2022-2023_Partiel

February 6, 2024

[1]:
```
import math
```

0.1 Question 15

Determine the value of the semi-major axis in km :

$$z_A = r_A - R_{terre}$$
$$z_P = r_P - R_{terre}$$

$$\equiv z_A = a(1+e) - R_{terre}$$
$$\equiv a(1+e) = -R_{terre} - z_A$$
$$\equiv a = -(\frac{1}{(1+e)}(-R_{terre} - z_A))$$

[2]:
```
z_a = 151455.47 #km
z_p = 11256.53 #km
w = -150  # Assume the value of w in degrees
i = 79.20  # Assume the value of i in degrees
e = 0.7990
la = 0
L_omega = -51.6
rayon_terrestre = 6378
mu = 398600
alpha = 360/86164
```

0.2 FONCTION NÉCESSAIRES

[3]:
```
def calculate_time(v, periapsis=False):
    term1 = -math.sqrt(a**3 / mu) if periapsis else math.sqrt(a**3 / mu)
    term2 = math.asin(math.sqrt(1 - e**2) * math.sin(math.radians(v)) / (1 + e␣
↪* math.cos(math.radians(v))))
    term3 = e * math.sqrt(1 - e**2) * math.sin(math.radians(v)) / (1 + e * math.
↪cos(math.radians(v)))

    v_rad = math.radians(v)
```

```python
    if -v_c <= v_rad <= v_c:
        t_p = term1 * (term2 - term3)

    elif v_c < v_rad < 2*math.pi - v_c:
        t_p = term1 * (math.pi - term2 - term3)

    elif v_rad > 2*math.pi - v_c:
        t_p = term1 * (2*math.pi + term2 - term3)

    elif -2*math.pi + v_c < v_rad < -v_c:
        t_p = term1 * (-math.pi - term2 - term3)

    elif v_rad < -2*math.pi + v_c:
        t_p = term1 * (-2*math.pi + term2 - term3)

    return t_p


def calculate_la(v):
    v_rad = math.radians(v)
    w_rad = math.radians(w)
    i_rad = math.radians(i)

    la = math.degrees(math.asin(math.sin(w_rad + v_rad) * math.sin(i_rad)))
    return la

def calculate_L0(la, i, w, v):
    la_rad = math.radians(la)
    i_rad = math.radians(i)
    w_rad = math.radians(w)
    v_rad = math.radians(v)

    tan_value = math.tan(la_rad) / math.tan(i_rad)

    if -1 <= tan_value <= 1:
        L0 = math.degrees(math.asin(tan_value))
    else:
        if tan_value > 1:
            tan_value = 1
        elif tan_value < -1:
            tan_value = -1

        L0 = math.degrees(math.asin(tan_value))


    if -w_rad - math.radians(90) <= v_rad <= -w_rad + math.radians(90):
```

```
        pass
    elif -w_rad - math.radians(450) <= v_rad <= -w_rad - math.radians(270):
        L0 = -360 + L0
    elif -w_rad - math.radians(270) <= v_rad <= -w_rad - math.radians(90):
        L0 = -180 - L0
    elif -w_rad + math.radians(90) <= v_rad <= -w_rad + math.radians(270):
        L0 = 180 - L0
    elif -w_rad + math.radians(270) <= v_rad <= -w_rad + math.radians(450):
        L0 = 360 + L0
    elif -w_rad + math.radians(450) <= v_rad <= -w_rad + math.radians(630):
        L0 = 540 - L0
    else:
        L0 = None

    return L0
```

[4]:
```
a = - (1/(1+e)*(-rayon_terrestre-z_a))
a
```

[4]: 87734.00222345749

0.3 Question 16

Determine the radius of periapsis r_p in km :

$$r_P = a(1 - e)$$

[5]:
```
r_p = a*(1-e)
r_p
```

[5]: 17634.53444691495

0.4 Question 17

Determine the radius of apoapsis r_A in km :

$$r_P = a(1 + e)$$

[6]:
```
r_a = a*(1+e)
r_a
```

[6]: 157833.47

0.5 Question 18

Determine the orbital period T in secondes :

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

[7]:
```
T = 2*math.pi*math.sqrt(a**3/mu)
T
```

[7]: 258620.61085459465

0.6 Question 19

Determine the velocity at periapsis v_P in km/s :

Velocities at apoapsis and periapsis, V_A and V_P

Nous avons,

$$r_A V_A = r_P V_P$$

et

$$W = \frac{V^2}{2} - \frac{\mu}{r} = -\frac{\mu}{2a}$$

$$\equiv \frac{V_P^2}{2} - \frac{\mu}{r_P} = -\frac{\mu}{2a}$$

$$\equiv \frac{V_P^2}{2} = \frac{\mu}{r_P} - \frac{\mu}{2a}$$

$$\equiv V_P = \sqrt{\frac{2\mu}{r_P} - \frac{\mu}{a}}$$

[8]:
```
V_P = (((2*mu)/r_p) - (mu/a))**0.5
V_A = (r_p*V_P)/r_a

print("Vitesse au periapsis:", round(V_P,2), "km/s")
```

```
Vitesse au periapsis: 6.38 km/s
```

0.7 Question 20

Determine the velocity at apoapsis v_A in km/s :

$$V_A = \frac{r_P V_P}{r_A}$$

[9]:
```python
V_A = (r_p*V_P)/r_a #0.95

print("Vitesse à l'apoapsis:", round(V_A,2), "km/s")
```

```
Vitesse à l'apoapsis: 0.71 km/s
```

0.8 Question 21

The passing time at periapsis is a constant of the problem, determined by assuming that the time reference is set when the satellite passes at the ascending node. Determine the value of the passing time at periapsis t_P in seconds

$v < -2\pi + v_c$ **donc faut ajouter** $-2\pi+$ **dans l'equation de t-tp**

$$t_p = -\sqrt{\frac{a^3}{\mu}}[-2\pi + sin^{-1}(\frac{\sqrt{1-e^2}sin(v)}{1+ecos(v)}) - e\frac{\sqrt{1-e^2}sin(v)}{1+ecos(v)}]$$

[10]:
```python
v_c = math.acos(-e)
tp = calculate_time(v=-w,periapsis=True)
print('Temps à periapsis =>',tp)

print(i)
```

```
Temps à periapsis => -41583.71479867027
79.2
```

0.9 Question 22 - 37

What is the value of the passing time for the true anomaly -200°, cell a*, in seconds ?

v (°)	t (sec.)	la (°)	L0 (°)	LS (°)
-200	a*	9.4	h*	k*
-160	b*	46.1	i*	44.0
-120	-56265.2	e*	-270.0	-86.5
+40	c*	f*	-137.1	l*
+80	-36175.7	g*	-49.2	56.6
+160	20622.4	9.4	3.4	m*
+200	d*	46.1	j*	n*

Several cases can occur if one compares v to v_c:

if $-v_c \leq v \leq +v_c$

$$t - t_p = \sqrt{\frac{a^3}{\mu}}[sin^{-1}(\frac{\sqrt{1-e^2}sin(v)}{1+ecos(v)}) - e\frac{\sqrt{1-e^2}sin(v)}{1+ecos(v)}]$$

if $+v_c < v < 2\pi - v_c$

$$t - t_p = \sqrt{\frac{a^3}{\mu}}[\pi - sin^{-1}(\frac{\sqrt{1-e^2}sin(v)}{1+ecos(v)}) - e\frac{\sqrt{1-e^2}sin(v)}{1+ecos(v)}]$$

if $v > 2\pi - v_c$

$$t - t_p = \sqrt{\frac{a^3}{\mu}}[2\pi + sin^{-1}(\frac{\sqrt{1-e^2}sin(v)}{1+ecos(v)}) - e\frac{\sqrt{1-e^2}sin(v)}{1+ecos(v)}]$$

