Modelação de Sistemas Físicos - Aula Prática nº6

Realização e resolução de problemas sobre:

- Movimento a 2D

- Método de Euler-Cromer

Exercício 1

Simule a órbita da Terra à volta do sol, usando o método de Euler, sabendo que a força de atração da Terra exercida pelo Sol é,

$$\mathbf{F}_{\text{grav}} = -G\frac{m\,M}{r^2}\hat{\mathbf{r}}$$

em que $\hat{\mathbf{r}} = \mathbf{r}/r$ e $\mathbf{r}$ é o vetor posição da Terra relativamente ao Sol.

Como as quantidades envolvidas são enormes, trabalhe no sistema astronómico de unidades:

Grandeza	Unidade	Definição	Valor no SI
Distância	AU	Distância média da terra ao Sol	1.498e11 m
Tempo	ano	Período da Terra à volta do Sol	3.15e7 s
Massa	M	Massa do Sol	1.989e30 kg

Considere a posição inicial da Terra $\mathbf{r}(t=0) = (1, 0)$ AU, e a velocidade inicial $\mathbf{v}(t=0) = (0, 2\pi)$ AU/ano e o Sol como fixo na origem do sistema de eixos.

Considerar o seguinte:

- $G = 6.67408 \times 10^{-11}\ \mathrm{m^3/kgs} = 4\pi^3\ \mathrm{AU^3/M\,ano}$
- $m = 5.9722 \times 10^{24}\ \mathrm{kg} = 3.003 \times 10^{-6}\ \mathrm{M}$

a) A órbita da Terra à volta do sol é fechada?

b) Consegue obter elipses?

Resolução:

Forças a atuar na Terra: gravítica (note que no sistema astronómico $M = 1$).

$$\mathbf{F}_{\text{grav}} = -G\frac{m}{r^2}\hat{\mathbf{r}}$$

Portanto a lei da aceleração é,

$$\mathbf{a} = -G\frac{\hat{\mathbf{r}}}{r^2} = -G\frac{\mathbf{r}}{r^3}$$

Lei da velocidade (resolver pelo método de Euler),

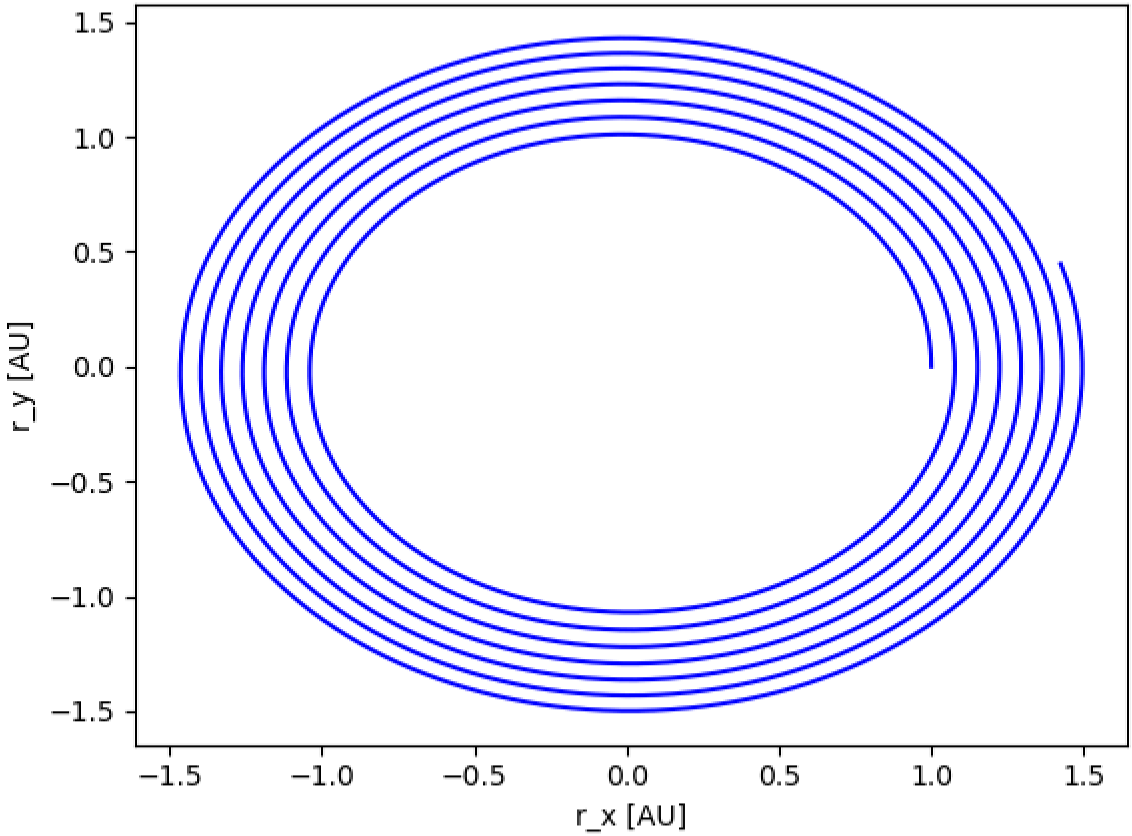
$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{a}$$

