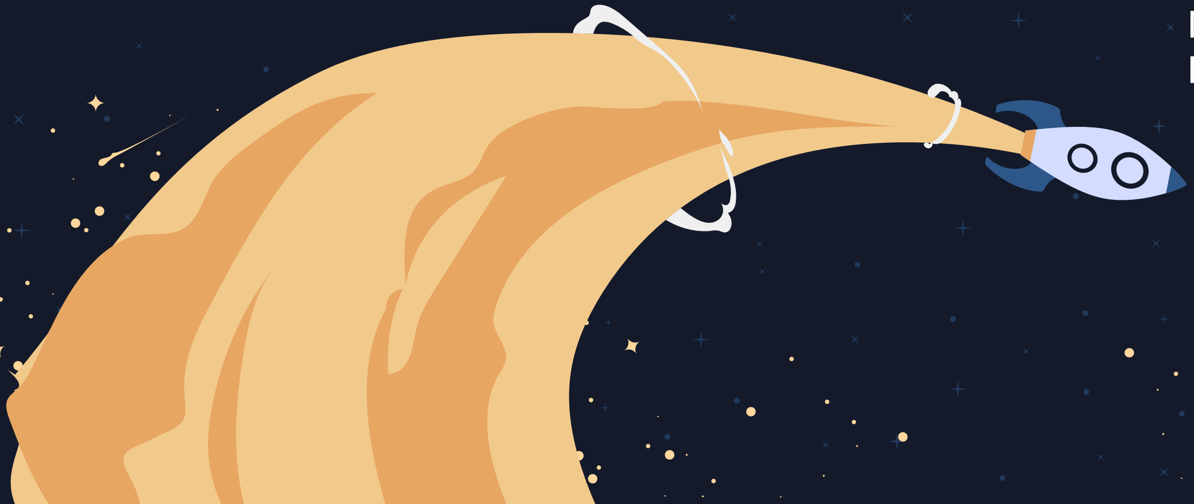
Orbitas de uma sonda a volta de jupiter

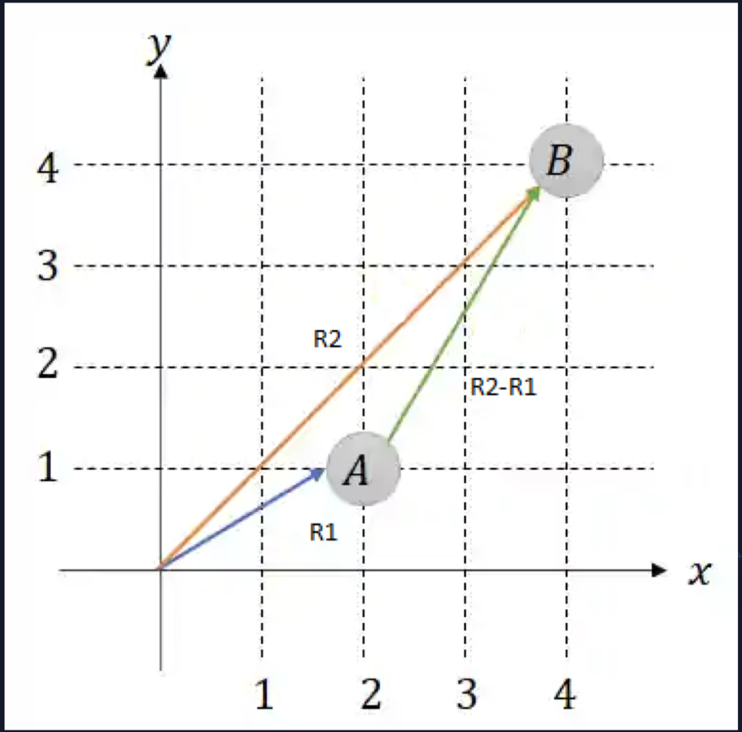
Projeto realizado por: Diogo
David
Maria
Lourenço



