Orbital Elements -Numerical Analysis -1 using MATLAB®

SPACECRAFT NAVIGATION AND GUIDANCE

ALLIANCE UNIVERSITY
ALLIANCE COLLEGE OF ENGINEERING AND DESIGN

Submitted to:
Prof. Yadu Krishnan.S
yadukrishnan.s@alliance.edu.in

Submitted by:
L.Ramkiran
17030141AE007
lramkiranBTECH17@ced.alliance.edu.in

1 Classical Orbital Elements

An orbit has six degrees of freedom. So, there are six classical orbital elements. These six can completely define an orbit, with a few exceptions[1]. The six classical orbital elements are:

1. Semi-major Axis
2. Eccentricity
3. Inclination
4. Right Ascension of Ascending Node
5. Argument of Perigee
6. True Anomaly

1.0.1 Code:

These are the Sections 1,2,3& 4 of the code.
In Section-1 input from the user is taken. The inputs are the Position vector and the Velocity Vector.
In Section-2 Error check is done to avoid misbehaving of the code.
In Section-3 Constants(G,m,μ) required for the calculations are defined.
In Section-4 Additional values($\overrightarrow{h}$, $\overrightarrow{n}$, $|\overrightarrow{h}|$, $|\overrightarrow{r}|$, $|\overrightarrow{v}|$, $|\overrightarrow{n}|$) that are required to calculate the results, are calculated.

```
%% %Inputs from the user
r = input("Enter the Position Vector: "); %Radius Vector
v = input("Enter the Velocity Vector: "); %Velocity Vector
%% %Checking if the entered inputs are valid or not
if length(r) ~= 3 || length(v) ~= 3
    error("%%@@**!!PLEASE ENTER A VALID VELOCITY AND POSITION VECTOR!!**@@%%")
else
    ii = 1;
end
%% %Constants
G = 6.674480911*10^(-20); %Universal Gravitational Constant
m = 5.972*10^24; %Mass of Earth
mu = G*m; %Standard Gravitational Parameter
%% %Required values to calculate COE and AOE
h = cross(r,v); %Specific Angular Momentum
I = [1 0 0];
J = [0 1 0];
K = [0 0 1];
n = cross(K,h); %line of node or vector along the nodes
magr = norm(r);
```

[1]In those cases we use Alternate Orbital Elements to define the orbit

```
magv = norm(v);
magh = norm(h);
magn = norm(n);
```

Listing 1: Inputs-ErrorCheck-Constants-RequiredValues

1.1 Semi-major Axis(a)

This is a constant that defines the size of the orbit. In Circular it is the radius of the circle. It is the longest diameter of an ellipse. In figure(**??**), the semi-major is the distance between C and A.

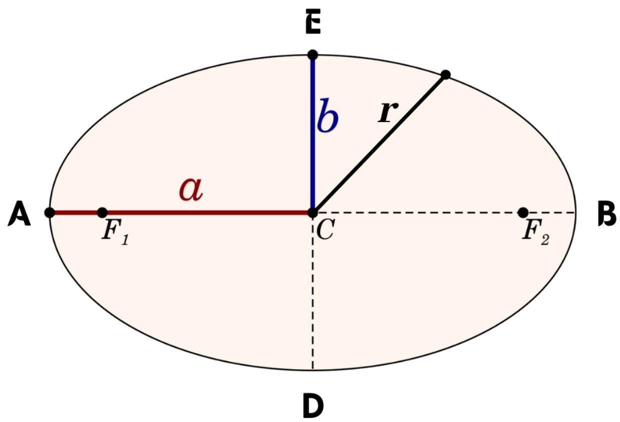

Figure 1: Ellipse with Semi-major axis of a.

To calculate the semi-major axis the following equations can be used.

$$a = \frac{r_a + r_p}{2} \tag{1}$$

$$a = \frac{h^2/\mu}{1 - e^2} \tag{2}$$

$$a = \frac{1}{\left(\frac{2}{r} - \frac{v^2}{\mu}\right)} \tag{3}$$

$where,$

$r_a = Radius\ of\ Apogee(F_1 B)$

$r_p = Radius\ of\ Perigee(F_1 A)$

$\mu = Standard\ Gravitational\ Parameter,$

$\epsilon = Specific\ Mechanical\ Energy,$

$v = Velocity\ at\ r,$

$r = Distance\ from\ one\ of\ the\ focal\ point\ to\ the\ position\ of\ the\ object.$

1.1.1 Code:

The **??**rd formula is used to calculate the semi-major axis. Any of the aforementioned equations can be used.

```
a = 1/((2/magr)-(magv^2/mu));
fprintf("Semi-major Axis(a) = %8.4f km \n",a);
```

Listing 2: Semi-Major Axis

1.2 Eccentricity(e)

This the parameter of the conic section which determines its shape. It is defined as the ratio of distance b/w two foci to the Major-axis. The following table lists out the orbit shape depending on it's eccentricity.

Eccentricity	**Orbit Shape**
$e = 0$	Circle
$0 < e < 1$	Ellipse
$e = 1$	Parabola
$e > 1$	Hyperbola
$e = \infty$	Straight Line

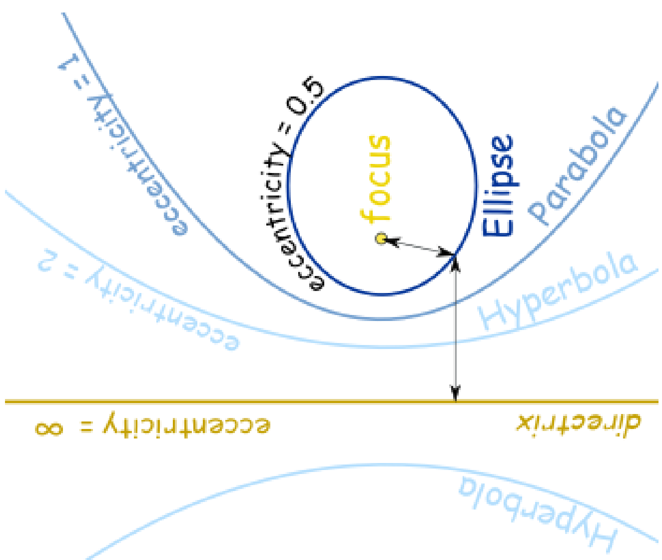

Figure 2: Eccentricity of Different Orbits

$$e = \frac{c}{a} \tag{4}$$

$$e = \frac{1}{\mu}\left[\left(\frac{v^2-\mu}{r}\right) \times \overrightarrow{r} - (\overrightarrow{r}.\overrightarrow{v}) \times \overrightarrow{v}\right] \tag{5}$$

$$\begin{aligned}
&where, \\
r_a \;\&\; r_p &= Radius\ of\ Apogee\ \&\ Perigee\ respectively, \\
\mu &= Standard\ Gravitational\ Parameter, \\
\overrightarrow{v}\ \&\ v &= Velocity\ vector\ \&\ Magnitude\ of\ Velocity\ Vector\ at\ r, \\
\overrightarrow{r}\ \&\ r &= Radius\ Vector\ \&\ Magnitude\ of\ Radius\ Vector.
\end{aligned}$$

1.2.1 Code:

The **??**th equation is used to calculate the eccentricity as we have the required variables.

```
e = (1/(mu))*((magv.^2-(mu)/(magr))*r-(dot(r,v))*v);
mage = norm(e);
fprintf("Eccentricity(e) = %8.4f \n",mage)
fprintf("Eccentricity Vector is = %4.3fi+%4.3fj+%4.3fk\n% \n",e(1),e(2),e(3))
```

Listing 3: Eccentricity

1.3 Inclination(*i*)

It is the tilt of the orbit w.r.t the equatorial plane, measured at ascending node[2]. It can also be defined as the angle from $\hat{K}$ unit vector to the specific angular momentum vector $\overrightarrow{h}$. There are 4 types of orbits. They are classified as below:

Inclination	Orbital Type
$0^o\ or\ 180^0$	Equatorial
90^o	Polar
$0^o \le i < 90^o$	Prograde(In the direction of the Earth's rotation)
$90^o < i \le 90^o$	Retrograde(Against the direction of the Earth's Rotation)

[2] The point where the orbit passes upward through the equatorial plane

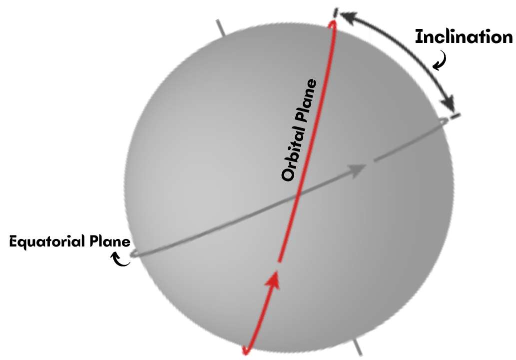

Figure 3: Orbital Inclination

$$i = \cos^{-1}\left(\frac{\overrightarrow{h}.\overrightarrow{K}}{|h|}\right) \tag{6}$$

$$\begin{aligned} &where, \\ &\quad \overrightarrow{h} = Specific\ Angular\ Momentum, \\ &\quad \overrightarrow{K} = Unit\ Vector\ in\ Z\ direction, \\ &\quad |h| = Magnitude\ of\ Specific\ Angular\ Momentum. \end{aligned}$$