A lei do movimento (resolver pelo método de Euler),

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \mathbf{v}$$

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt

t0 = 0.0                    # condição inicial, tempo [ano]
tf = 10.0                    # limite do domínio, tempo final [ano]
dt = 0.001                  # passo [ano]
v0 = 2.0 * np.pi            # condição inicial, módulo da velocidade inicial [AU/ano]
x0 = 1.0                    # condição inicial, coordenada x da posição inicial [AU]

G = 4.0 * np.pi ** 2        # constante de gravitação [AU^3 / M ano^2]

# inicializar domínio [ano]
t = np.arange(t0, tf, dt)

# inicializar solução, aceleração a 2D [AU / ano^2]
a = np.zeros([np.size(t), 2])

# inicializar solução, velocidade [m / s]
v = np.zeros([np.size(t), 2])
v[0, :] = np.array([0, v0])  # velocidade [AU / ano] para t = 0 ano

# inicializar solução, posição [m]
r = np.zeros([np.size(t), 2])
r[0, :] = np.array([x0, 0.0])    # posição [AU] para t = 0 ano

for i in range(np.size(t) - 1):
    a[i, :] = - G * r[i, :] / np.linalg.norm(r[i, :]) ** 3   # aceleração instantânea
    v[i + 1, :] = v[i, :] + a[i, :] * dt
    r[i + 1, :] = r[i, :] + v[i, :] * dt

#plt.figure(figsize=(6,6))
#plt.xlim(-1.5, 1.5)
#plt.ylim(-1.5, 1.5)
plt.plot(r[:, 0], r[:, 1], 'b-')
plt.xlabel("r_x [AU]")
plt.ylabel("r_y [AU]")
plt.show()
```



a) A órbita da Terra à volta do sol é fechada?

Não

b) Consegue obter elipses?

Dependem da velocidade inicial, mas são na generalidade trajetórias curvilíneas abertas.

Pergunta 1:

Porque é que é razoavel considerar o Sol como fixo na origem?

Exercício 2

Implemente o método de Euler-Cromer para simular a órbita da Terra á volta do sol.

Este método integra as equações diferenciais

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{a}$$

e

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \mathbf{v}$$

da seguinte forma:

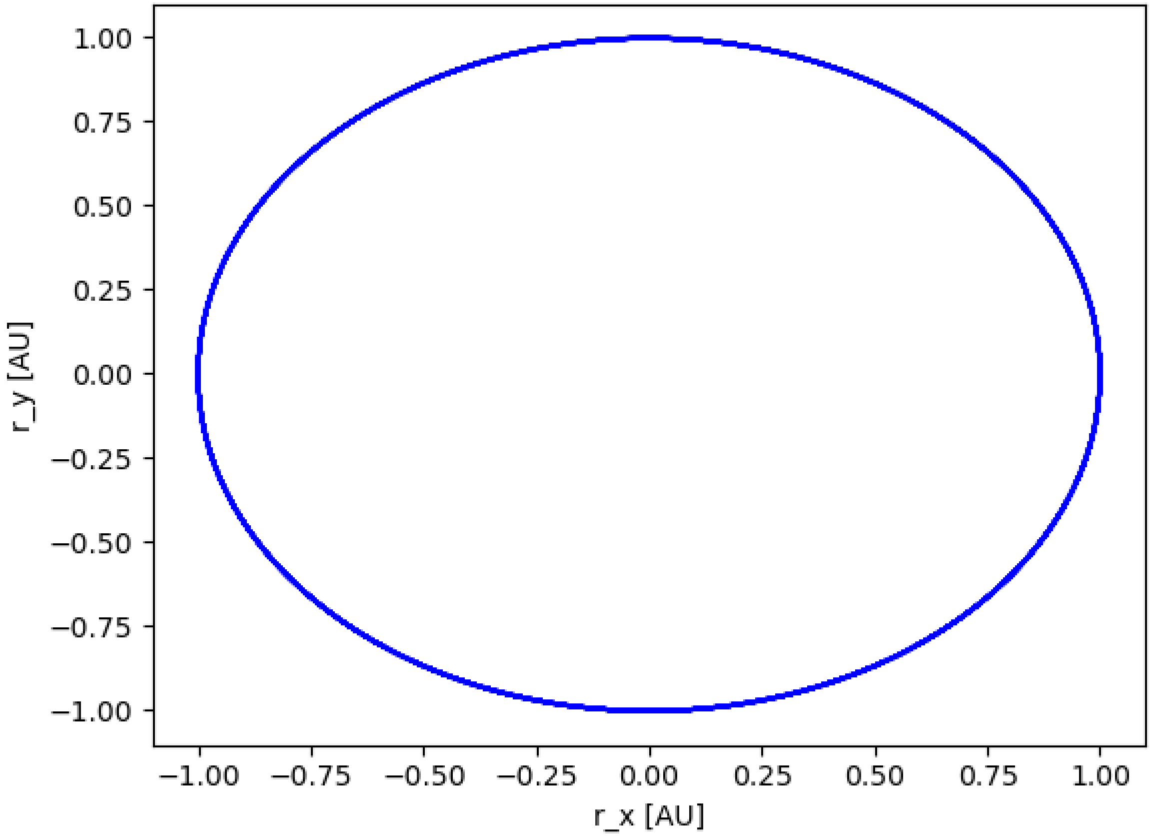
$$\mathbf{v}_{i+1} = \mathbf{v}_i + \mathbf{a}_i\,\delta\tau$$

e

$$\mathbf{r}_{i+1} = \mathbf{r}_i + \mathbf{v}_{i+1}\,\delta\tau$$

Note a utilização de $\mathbf{v}_{i+1}$ no cáculo da posição (no método de Euler usa-se $\mathbf{v}_i$).

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt

t0 = 0.0                    # condição inicial, tempo [ano]
tf = 10.0                   # limite do domínio, tempo final [ano]
dt = 0.001                  # passo [ano]
v0 = 2.0 * np.pi            # condição inicial, módulo da velocidade inicial [AU/ano]
x0 = 1.0                    # condição inicial, coordenada x da posição inicial [AU]

G = 4.0 * np.pi ** 2        # constante de gravitação [AU^3 / M ano^2]

# inicializar domínio [ano]
t = np.arange(t0, tf, dt)

# inicializar solução, aceleração a 2D [AU / ano^2]
a = np.zeros([np.size(t),2 ])

# inicializar solução, velocidade [m / s]
v = np.zeros([np.size(t), 2])
v[0, :] = np.array([0, v0])  # velocidade [AU / ano] para t = 0 ano

# inicializar solução, posição [m]
r = np.zeros([np.size(t), 2])
r[0, :] = np.array([x0, 0.0])    # posição [AU] para t = 0 ano

for i in range(np.size(t) - 1):
    a[i, :] = - G * r[i, :] / np.linalg.norm(r[i, :]) ** 3   # aceleração instantânea
    v[i + 1, :] = v[i, :] + a[i, :] * dt
    r[i + 1, :] = r[i, :] + v[i + 1, :] * dt   # Método de Euler-Cromer

#plt.figure(figsize=(10,10))
#plt.xlim(-0.5, 1.5)
#plt.ylim(-0.5, 0.5)
plt.plot(r[:, 0], r[:, 1], 'b-')
plt.xlabel("r_x [AU]")
plt.ylabel("r_y [AU]")
plt.show()
```



Consegue órbitas fechadas?

Sim. As orbitas são fechadas.

Experimente com diferentes magnitudes da velocidade inicial (entre π e 3π, aproximademente). As órbitas são elipses (concordam com a primeira lei de Kepler)?

Sim.

Pergunta 2:

Se a posição inicial da planeta for (0.5,0), a velocidade inicial deve ser maior ou menor do que $v_0 = 2\pi$ UA, para conseguir uma órbita circular?

Exercício 3

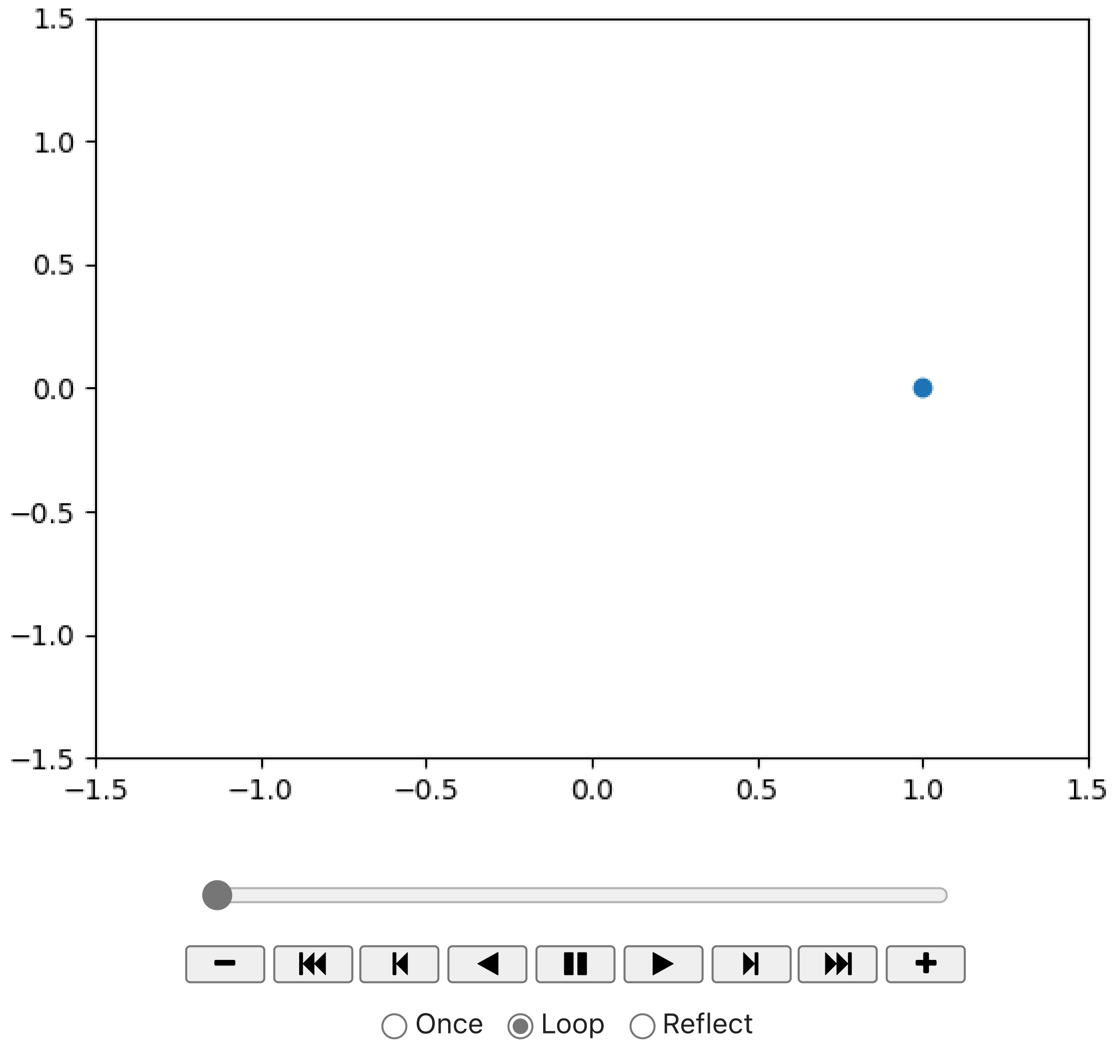
Usando os resultados da simulação, cria uma animação do movimento da Terra à volta do sol durante um ano.

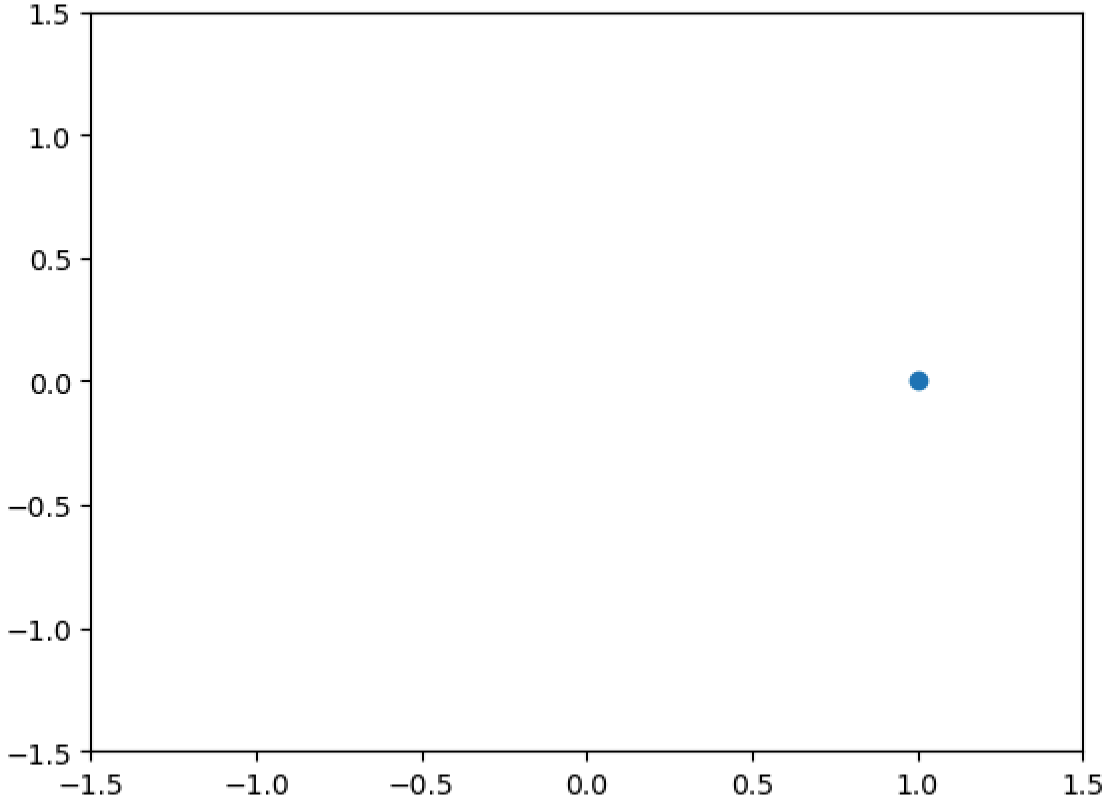
Pode adaptar o seguinte código:

In [3]:
```
from matplotlib.animation import FuncAnimation
from IPython.display import HTML


fig = plt.figure()
ax = plt.axes(xlim=(-1.5, 1.5), ylim=(-1.5, 1.5))
```

```python
terra = ax.plot([], [], 'o')[0]          # terra, posição inicial

def update(frame):
    # atualizar o plot da posição da Terra
    terra.set_xdata([r[frame, 0]])
    terra.set_ydata([r[frame, 1]])
    return terra

nframes = 1000
total_frames = np.size(t)
iframes = np.arange(0, total_frames, total_frames // nframes)
ani = FuncAnimation(fig=fig, func=update, frames=iframes, interval=100)

HTML(ani.to_jshtml())
```

Out[3]:




1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5