Cartesian to Orbital Simulation

Ricard Okimoto and Sung wan Park

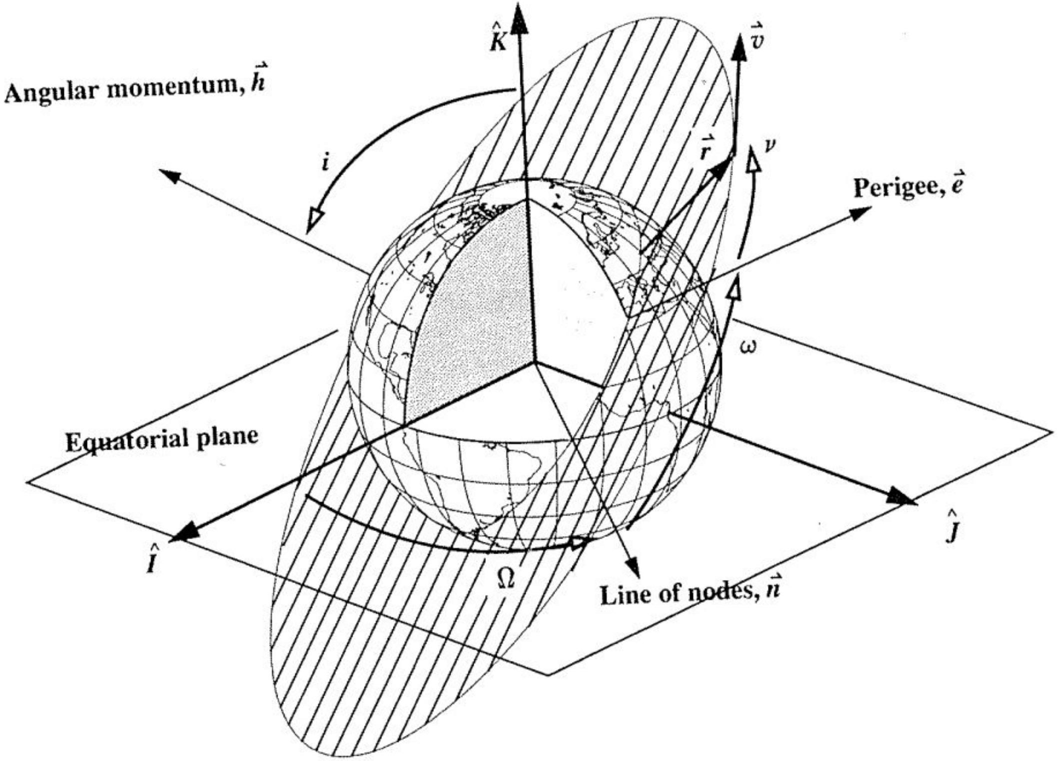
