

3_projectile-plot

May 28, 2023

```
[ ]: #!/usr/bin/env python3

import numpy as np
import matplotlib.pyplot as plt

# Plot the trajectory of a projectile.

# Time mesh.
tmin = 0.0 # s
tmax = 10.0 # s
tt = np.linspace(tmin, tmax, 101)

# Coordinates as a function of time, initial velocity and angle.
def coordinates(t, v, a):
    g = 9.8 # m/s^2
    x = v * np.cos(a) * t
    y = v * np.sin(a) * t - 0.5 * g * (t**2)
    return x, y

# Calculate the coordinates for specific parameter values.
v = 9.0 # m/s
a_1 = 0.2 * np.pi
xx_1, yy_1 = coordinates(tt, v, a_1)

a_max = 0.25 * np.pi
xx_max, yy_max = coordinates(tt, v, a_max)

a_2 = 0.4 * np.pi
xx_2, yy_2 = coordinates(tt, v, a_2)

# Plot the trajectories.
plt.figure(figsize=(3.5,3.5), frameon=True) #figsize: misure dell'immagine in
↳indici
plt.axes([0.2,0.2,0.7,0.7]) #left, bottom, width, height
plt.plot(xx_1, yy_1, "-b")
plt.plot(xx_max, yy_max, "-r")
plt.plot(xx_2, yy_2, "-b")
```

```

plt.xlim([0.0, 10.0])
plt.ylim([0.0, 10.0])
plt.xlabel("x [m]")
plt.ylabel("y [m]")
plt.savefig("trajectory1.png", dpi=300)

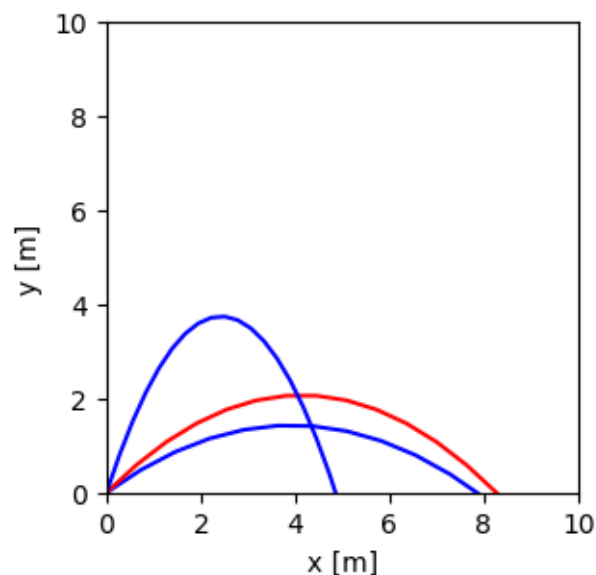
# Plot a whole fan of trajectories.

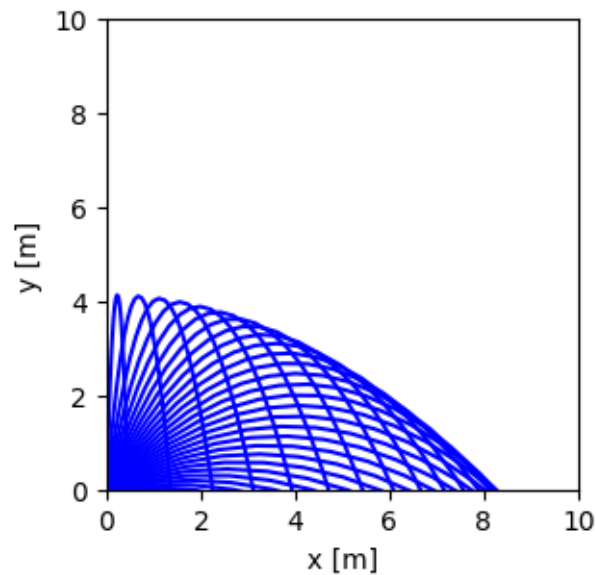
# List of values of the angle.
a_fan = np.linspace(0.0, np.pi/1.9, 31)

# Iterate over the list of angles and calculate the corresponding coordinates.
xx_fan = [] # define an empty list to store the coordinates.
yy_fan = []
for a in a_fan: # iterate over the angles
    xx, yy = coordinates(tt, v, a) # calculate for the specific angle.
    xx_fan.append(xx) # save the coordinates to a new slot in the lists
    yy_fan.append(yy)

# Plot the trajectory.
plt.figure(figsize=(3.5,3.5), frameon=True)
plt.axes([0.2,0.2,0.7,0.7])
for xx, yy in zip(xx_fan, yy_fan): # iterate over the saved coordinates.
    plt.plot(xx, yy, "-b")
plt.xlim([0.0, 10.0])
plt.ylim([0.0, 10.0])
plt.xlabel("x [m]")
plt.ylabel("y [m]")
plt.savefig("trajectory2.png", dpi=300)

```





RIASSUNTO:

in questa lezione si è disegnata la traiettoria di un moto parabolico con parametri iniziali ben definiti e un fascio di parabole con parametri variabili. Per il primo caso, sono importanti i seguenti concetti:

1. `def` coordinate, `return` coordinate;
2. nel `plot`:
 - `figsize`: esprime le dimensioni dell'immagine in pollici,
 - `plt.axes`: esprime le dimensioni dei parametri dell'immagine.
3. “`zip`”: scorre le liste contenute nella funzione insieme

Per il secondo caso, ovvero il fascio di parabole, è importantissimo il “ciclo `for`”:

for “variabile” in “lista”:

i puntini definiscono azioni elementari, come l'aggiunta di valori a liste istituite (tramite la funzione “`lista.append()`”)