Índice

- Vetores e fórmulas
- Programação
- Resultados
- Conclusões

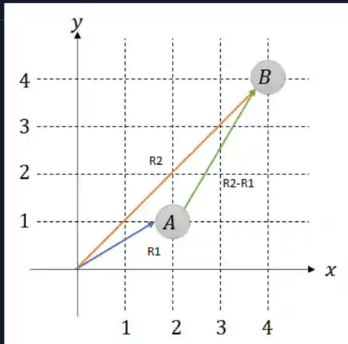
Vetores de posição



Força gravítica

$$Fg = G \times \frac{m.M}{r^2} \quad \leftrightarrow \quad \overrightarrow{Fg} = G \times \frac{m.M}{r^2} \times \hat{e}_{\overrightarrow{R2}-\overrightarrow{R1}} \quad \leftrightarrow$$

$$\leftrightarrow \quad \overrightarrow{Fg} = G \times \frac{m.M}{||\overrightarrow{R2} - \overrightarrow{R1}||^2} \times \frac{(\overrightarrow{R2} - \overrightarrow{R1})}{||\overrightarrow{R2} - \overrightarrow{R1}||} \quad \leftrightarrow \quad \overrightarrow{Fg} = G \times \frac{m.M.(\overrightarrow{R2} - \overrightarrow{R1})}{||\overrightarrow{R2} - \overrightarrow{R1}||^3}$$



Método de Euler

$$\begin{cases} \dfrac{d\vec{R}}{d_t} = \vec{v} \\ \dfrac{d\vec{v}}{d_t} = \vec{a} = \dfrac{\vec{F}}{m} \end{cases} \leftrightarrow \begin{cases} \vec{v} = \dfrac{\vec{R}(\Delta t + t) - \vec{R}(t)}{\Delta t} \\ \dfrac{\vec{F}}{m} = \dfrac{\vec{v}(\Delta t + t) - \vec{v}(t)}{\Delta t} \end{cases} \leftrightarrow$$

$$\leftrightarrow \begin{cases} \vec{R}(t + \Delta t) = \vec{v}\Delta t + \vec{R}(t) \\ \vec{v}(t + \Delta t) = \dfrac{\vec{F}}{m}\Delta t + \vec{v}(t) \end{cases}$$

Mola

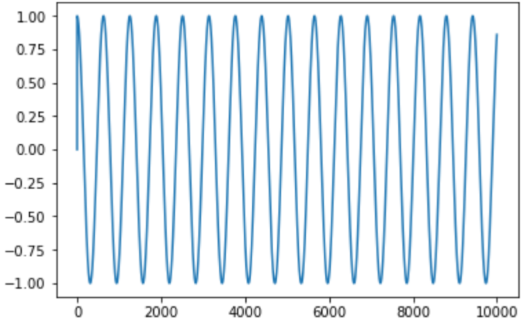
```
In [31]: def time_step(x0,v0):
             F=-1*x0
             vf=v0+(F/1)*0.01
             xf=x0+vf*0.01

             return xf,vf
```

```
In [32]: x0,v0 = 1,0
         for t in range (1,10001):
             (xf,vf)=time_step(x0,v0)
             (x0,v0)=(xf,vf)
             X[t]=x0
             print(xf,vf)
```

```
In [33]: plt.plot(X)
```

```
Out[33]: [<matplotlib.lines.Line2D at 0x15fe9f0>]
```



Python

```
In [22]: def pitágoras(a,b):
             c=((a**2)+(b**2))**0.5
             return c

In [23]: pitágoras(1,1)
Out[23]: 1.4142135623730951

In [26]: pitagoras(3,4)
Out[26]: 5.0
```

```
In [ ]: def f_grav(x0,y0,m,M):
            G=1 #6.67*10**-11
            norma=((x0**2)+(y0**2))**0.5
            R1=np.array([x0,y0])
            Fg=(G*m*M*(-R1))/((norma)**3)
            return Fg

In [54]: f_grav(0,2,1,1)
Out[54]: array([-0.  ,  0.25])
```