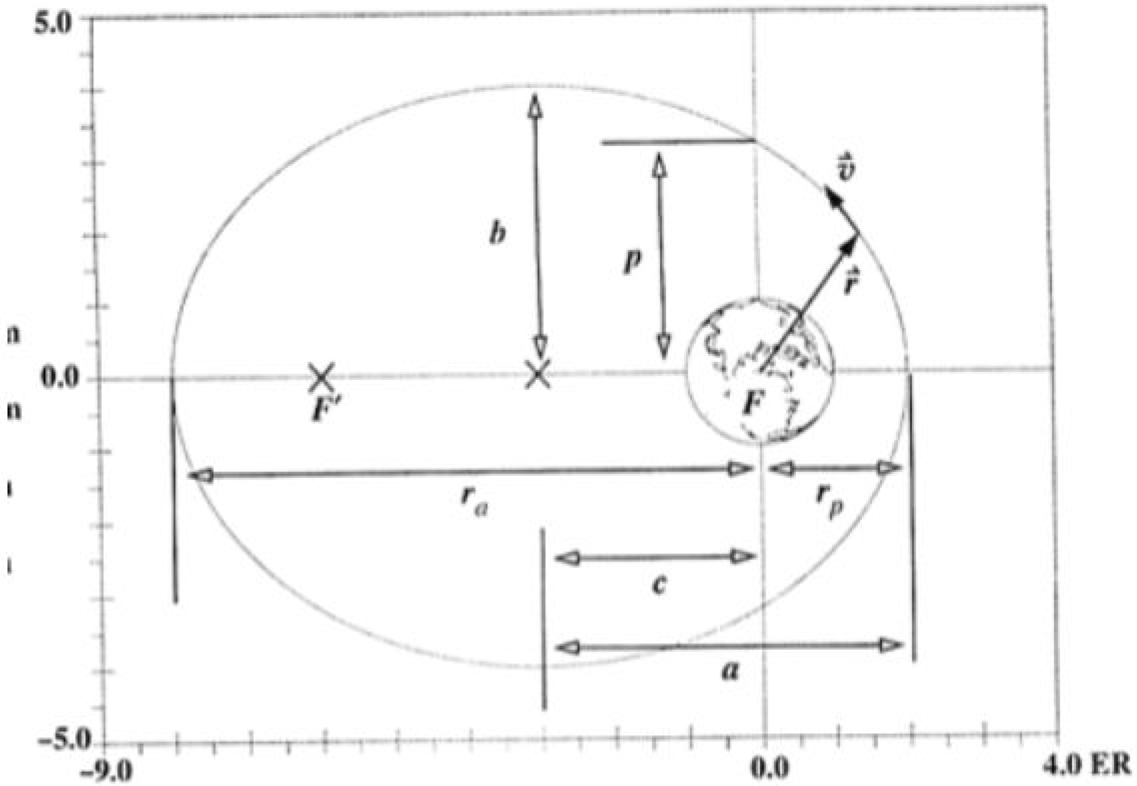
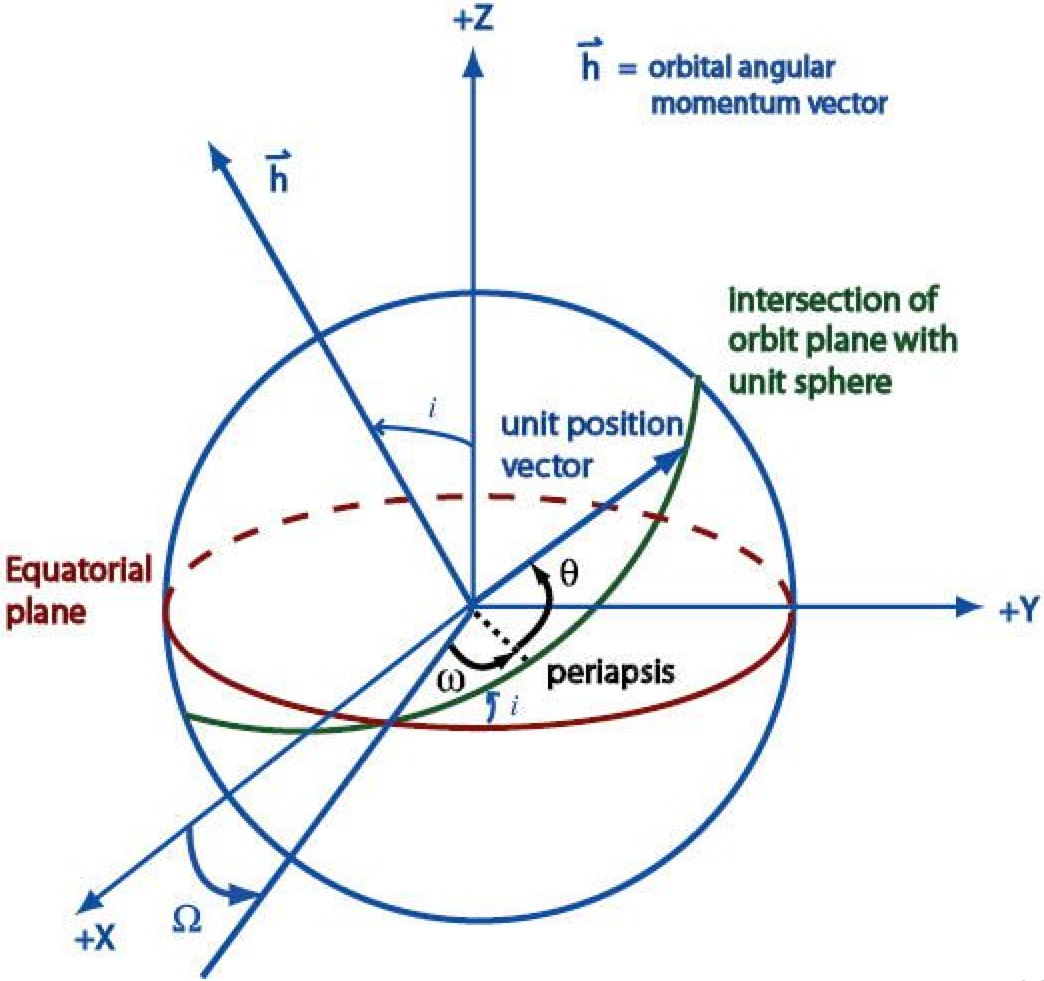


