Laboratorio 5

June 11, 2020

[2]:
```
#%matplotlib notebook
%matplotlib inline
```

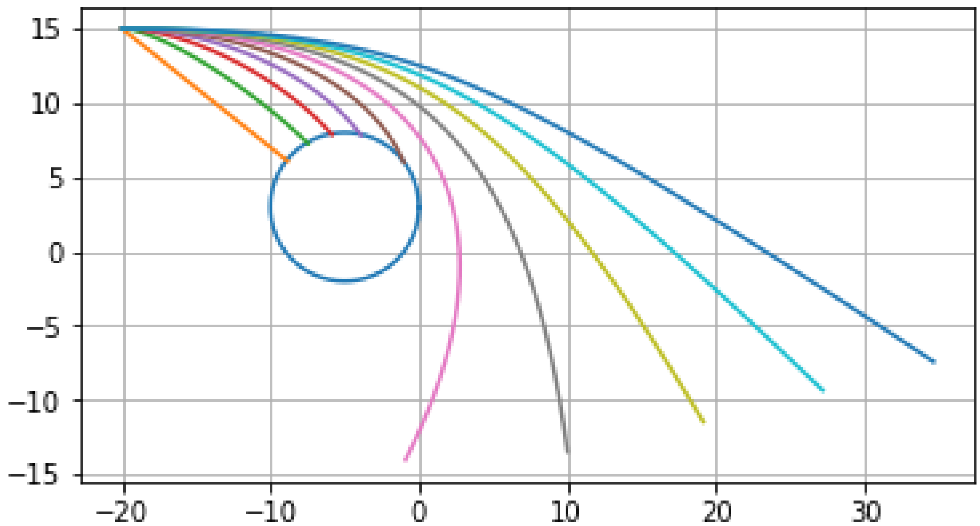
1 Movimiento Gravitatorio

[1]:
```
import numpy as np
from matplotlib import pyplot as plt
```

1.1 Con dos cuerpo

[13]:
```
#circulo

c0 = np.array([ -5 , 3 ])
r = 5
tf = 110
h = 0.1



angle = np.linspace(0, 2*np.pi, tf+1)
fig = plt.figure()
plt.plot(r * np.cos(angle) + c0[0], r * np.sin(angle) + c0[1])

for vx0 in np.arange(0, 0.6, 0.06):
    p0 = np.array([ -20 , 15 ])
    v0 = np.array([ vx0 , 0 ])
    a0 = np.array([ 0 , 0 ])
    xs = [ p0[0] ]
    ys = [ p0[1] ]
    for i in np.arange(0 , tf, h):
        r_c_2 = (p0[0] - c0[0])**2 + (p0[1] - c0[1])**2
        a0 = np.array([-(p0[0] - c0[0])/( (r_c_2)**(1.5) ), -(p0[1] - c0[1])/(␣
 ↪(r_c_2)**(1.5) )  ])
        v0 = v0 + a0*h
        p0 = p0 + v0*h
        xs.append(p0[0])
        ys.append(p0[1])
```

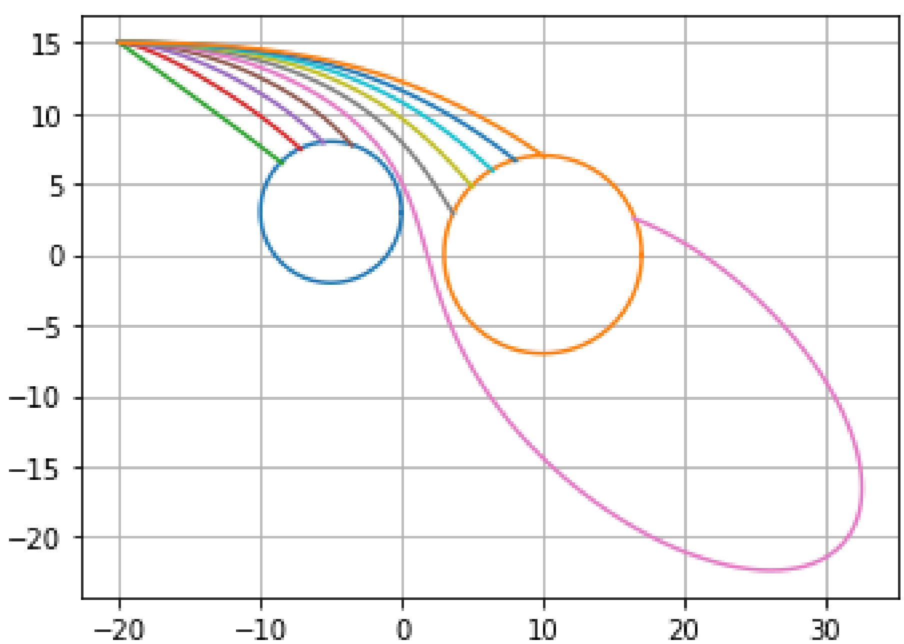
```
        if (r_c_2 < r**2):
            break
    plt.plot(xs, ys)



plt.gca().set_aspect('equal')

plt.grid()
```



