolve around Earth roughly passing over both north and south poles. A polar orbit essentially a Low Earth orbit with altitudes between 200 to 1000 km. These satellites are mainly used for scientific research and satellite imaging to monitor climate changes.

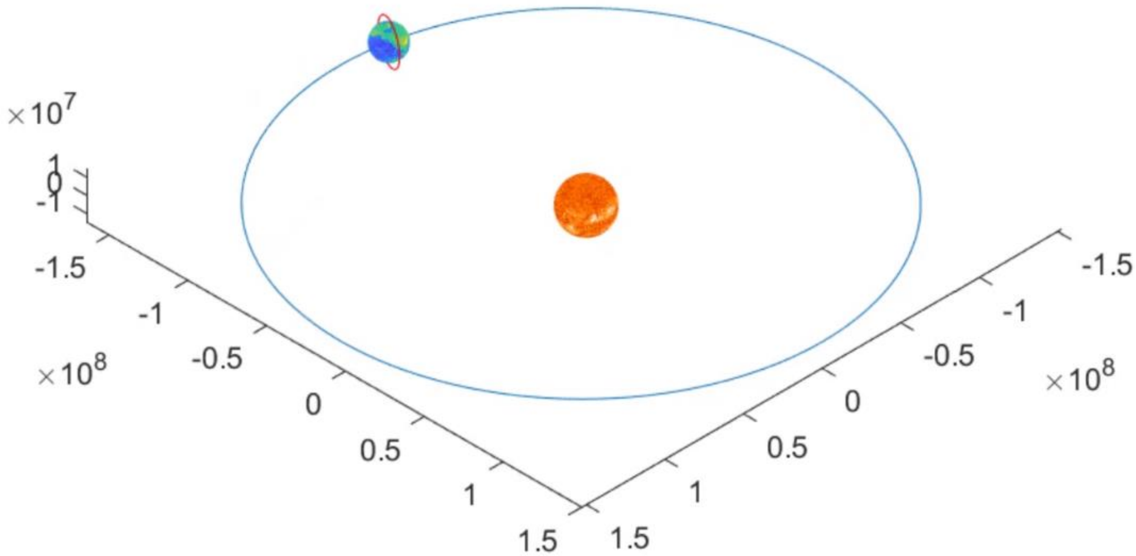


Fig 1.10 Polar Orbit

2.2.4 Molniya Orbit:

A Molniya orbit is a highly elliptical orbit with an inclination of 63.4 degrees with an orbit time period of 12 hours approximately. This orbit is named after a series of soviet Russian military satellites in a high dwell elliptical orbit in the mid-60s. This type of orbit has a long dwell period over the hemisphere of requirement that is northern or southern. The satellite moves quickly when over the hemisphere that is not of interest. When at a higher point in orbit the satellite has a wide angle of view making it compatible with telecommunications. This type of orbit has the advantage of covering inclinations not possible by a geostationary orbit, however, multiple satellites are required for uninterrupted communication.

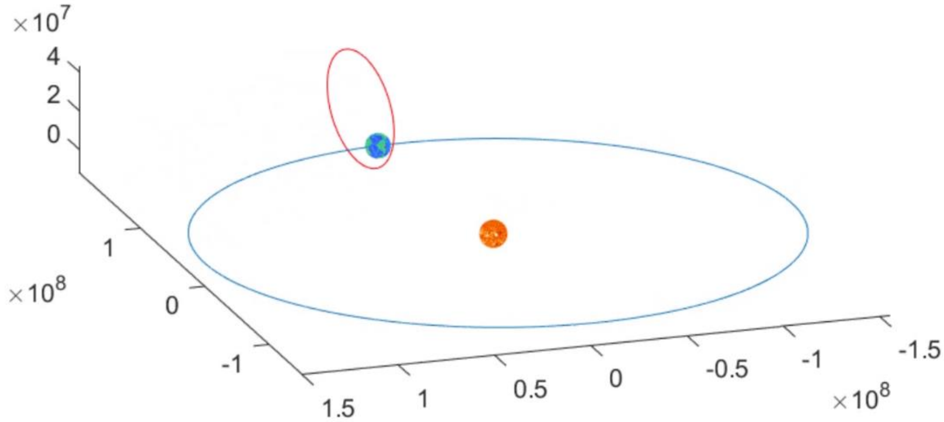


Fig 1.11 Molniya Orbit

2.3 Orbital Transfer

Orbital maneuvering is the act of changing the orbital parameters of an object. Change in orbital elements depends on magnitude or direction change of orbital velocity. Any maneuver to change an orbit occurs at the intersection of the initial and final orbit. In this project Hohmann transfer is simulated.

Hohmann transfer

The most energy efficient (time free) coplanar circular to circular orbit is the Hohmann transfer. This transfer consists of two velocity impulses. The first impulse acts on the initial circular orbit tangentially, which drives the spacecraft into an elliptical transfer orbit. A second tangential velocity impulse at a transfer angle of 180 degrees from the first impulse sets the spacecraft into its desired final orbit and altitude.

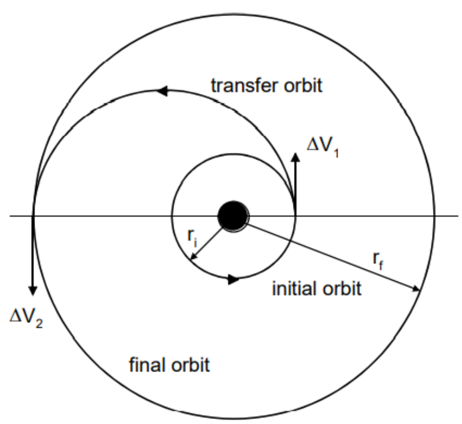


Fig 1.12 – Coplanar Hohmann Transfer

Coplanar Transfers –
As the name suggests, in this type of transfer both the initial and final orbits are coplanar. Both velocity impulses are restricted to the orbital planes of the initial and final orbits. The perigee

altitude of the elliptical transfer orbit is equal to the altitude of the initial circular orbit and the apogee altitude is equal to the altitude of the final orbit.

We begin by defining the initial orbit radius (ri) and final orbit radius (rf).

$$ri = re + hi$$

$$rf = re + hf$$

Where,

re is the radius of the Earth.
hi is the altitude of the initial orbit.
hf is the altitude of the final orbit.
We can define the semi major axis, a as

$$a = (ri + rf)/2$$

The time of transfer is half the total time period as the elliptical transfer orbit terminates at 180 degrees. It is defined as

$$\tau = \pi \sqrt[3]{\frac{a^3}{\mu}}$$

Where,

$$\mu = G \times M$$

G is the gravitational constant of Earth.
M is the mass of Earth.
The orbital eccentricity of the transfer orbit can be defined as

$$e = \frac{\max(r_i, r_f) - \min(r_i, r_f)}{r_f - r_i}$$

Now, to find the two velocity impulses we define local circular velocity (Vlc) and three normalized radii (R1, R2, R3)

$$Vlc = \sqrt{\frac{\mu}{r_i}}$$

$$R1 = \sqrt{2\frac{r_f}{r_f + r_i}}$$

$$R2 = \sqrt{\frac{r_i}{r_f}}$$

$$R3 = \sqrt{2\frac{r_i}{r_f + r_i}}$$

Using the above equations, the first velocity impulse is calculated by

$$\Delta V1 = Vlc\sqrt{1 + R_1^2 - 2R_1}$$

which is the difference between the initial orbit speed and the perigee speed of the transfer orbit. The second velocity impulse is calculated by,

$$\Delta V2 = \text{Vlc}\ \sqrt{R_2^2 + R_2^2 R_3^2 - 2R_2^2 R_3}$$

which is the difference between the final orbit speed and the apogee speed of the transfer orbit. Apart from the equations stated above, there are two main functions which are used while simulating orbital maneuvers

1. ord2eci – The coordinate system relative to which velocity and position are measured is called the Earth-Centered Inertial (ECI) coordinate system. It is a fixed with respect to Earth's surface and rotates with respect to stars. The function converts classical orbital elements to ECI. Its inputs are μ, a, e, orbital inclination, argument of perigee, right ascension of ascending node and true anomaly.

```
% determine correct true anomaly (radians)
if (alt2 > alt1)
        oevti(6) = 0.0;
    else
        oevti(6) = deg2rad(180.0);
    end
[rti, vti] = orb2eci(mu, oevti);
oevtf(1) = smat;
oevtf(2) = e;
oevtf(3) = 0.0;
oevtf(4) = 0.0;
oevtf(5) = 0.0;
```