Figure 2-12. **Classical Orbital Elements.** The six classical orbital elements are the *semimajor axis, a*; *eccentricity, e*; *inclination, i*; *longitude of ascending node,* Ω—often referred to as simply the *node*; *argument of perigee,* ω; and *true anomaly,* ν. I haven't shown scale and shape elements (*a* and *e*) because I've introduced them in Chap. 1.

Elliptical orbit





https://www.agi.com/resources/help/online/AGIComponentsJava/SVG/orbitalElements.jpg

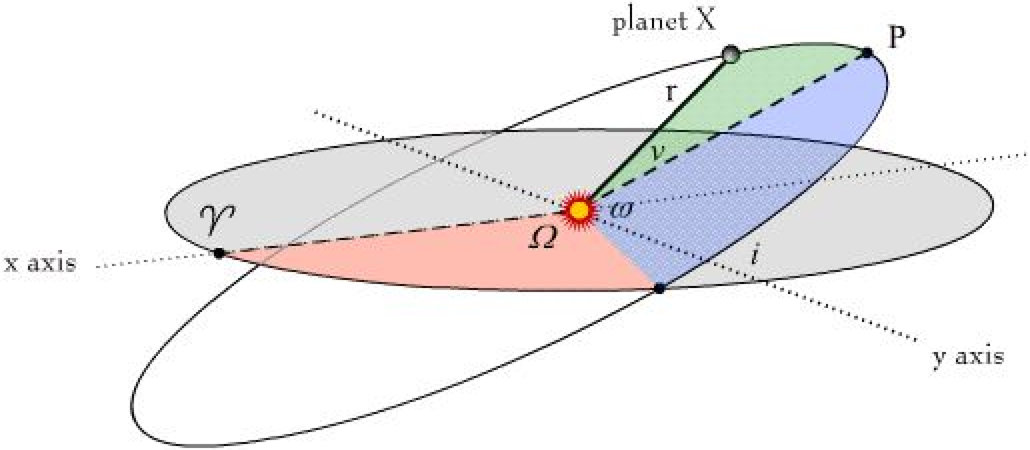


Figure 2: Planet X's orbit as it intersects with the plane of the Sun's equator.

r = distance between planet X and the Sun

P *= perihelion (point where X is closest to Sun)*

♈ *= first point of Aries*

i = angle between plane of Sun's equator and planet X's orbit

ϖ = longitude of perihelion

Ω = longitude of asending node

http://www.davidcolarusso.com/astro/images/figure6.gif

ALGORITHM 5: ***ELORB*** $(\vec{r}_{IJK}, \vec{v}_{IJK} \Rightarrow p, a, e, i, \Omega, \omega, \nu\ (u, \lambda_{true}, \tilde{\omega}_{true})\ )$