1.2 Con tres cuerpos

[48]:
```
tf = 400
h = 0.1

c0 = np.array([ -5 , 3 ])
r0 = 5
angle0 = np.linspace(0, 2*np.pi, tf+1)
fig = plt.figure()
plt.plot(r0 * np.cos(angle0) + c0[0], r0 * np.sin(angle0) + c0[1])


c1 = np.array([ 10 , 0 ])
r1 = 7
angle1 = np.linspace(0, 2*np.pi, tf+1)
plt.plot(r1 * np.cos(angle1) + c1[0], r1 * np.sin(angle0) + c1[1])
```

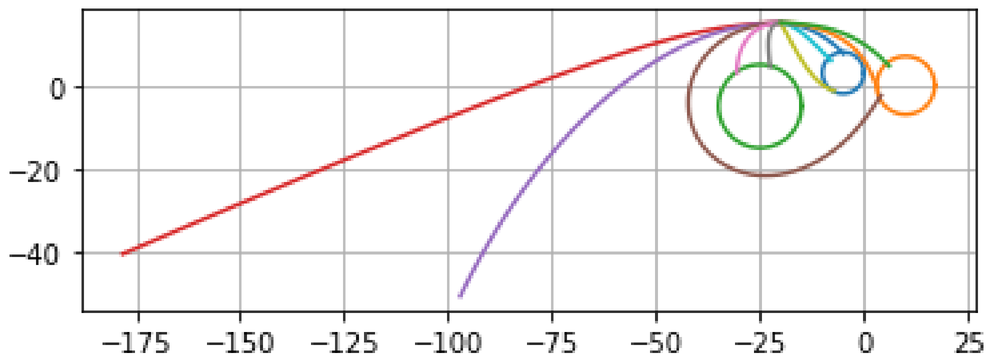
```
for vx0 in np.arange(0, 0.6, 0.06):
    p0 = np.array([ -20 , 15 ])
    v0 = np.array([ vx0 , 0 ])
    a0 = np.array([ 0 , 0 ])
    xs = [ p0[0] ]
    ys = [ p0[1] ]
    for i in np.arange(0 , tf, h):
        r_c_2 = (p0[0] - c0[0])**2 + (p0[1] - c0[1])**2
        r_c_3 = (p0[0] - c1[0])**2 + (p0[1] - c1[1])**2
        ax = -(p0[0] - c0[0])/( (r_c_2)**(1.5) ) -(p0[0] - c1[0])/( (r_c_3)**(1.
 ↪5) )
        ay = -(p0[1] - c0[1])/( (r_c_2)**(1.5) ) -(p0[1] - c1[1])/( (r_c_3)**(1.
 ↪5) )
        a0 = np.array([ ax , ay  ])
        v0 = v0 + a0*h
        p0 = p0 + v0*h
        xs.append(p0[0])
        ys.append(p0[1])
        if (r_c_2 < r0**2 or r_c_3 < r1**2):
            break
    plt.plot(xs, ys)



plt.gca().set_aspect('equal')

plt.grid()
```