$$\overrightarrow{R_2}(t) = (R \times cos(wt), R \times sen(wt))$$

```
In [6]: def posjupiter(t,P,R):
            w=(2*np.pi)/P
            R2=np.array([R*np.cos(w*t),R*np.sin(w*t)])
            return R2

In [20]: def f_grav(R1,m,M,R2):
             G=6.67*10**-11
             norma=(((R1[0]-R2[0])**2)+((R1[1]-R2[1])**2))**0.5
             Fg=(G*m*M*(R2-R1))/((norma)**3)
             return Fg

In [5]: def time_step(r0,v0,m,M,R2):
            rf=r0 + v0 * 360
            F=f_grav(rf,m,M,R2)
            vf=v0 + (F/m) * 360

            return rf,vf
```

Python

```
In [220]: P = 2*np.pi / 1.7134e-8
          R = 7.649 * 10**11
          R1 =   np.array([7.592*10**11, 3.503*10**9])
          V0 = np.array([2510,10000])
          m = 1
          M = 1.9*10**27
```

```
In [222]: for i in range (0,50001):
              t=i*360
              R2=posjupiter (t,P,R)
              Rf,Vf=time_step(R1,V0,m,M,R2)
              Xs[i]=Rf[0]
              Ys[i]=Rf[1]
              Xj[i]=R2[0]
              Yj[i]=R2[1]
              V[i]=((Vf[0]**2)+(Vf[1]**2))**0.5
              Dist_jup[i] = np.sqrt((R1[0]-R2[0])**2+((R1[1]-R2[1])**2))
              R1=Rf
              V0=Vf
```

Dados utilizados

Massa de Júpiter: 1,9*10^27 Kg

Raio médio da órbita de Júpiter: 7,649*10^11 m

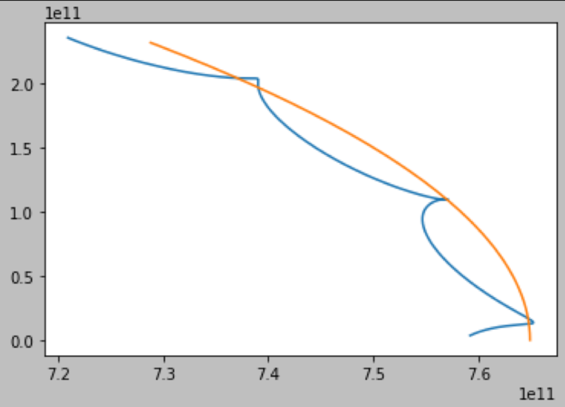
Período da órbita de Júpiter: 3,67*10^8 s

Posição inicial de Júpiter: x=7,649*10^11 m, y=0 m

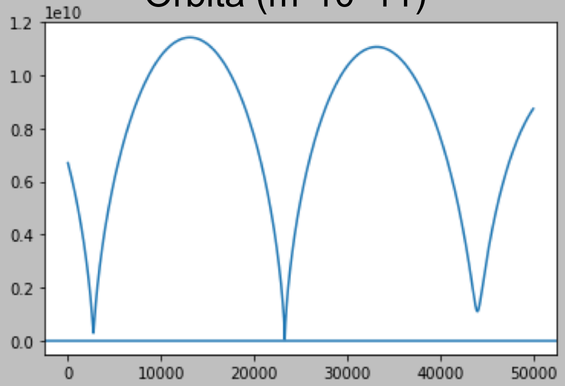
Posição inicial da sonda: x=7,592*10^11 m, y=3,503*10^9 m

Velocidade inicial da sonda: Variável

Resultados

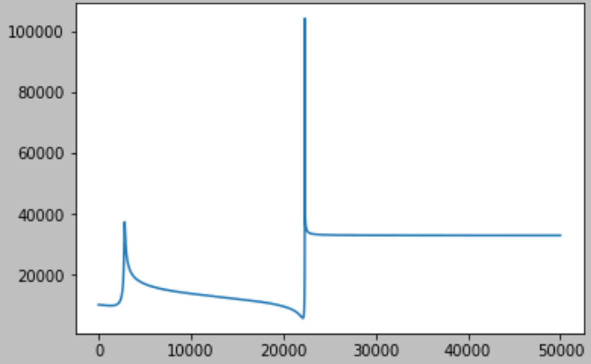


Órbita (m*10^11)