if $-2\pi + v_c < v < -v_c$

$$t - t_p = \sqrt{\frac{a^3}{\mu}}[-\pi - sin^{-1}(\frac{\sqrt{1-e^2}sin(v)}{1+ecos(v)}) - e\frac{\sqrt{1-e^2}sin(v)}{1+ecos(v)}]$$

if $v < -2\pi + v_c$

$$t - t_p = \sqrt{\frac{a^3}{\mu}}[-2\pi + sin^{-1}(\frac{\sqrt{1-e^2}sin(v)}{1+ecos(v)}) - e\frac{\sqrt{1-e^2}sin(v)}{1+ecos(v)}]$$

$$l_a = sin^{-1}(sin(w+v)sini)$$

$$L_0 = sin^{-1}(\frac{tan(la)}{tan(i)})$$

if $-\omega - 90 \leq v \leq -\omega + 90$

$$L_0 = sin^{-1}(\frac{tan(la)}{tan(i)})$$

if $v < -\omega - 90$

$$L_0 = -180 - sin^{-1}(\frac{tan(la)}{tan(i)})$$

if $v > -\omega + 90$

$$L_0 = +180 - sin^{-1}(\frac{tan(la)}{tan(i)})$$

etc

Whilethe Earthis«turning»:

–The latitude calculatedin the «fixed» case reamains!!!

–The longitude L_s corrected by the Earth rotation depending on L_0

$$L_S = L_\omega(t_0) + L_0 - \dot{\alpha}(t - t_0)$$

$$L_S = L_\omega(t_0) + L_0 - \dot{\alpha}t$$

We consider that the satellite goes through ω to t_0 ; $(t - t_0)$ and thatisits passing time to reach S.

$L_\omega(t_0)$ is given

Or, $\dot{\alpha} = \frac{360}{86164}°/s$

[11]:
```
v_angles = [-200, -160, -120, 40, 80, 160, 200]


t = []

for j in v_angles:
    t.append( round(calculate_time(v=j) + tp,2) )

print("t =>", t)


La = []


for index, v in enumerate(v_angles):
    result_la = calculate_la(v)
    La.append(round(result_la,2))

print('la =>',la )

L0 = [ ]

for index , v in enumerate ( v_angles ) :
    result_la = calculate_la ( v )
    result_L0 = calculate_L0 ( result_la , i , w , v )
    L0.append ( round(result_L0,2) )

print('l0 =>',L0)

Ls = []

for index,l in enumerate(L0) :
    ls = L_omega + l - alpha * t[index]
    Ls.append(round(ls,2))

print('Ls =>',Ls)
```

```
t => [-237998.26, -103789.78, -56265.24, -39502.91, -36175.74, 20622.35,
154830.84]
la => 0
l0 => [-358.11, -347.41, -270.0, -152.76, -27.24, 1.89, 12.59]
Ls => [584.67, 34.63, -86.52, -39.31, 72.31, -135.87, -685.91]
```

 Visulisation des donnés

[12]:
```python
import pandas as pd

data = {
    'v (°)': v_angles,
    't (sec.)': t,
    'la (°)': La,
    'L0 (°)': L0,
    'Ls (°)' : Ls,
}

df = pd.DataFrame(data)

df
```

[12]:

	v (°)	t (sec.)	la (°)	L0 (°)	Ls (°)
0	-200	-237998.26	9.82	-358.11	584.67
1	-160	-103789.78	48.81	-347.41	34.63
2	-120	-56265.24	79.20	-270.00	-86.52
3	40	-39502.91	-67.38	-152.76	-39.31
4	80	-36175.74	-67.38	-27.24	72.31
5	160	20622.35	9.82	1.89	-135.87
6	200	154830.84	48.81	12.59	-685.91