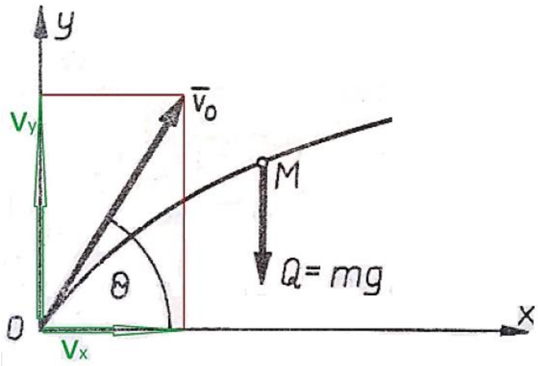
Flight dynamics of missiles and bombs: *Part 1 of 4*

A fundamental issue in rocket technology: the flight of a projectile in a vacuum.

Technology used: Python 3.10 | PyCharm | Linux – Ubuntu | Flask

Part 1: Motion of the center of mass in a vacuum

A projectile moving through air is under the action of two forces: gravity and drag.



At high speeds, the drag force is many times greater than the force of gravity. However, in the case of a large caliber projectile moving slowly (or flying at high altitude flight), the drag may be equal to or even less than it.

In the motion under consideration, the drag force in a vacuum is equal to zero. Moreover, we assume motion in a vacuum in a uniform gravitational field, and locally in this case the earth is flat. Therefore we consider any point of the projectile path M.

If we fire a projectile at an angle $\theta = \theta_0$ in relation to the ground and with an initial velocity v_0 the initial velocity components are $v_x = v_0 cos\theta_0$ and $v_y = v_0 sin\theta_0$.

There is only one force acting on a projectile moving in a vacuum: gravity: $Q = mg$. Arranging the equations of motion:

$$\sum_{i=1}^{n} F_{i\_x} = m \cdot \ddot{x}, \sum_{i=1}^{n} F_{i\_y} = m \cdot \ddot{y}, \sum_{i=1}^{n} F_{i\_z} = m \cdot \ddot{z}^{\,3}$$

Hence because there is no force acting on the x-axis: $m \cdot \ddot{x} = 0$

The y-axis is affected by the force of gravity: $m \cdot \ddot{y} = -mg$

When divided by the mass we get : $\ddot{y} = -g$

At the same time it should be noted that the projectile takes off at an angle of θ_0, so the velocity is already decomposed into components v_x, v_y. Therefore, entering the conditions at the beginning of the launch (initial), $t = 0$, $x = 0, y = 0$ – the projectile has not yet taken off (it has no time of flight, and no distance traveled) we have:

$$\dot{x} = v_0 cos\theta_0 \ , \dot{y} = v_0 sin\theta_0$$

Since neither into the equations $\ddot{x} = 0$, $\ddot{y} = -g$ nor to the initial conditions does mass enter, therefore we have a conclusion:

The motion of a projectile's centre of mass in a vacuum does not depend on its gravity.

Integrating:

$$\int \ddot{x}\, dt = \int 0\, dt \quad \rightarrow \quad \dot{x} = 1 \cdot c_1$$

$$\int \ddot{y}\, dt = -\int g\, dt \quad \rightarrow \quad \dot{y} = -g \cdot t + c_2$$

By substituting the initial conditions we have to the equations of motion:

$$\dot{x} = 1 \cdot c_1 \rightarrow v_0 cos\theta_0 = 1 \cdot c_1 \rightarrow c_1 = v_0 cos\theta_0$$

$$\dot{y} = -g \cdot t + c_2 \rightarrow v_0 sin\theta_0 = -g \cdot t + c_2 \rightarrow but\ -g \cdot t\ because\ t = 0 \rightarrow c_2 = v_0 sin\theta_0$$

therefore the equations of motion with initial conditions

$$\boldsymbol{\dot{x} = v_0 cos\theta_0,\ \dot{y} = -g \cdot t + v_0 sin\theta_0}$$

From equation $\dot{x} = v_0 cos\theta_0$ it follows that horizontal motion in a vacuum remains constant along its entire path.

To determine the path, we further integrate equations:

$$\int \dot{x}\, dt = \int (v_0 \cdot cos\theta_0) dt \qquad \rightarrow \boldsymbol{x = v_0 \cdot t \cdot cos\theta_0 + c_3}$$

$$\int \dot{y}\, dt = \int (-g \cdot t + v_0 \cdot sin\theta_0)\, dt \qquad \rightarrow \boldsymbol{y = -\frac{g \cdot t^2}{2} + v_0 \cdot t \cdot sin\theta_0 + c_4}$$

Based on initial conditions: $t = 0$ it follows that:

$x = v_0 \cdot t \cdot cos\theta_0 + c_3 = 0$ similarly $y = -\frac{gt^2}{2} + v_0 \cdot t \cdot sin\theta_0 + c_4 = 0$ i.e. $c_3 = 0$ and $c_4 = 0 \rightarrow c_3 = c_4 = 0$, so:

$$\boldsymbol{x = v_0 \cdot t \cdot cos\theta_0,\ \dot{y} = -g \cdot t + v_0 sin\theta_0}$$