1.3.1 Code:

```
i = acos(dot(h,K)/magh)*180/pi;
fprintf("Inclination(i) = %8.4f degrees \n",i)
```

Listing 4: Inclination

1.4 Right Ascension of Ascending Node(RAAN)(Ω)

RAAN horizontally orients the ascending node of the orbit w.r.t the Equatorial plane's vernal equinox, measured in equatorial plane This is not defined when the inclination is 0^o or 180^o. It's lies between 0^0 to 360^0.

$$\Omega = \cos^{-1}\left(\frac{\overrightarrow{I}.\overrightarrow{n}}{|n|}\right) \tag{7}$$

$$\begin{aligned} &where, \\ &\quad \overrightarrow{n} = Nodal\ vector(Its\ the\ vector\ that\ joins\ the\ ascending\ node\ and\ the\ descending\ node). \end{aligned}$$

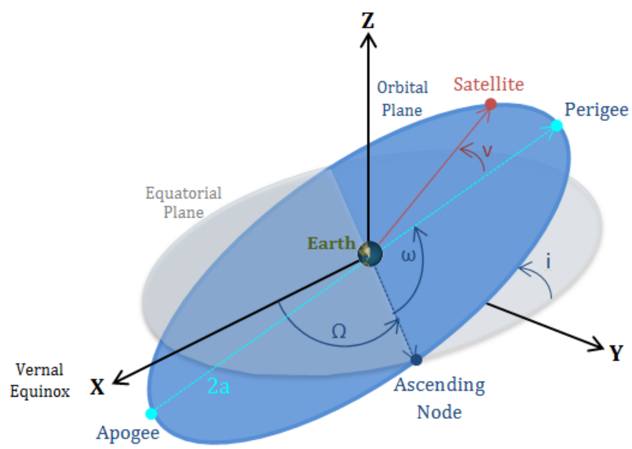

Figure 4: Right Ascension of Ascending Node.

1.4.1 Code:

```
ohm = acos(dot(I,n)/magn)*180/pi;
fprintf("Right Ascension of Ascending Node = %8.4f degrees \n",ohm)
```

Listing 5: Right Ascension of Ascending Node

1.5 Argument of Perigee(ω)

Argument of Perigee defines the orientation of the ellipse in the orbital plane, as an angle measured from the ascending node to the periapsis measure in the direction of the spacecraft's motion. This is not defined when inclination is 0^o or 180^o or eccentricity is 0. It's lies between 0^0 to 360^0.
In the figure(**??**) the argument perigee is represented as "ω".

$$\omega = \cos^{-1}\left(\frac{\vec{n}\,.\,\vec{e}}{|n||e|}\right) \tag{8}$$

$$where,$$
$$\vec{e} = EccentricityVector.$$

1.5.1 Code:

```
omega = acos(dot(n,e)/(magn*mage))*180/pi;
fprintf("Argument of Perigee(\omega) = %8.4f degrees \n",omega)
```

Listing 6: Argument of Perigee

1.6 True Anomaly(ν)

True Anomaly defines the position of the spacecraft w.r.t perigee. It's the angle between the spacecraft and the perigee of the orbit. It's lies between 0^0 to 360^0.
In the figure (**??**) the true anomaly is denoted as "ν".
It cannot be defined when the eccentricity is 0 as there is no perigee to have a reference with.

$$\nu = \cos^{-1}\left(\frac{\overrightarrow{e}.\overrightarrow{r}}{|e||r|}\right) \tag{9}$$

$$where,$$
$$\overrightarrow{r} = RadiusVector$$

1.6.1 Code:

```
nu = acos(dot(e,r)/(mage*magr))*180/pi;
fprintf("True Anomaly(v) = %8.4f degrees \n",nu)
```

Listing 7: True Anomaly

2 Alternate Orbital Elements

As mentioned above $\Omega,\ \omega,\ \&\ \nu$ are not defined during the conditions mentioned alongside it. It is also tabulated below. To correct this deficiency we use Alternate Orbital elements to replace those COE.