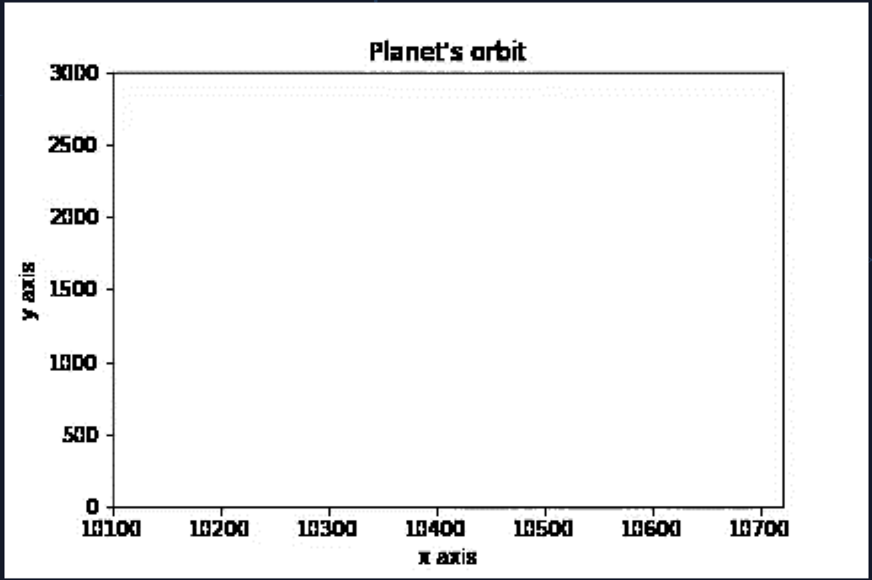
Distância a Júpiter (m*10^10)

Velocidade inicial (m/s):
(Vx=2510;Vy=10000)
Tempo de simulação: 5000 horas



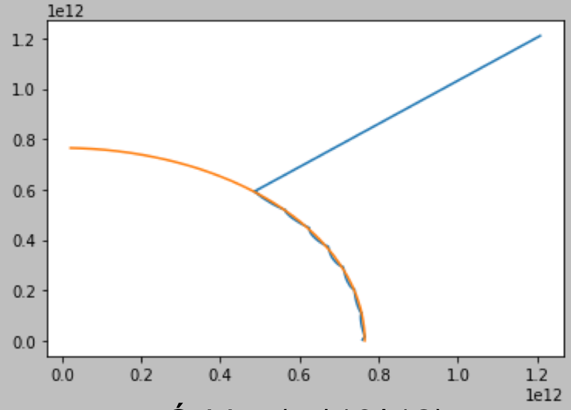
Velocidade (m/s)

Resultados



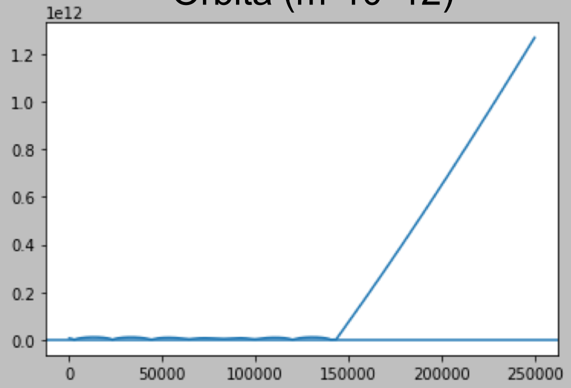
Planet's orbit
3000
2500
2000
1500
1000
500
0
y axis
10100
10200
10300
10400
10500
10600
10700
x axis