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

$$\vec{n} = \hat{K} \times \vec{h}$$

$$\vec{e} = \frac{\left(v^2 - \frac{\mu}{r}\right)\vec{r} - (\vec{r} \cdot \vec{v})\,\vec{v}}{\mu} \qquad e = |\vec{e}|$$

$$\xi = \frac{v^2}{2} - \frac{\mu}{r}$$

IF $e \neq 1.0$ THEN

$$a = -\frac{\mu}{2\xi}$$

$$p = a\,(1 - e^2)$$

ELSE

$$p = \frac{h^2}{\mu} \text{ and } a = \infty$$

$$\cos(i) = \frac{h_K}{|\vec{h}|}$$

$$\cos(\Omega) = \frac{n_I}{|\vec{n}|} \qquad \text{IF } (n_J < 0) \text{ THEN } \Omega = 360° - \Omega$$

$$\cos(\omega) = \frac{\vec{n} \cdot \vec{e}}{|\vec{n}||\vec{e}|} \qquad \text{IF } (e_K < 0) \text{ THEN } \omega = 360° - \omega$$

$$\cos(\nu) = \frac{\vec{e} \cdot \vec{r}}{|\vec{e}||\vec{r}|} \qquad \text{IF } (\vec{r} \cdot \vec{v} < 0) \text{ THEN } \nu = 360° - \nu$$

Special Cases

Elliptical equatorial:

$$\cos(\tilde{\omega}_{true}) = \frac{e_I}{|\vec{e}|} \qquad \text{IF } (e_J < 0) \text{ THEN } \tilde{\omega}_{true} = 360° - \tilde{\omega}_{true}$$

$$\cos(u) = \frac{\vec{n} \cdot \vec{r}}{|\vec{n}||\vec{r}|} \qquad \text{IF } (r_K < 0) \text{ THEN } u = 360° - u$$

Circular equatorial:

$$\cos(\lambda_{true}) = \frac{r_I}{|\vec{r}|} \qquad \text{IF } (r_J < 0) \text{ THEN } \lambda_{true} = 2\pi - \lambda_{true}$$

```
clc

%prompt='what is r in vector format in km';

%r=input(prompt)

%prompt='what is v in vector format in km/s';

%v=input(prompt)

r=[6524.834 6862.875 6448.296]

v=[4.901327 5.533756 -1.976341]

r1=norm(r);

v1=norm(v);

h=cross(r,v);

u=398600.4415;

k=[0 0 1];

n=cross(k,h);

E=(v1^2/2)-u/r1;

e=((v1^2-u/r1)*r-(dot(r,v)*v))/u;

e1=norm(e);

if e~=1

        a=-u/(2*E);

        p=a*(1-e1^2);

else

        a=inf;

        p=norm(h)^2/u;

end

I=57.2958*(acos(h(1,3)/norm(h)));

omega=57.2958*(acos(n(1,1)/norm(n)));

if n(1,2)<0

omega=57.2958*(2*pi-acos(n(1,1)/norm(n)));

else

        omega=57.2958*(acos(n(1,1)/norm(n)));

end

w=57.2958*(acos(dot(n,e)/(norm(n)*e1)));

if e(1,3)<0

        w=57.2958*(2*pi-acos(dot(n,e)/(norm(n)*e1)
));

else

        w=57.2958*(acos(dot(n,e)/(norm(n)*e1)));

end

nu=57.2958*(acos(dot(e,r)/(r1*e1)));

if dot(r,v)<0

        nu=57.2958*(2*pi-acos(dot(e,r)/(r1*e1)));
```

```
else
    nu=57.2958*(acos(dot(e,r)/(r1*e1)));
end
%eliptical equitorial
wt=57.2958*(acos(e(1,1)/e1));
if e(1,2)<1
    wt=57.2958*(2*pi-acos(e(1,1)/e1));
else
    wt=57.2958*(acos(e(1,1)/e1));
end
wprime=omega+w
%circular inclined
ut=57.2958*(acos(dot(n,r)/(norm(n)*r1)));
if r(1,3)<0
    ut=57.2958*(2*pi-acos(dot(n,r)/(norm(n)*r
1)));
else
    ut=57.2958*(acos(dot(n,r)/(norm(n)*r1)));
end
Lt=57.2958*acos(r(1,1)/r1);
if r(1,2)<0
    Lt=57.2958*acos(2*pi-r(1,1)/r1);
else
    Lt=57.2958*acos(r(1,1)/r1);
end
```