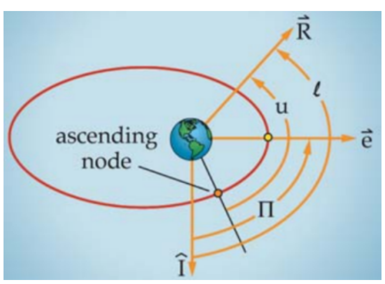

Figure 5: Alternate Orbital Elements

COE	Condition when it's undefined
a	Never
e	Never
i	Never
Ω	$i = 0^0\ or\ 180^o$
ω	$i = 0^0\ or\ 180^o$ or $e = 0$
ν	$e = 0$

Table 1: Conditions

2.1 Longitude of Perigee(Π)

Longitude of Perigee is the angle from the principle direction to perigee. This is used whenever inclination is either 0^o or 180^o as there is no ascending node. It's lies between 0^0 to 360^0.

In the figure(**??**) Longitude of perigee is represented as "Π".

$$\Pi = \cos^{-1}\left(\frac{\overrightarrow{I}\,.\,\overrightarrow{e}}{|e|}\right) \tag{10}$$

2.1.1 Code:

The if condition is used to check the inclination of the orbital plane and then enter the loop if inclination is either 0^0 or 180^0.

```
if i == 0 || i == 180
    if mage ~= 0 %Elliptical Orbit
        pie = acos(dot(I,e)/mage);
        fprintf("Longitude of Perigee = %8.4f degrees \n",pie)
```

Listing 8: Longitude of Perigee

2.2 True Longitude(l)

True Longitude is the angle from the principle direction to the spacecraft's position. This is used whenever there is no perigee and the the inclination is either 0^0 or 180^0. It's lies between 0^0 to 360^0.
In the figure(**??**)True Longitude is represented as "l"

$$l = \cos^{-1}\left(\frac{\overrightarrow{I}\,.\,\overrightarrow{r}}{|r|}\right) \tag{11}$$

2.2.1 Code:

These lines are a continuation of the previous section as the previous section was written for Elliptical orbits that are equatorial and this section is for circular orbits that are equatorial.

```
  elseif mage == 0 %Circular Orbit
          eal = acos(dot(I,r)/magr);
          fprintf("True Longitude = %8.4f degrees \n",eal)
      end
```

Listing 9: True Longitude

2.3 Argument of latitude(u)

Argument of Latitude is the angle from ascending to the spacecraft's position. This is used whenever a perigee is absent(i.e., e=0,Circular Orbit). It's lies between 0^0 to 360^0.
In the figure(**??**) Argument of latitude is represented as "u".

$$u = \cos^{-1}\left(\frac{\overrightarrow{n}.\overrightarrow{r}}{|n||r|}\right) \tag{12}$$

2.3.1 Code:

This should be the second condition as i could be zero too but then we have to use l too and that combined case has been defined in the previous section. So, in this section we just define the case with e equals 0.

```
elseif mage == 0
    yu = acos(dot(n,r)/magn*magr);
    fprintf("Argument of Latitude = %8.4f degrees \n",yu)
end
```

Listing 10: Longitude of Perigee

3 Numerical Example

1. **Given r** $= \ 8250\hat{i} + 390\hat{j} + 6900\hat{k}$**, v** $= \ -0.70\hat{i} + 6.6\hat{j} - 0.6\hat{k}$**, find the corresponding orbital elements.**
   Solution: Code: When the code is executed the vectors are given as input in the form of $3{\times}1$ array. If the input is not given properly then an error will pop-up.

```
Enter the Position Vector: [8250 390 6900]
Enter the Velocity Vector: [-0.7 6.6 -0.6]
end
```

Listing 11: Solving the Example-1

Once the inputs are given the code will calculate the orbital elements as mentioned in previous sections and the results will be displayed as follows:

```
Orbital Elements:
Semi-major Axis(a) = 13437.0788 km
Eccentricity(e) =   0.2229
Inclination(i) =  39.9115 degrees
Right Ascension of Ascending Node(Ohm) = 269.8498 degrees
Argument of Perigee(w) = 125.4009 degrees
True Anomaly(v) =  33.2089 degrees
```

Listing 12: Results

References

[1] Bate, Roger R., Donald D. Muller and Jerry E. White, *"Fundamentals Of Astrodynamics"*,New York, NY, Dover Publications,1971.

[2] Federal Aviation Administration, *"Advanced Aerospace Medicine Online"*, 2017.

[3] Physics LibreTexts™, *"Keplerian Elements Tutorial"*, 2020.

[4] L.Ramkiran, Code on gist.GitHub, *"SNG Assigniment 01"*.