Resultados

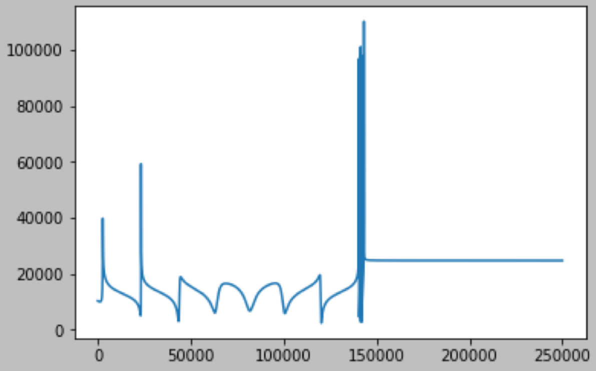


Órbita (m*10^12)

Velocidade inicial (m/s):
(Vx=2510;Vy=10000)
Tempo de simulação: 25000 horas

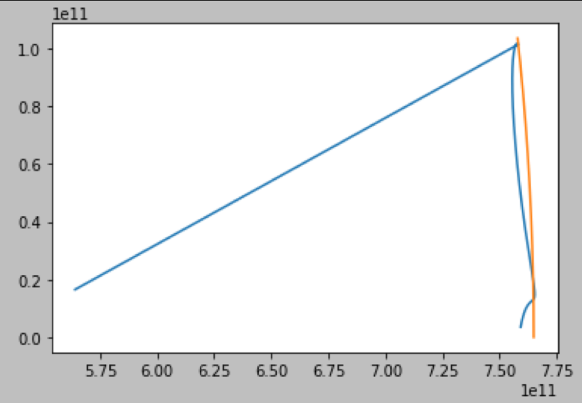


Distância a Júpiter (m*10^12)



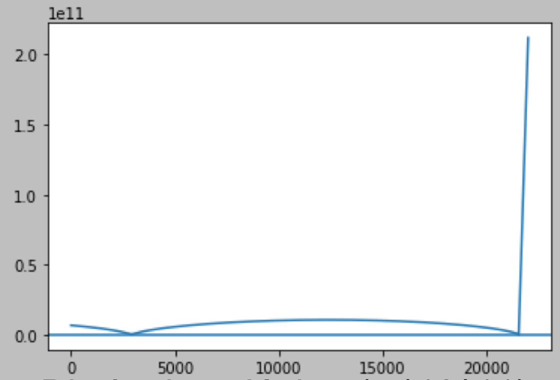
Velocidade (m/s)

Resultados

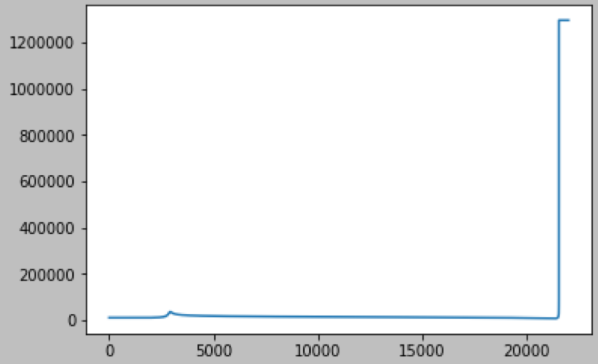


Órbita (m*10^11)

Velocidade inicial (m/s):
(Vx=2110;Vy=10000)
Tempo de simulação: 2200 horas

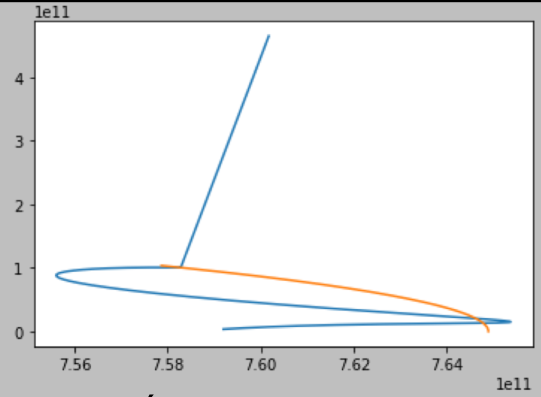


Distância a Júpiter (m*10^11)