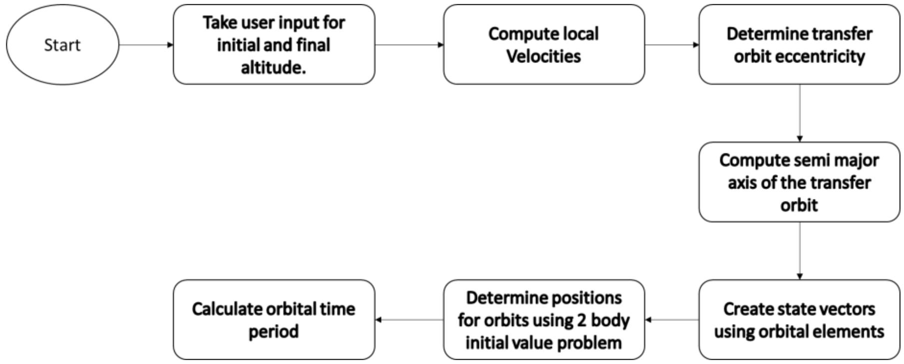
2. twobody2 – It is a function that is used to propagate two body or unperturbed satellite orbits. It gives the final position and velocity vectors. Its inputs are μ, τ, initial position vector and initial velocity vector.

```
[rwrk, vwrk] = twobody2 (mu, t1, ri, vi);
rp1_x(i) = rwrk(1) / req;
rp1_y(i) = rwrk(2) / req;
rp1_z(i) = rwrk(3) / req;
```

The MATLAB code follows the following structure –

- Takes user input of initial and final orbit altitudes.
- Determines the geocentric radii for both orbits.
- Computes the local circular velocity.
- Determines the eccentricity of the transfer orbit.
- Computes the apogee and perigee radii and velocity of the transfer orbit.
- Loads the arrays containing orbital elements and creates state vectors.
- Determines the true anomaly.
- Calculates the orbital periods.
- Finds the position vector for all three orbits using two body initial value problem method.
- Finally, plots the orbits with the Earth at its center.

The MATLAB code follows the following structure:



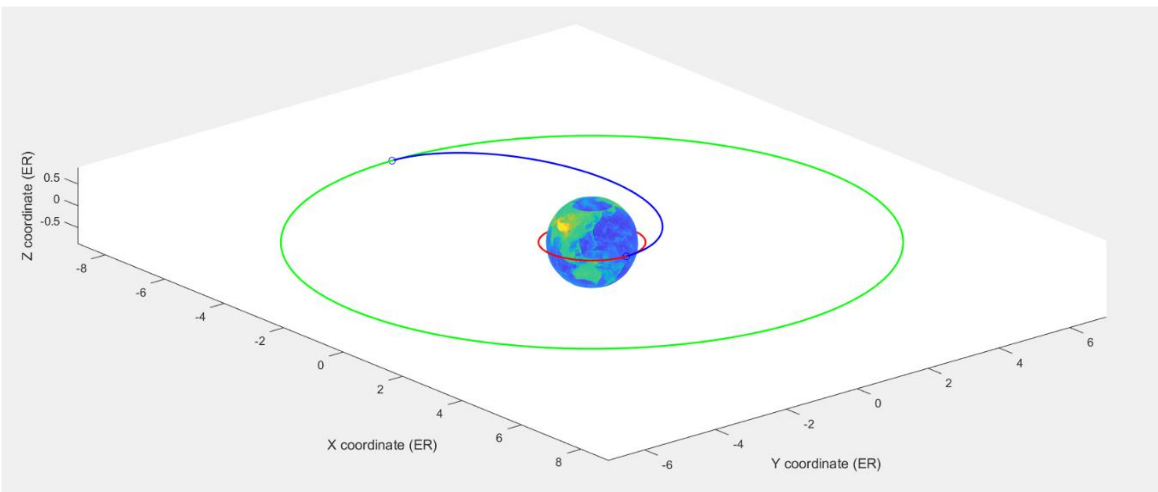


Fig 1.13 : Hohmann Transfer example

The above figure shows Hohmann transfer from ISS altitude to the geostationary altitude, with the elliptical transfer orbit. All the orbits and the Earth are to scale.

References

- https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/39160/versions/4/previews/twobody2.m/index.html
- https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/39248/versions/4/previews/orb2eci.m/index.html
- CelesTrak: "Orbital Coordinate Systems, Part I"
- FULLTEXT02 (diva-portal.org)
- The Hohmann Orbit Transfer - File Exchange - MATLAB Central (mathworks.com)
- https://www.mathworks.com/matlabcentral/fileexchange/57132-satellite-orbit-transfer-simulation
- Programming a simulation of the Earth orbiting the Sun (evgenii.com)
- ISSFD24_Paper_S2-6_Yu.pdf
- https://nssdc.gsfc.nasa.gov/planetary/factsheet/Sunfact.html
- https://webb.nasa.gov/content/about/orbit.html
- https://www.jwst.nasa.gov/content/about/comparisonWebbVsHubble.html
- https://www.wired.com/2011/08/james-webb-space-telescope-and-l2-orbits/
- equations-planetary-motion.pdf (wiu.edu)
- Romalyn.Mirador-F19.pdf (sjsu.edu)
- Viewer (mathworks.com)