These equations represent what position along the coordinate axis a projectile fired in a vacuum with an initial velocity v_0 at an angle of θ_0 to the earth's surface after a time t. If:

$$x = v_0 \cdot t \cdot cos\theta_0 \quad \rightarrow \quad t = \frac{x}{v_0 \cdot cos\theta_0}$$

and insert it into:

$$y = -\frac{g \cdot t^2}{2} + v_0 \cdot t \cdot sin\theta_0$$

then we obtain the **<u>equation of trajectory</u>**:

$$y = -g \cdot \frac{x^2}{2 \cdot (v_0 \cdot cos\theta_0)^2} + v_0 \cdot \frac{x}{v_0 \cdot cos\theta_0} \cdot sin\theta_0 = -\frac{g \cdot x^2}{2 \cdot {v_0}^2 \cdot cos^2\theta_0} + xtg\theta_0$$

```
import math

import matplotlib.pyplot as plt


G = 9.81


class Missile:


    def __init__(self, firing_angle, initial_velocity):

        self.firing_angle = math.radians(firing_angle)

        self.initial_velocity = initial_velocity

        self.tangens_teta0 = math.tan(self.firing_angle)

        self.sin_2_teta = round((math.pow(self.initial_velocity, 2)) * math.sin(2 *
self.firing_angle) / G, 2)

        self.a = (G / (2 * math.pow(self.initial_velocity, 2)))




    def equation_of_trajectory(self):

        self.tab_y = []

        for x in range(0, int(self.sin_2_teta) + 1):

            y = round(x * self.tangens_teta0 - math.pow(x, 2) * self.a *
(math.pow(self.tangens_teta0, 2) + 1), 2)

            self.tab_y.append(y)

            x += 1

        return self.tab_y
```

(Code part 2, 3, 4)

```
    def call_block(self):

        a = self.equation_of_trajectory()

        b = self.flight_path_envelope_for_constant_speed()

        c = self.axis_of_symmetry()
```

```
        d = self.velocity_of_flight()
        try:
            e = self.elements_of_path()
        except AttributeError:
            print(f'No data available for this firing angle')
            e = None
        plt.plot(a)
        plt.plot(b)
        plt.plot([c, c], [0, max(a)])
        plt.plot(d)
        plt.show()
        return e

    def __str__(self):
            return f'{self.x_start}, {self.x_end}, {self.time_of_flight}, {self.top_of_trajectory}, \
              {self.time_of_flightTOP}, {self.average_flight_altitude}, {self.average_time},
{self.x_end_angle}'



if __name__ == '__main__':
    Missile()
```

This is part 1 of a publication of 4.

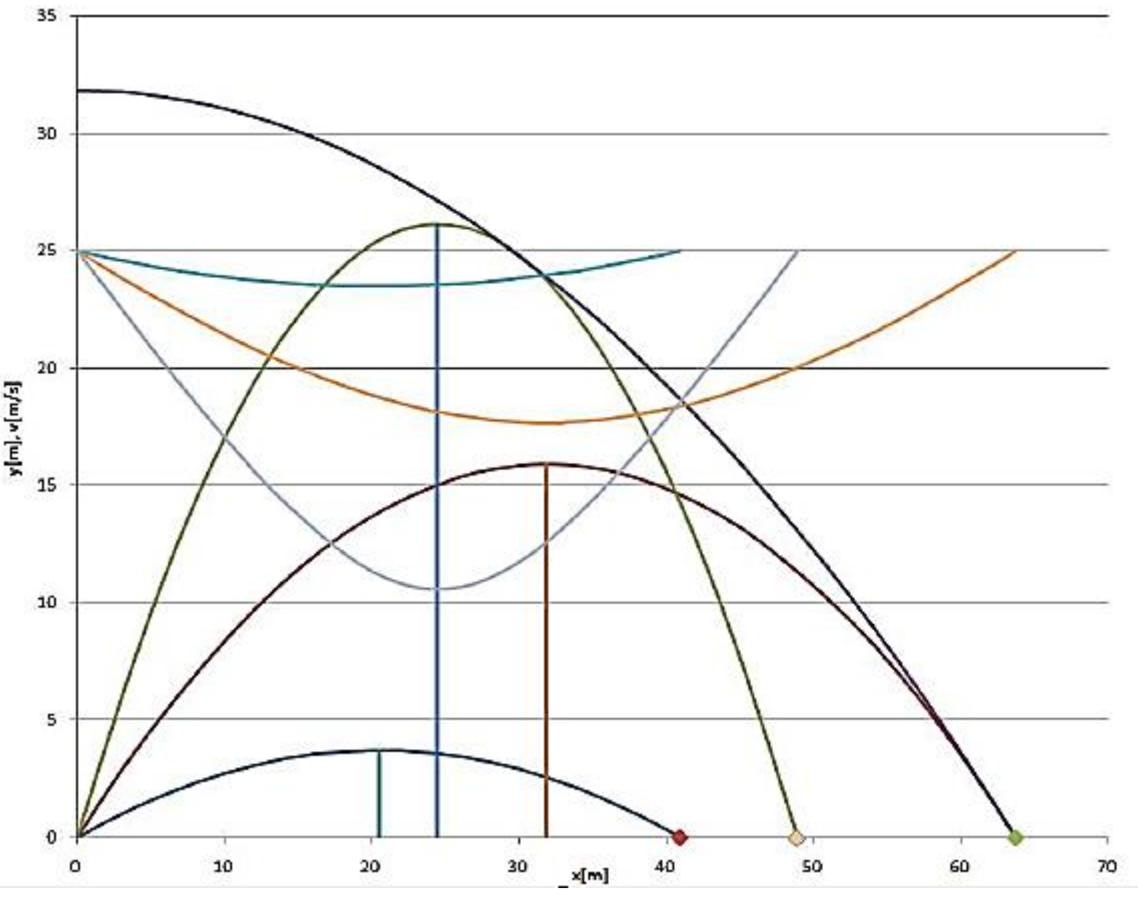
<u>In preparation:</u>

part 2 : Study of the projectile path equation;

part 3 : Missile Range Curve

part 4 : Appendix

In the last part we will present all the code to obtain the following graph:


y[m], v[m/s]
x[m]