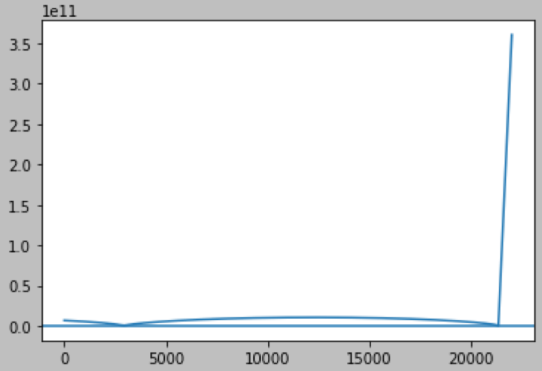
Velocidade (m/s)

Resultados

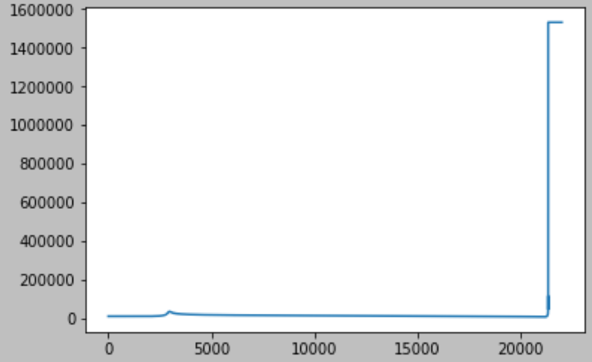


Órbita (m*10^11)

Velocidade inicial (m/s):
(Vx=2050;Vy=10000)
Tempo de simulação: 2200 horas

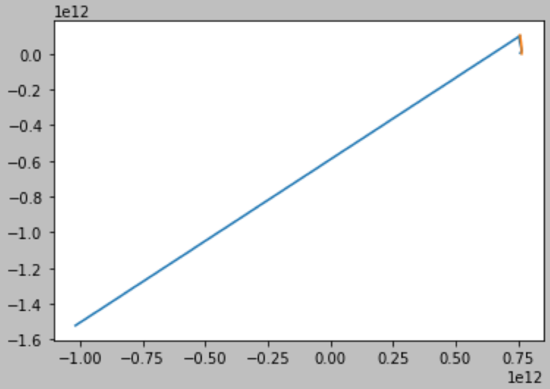


Distância a Júpiter (m*10^11)



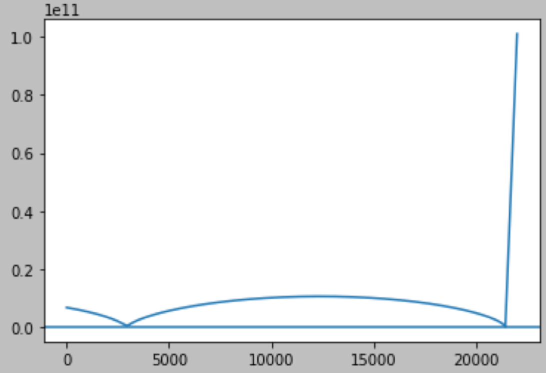
Velocidade (m/s)

Resultados

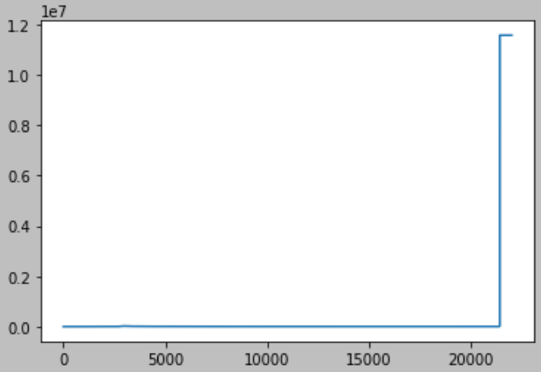


Órbita (m*10^12)

Velocidade inicial (m/s):
(Vx=2050;Vy=10000)
Tempo de simulação: 2200 horas



Distância a Júpiter (m*10^11)



Velocidade (m/s)

Conclusões