1.3 Con cuatro cuerpos

[32]:
```
tf = 400
h = 0.1

c0 = np.array([ -5 , 3 ])
r0 = 5
angle0 = np.linspace(0, 2*np.pi, tf+1)
fig = plt.figure()
plt.plot(r0 * np.cos(angle0) + c0[0], r0 * np.sin(angle0) + c0[1])


c1 = np.array([ 10 , 0 ])
r1 = 7
angle1 = np.linspace(0, 2*np.pi, tf+1)
plt.plot(r1 * np.cos(angle1) + c1[0], r1 * np.sin(angle0) + c1[1])



c2 = np.array([ -25 , -5 ])
r2 = 10
angle1 = np.linspace(0, 2*np.pi, tf+1)
plt.plot(r2 * np.cos(angle1) + c2[0], r2 * np.sin(angle0) + c2[1])

for vx0 in np.arange(-0.6, 0.6, 0.12):
    p0 = np.array([ -20 , 15 ])
    v0 = np.array([ vx0 , 0 ])
    a0 = np.array([ 0 , 0 ])
    xs = [ p0[0] ]
    ys = [ p0[1] ]
    for i in np.arange(0 , tf, h):
        r_c_2 = (p0[0] - c0[0])**2 + (p0[1] - c0[1])**2
        r_c_3 = (p0[0] - c1[0])**2 + (p0[1] - c1[1])**2
        r_c_1 = (p0[0] - c2[0])**2 + (p0[1] - c2[1])**2
        ax = -(p0[0] - c0[0])/( (r_c_2)**(1.5) ) -(p0[0] - c1[0])/( (r_c_3)**(1.
↪5) ) -(p0[0] - c2[0])/( (r_c_1)**(1.5) )
        ay = -(p0[1] - c0[1])/( (r_c_2)**(1.5) ) -(p0[1] - c1[1])/( (r_c_3)**(1.
↪5) ) -(p0[1] - c2[1])/( (r_c_1)**(1.5) )
        a0 = np.array([ ax , ay  ])
        v0 = v0 + a0*h
        p0 = p0 + v0*h
        xs.append(p0[0])
        ys.append(p0[1])
        if (r_c_2 < r0**2 or r_c_3 < r1**2 or r_c_1 < r2**2):
```

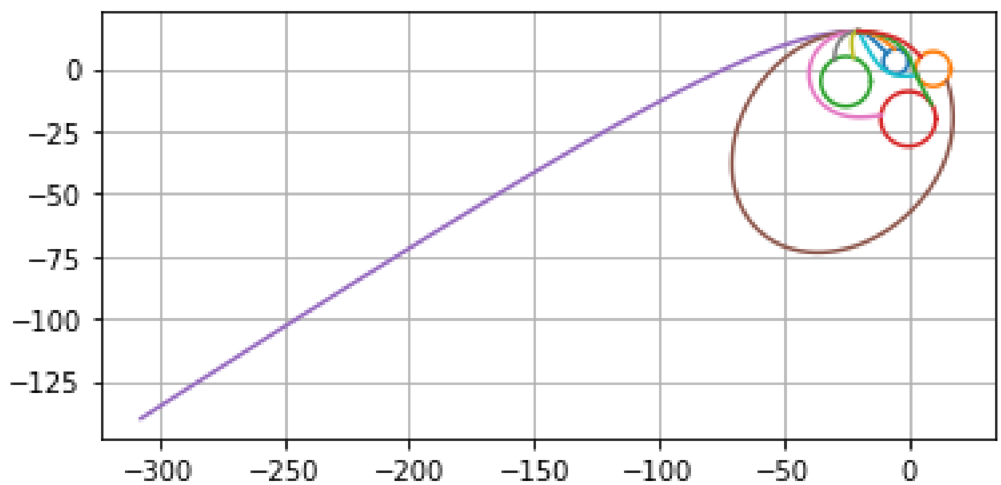
```
            break
    plt.plot(xs, ys)



plt.gca().set_aspect('equal')

plt.grid()
```



1.4 Con n cuerpos

[44]:
```
tf = 1000
h = 0.1
fig = plt.figure()
cs = [
    [ np.array([ -5 ,  3 ]),  5 ],
    [ np.array([ 10 , 0 ]),  7 ],
    [ np.array([ -25 , -5 ]), 10 ],
    [ np.array([ 0 , -20 ]), 11 ],
]
angles = np.linspace(0, 2*np.pi, tf+1)

for c in cs:
    plt.plot(c[1] * np.cos(angles) + c[0][0], c[1] * np.sin(angles) + c[0][1])



for vx0 in np.arange(-0.6, 0.6, 0.12):
    p0 = np.array([ -20 , 15 ])
    v0 = np.array([ vx0 , 0 ])
    a0 = np.array([ 0 , 0 ])
    xs = [ p0[0] ]
    ys = [ p0[1] ]
    continues = True
```

```
    for i in np.arange(0 , tf, h):
        if not continues:
            True
        a0 = np.array([ 0, 0  ])
        for c in cs:
            r2 = (p0[0] - c[0][0])**2 +(p0[1] - c[0][1])**2
            r = c[1]
            a0 = a0 - np.array([  (p0[0] - c[0][0])/( (r2)**(1.5) ), (p0[1] -␣
↪c[0][1])/( (r2)**(1.5) )  ])
            continues = continues and r2 > r**2

        v0 = v0 + a0*h
        p0 = p0 + v0*h
        xs.append(p0[0])
        ys.append(p0[1])
        if not continues:
            break

    plt.plot(xs, ys)
plt.gca().set_aspect('equal')
plt.grid()
```



[ ]: