

Cinemática y Python

FISICA COMPUTACIONAL

MARTIN CANDIA

Introducción

En el presente trabajo se desarrollará sobre Cinemática, el funcionamiento de esta rama de la física y como se puede implementar la informática para hacer un mejor análisis y expandir las herramientas disponibles para trabajar este tema. En el presenta caso se utilizará Python, se busca brindar graficas con una aproximación del recorrido que tendría el cuerpo a analizar y desarrollar sus posibles situaciones en momentos específicos de forma analítica al trabajar con los valores a preferencia del usuario.

Marco Teórico

La Cinemática se encarga de describir el movimiento de los cuerpos en función del tiempo. Para esto, sin analizar las causas de este, solo con sus condiciones iniciales se limita a estudiar la trayectoria en función del tiempo.

Para aplicar esta rama de la física dentro de la mecánica clásica, se toma la existencia de un espacio y tiempo absoluto, es decir, un espacio independiente de todos los objetos materiales y un tiempo que transcurre del mismo modo en todo el universo y es independiente de la existencia de fenómenos físicos.

En este caso se trabajará con el Movimiento Rectilíneo Uniforme (posteriormente llamado MRU), Movimiento Rectilíneo Uniformemente Variado o Acelerado (llamado MRUV) y Tiro Oblicuo. En el MRU, la aceleración es nula, por lo que la velocidad permanece constante, se va a analizar en una única dimensión. En el MRUV, se presenta una aceleración que varía a la velocidad, también será analizada en una única dimensión. En el Tiro Oblicuo, se presentan aceleraciones, se analizará en 2 dimensiones.

Desarrollo

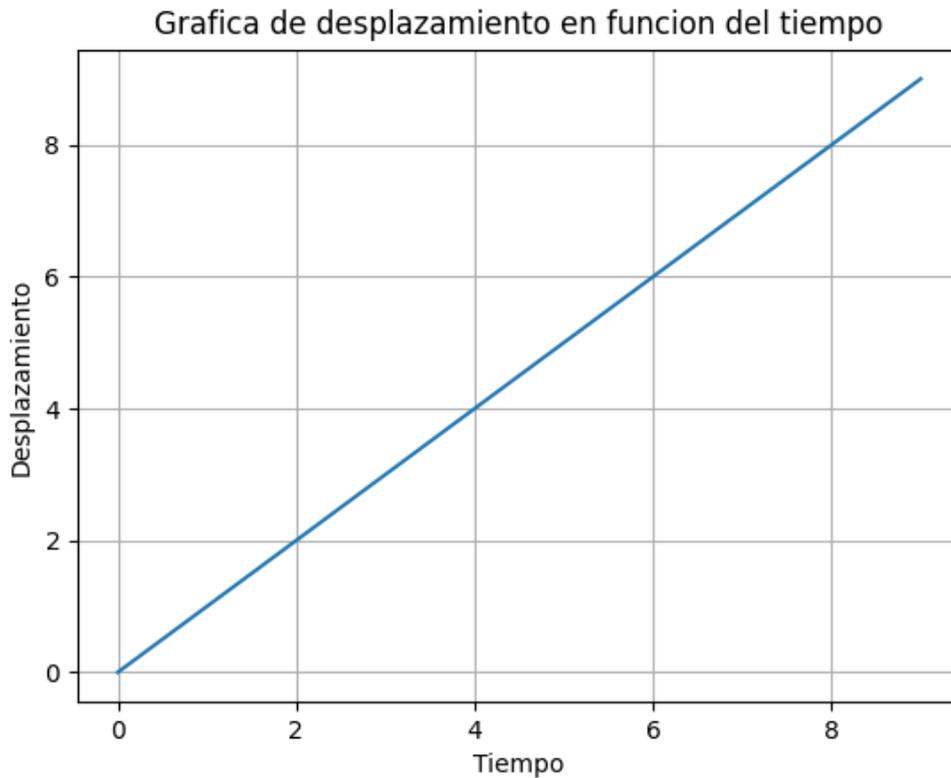
Para trabajarlo, se utilizará un sistema ideal sin tener en cuenta la fricción con el aire, rozamientos, entre otras variables que afectarían a la trayectoria del objeto. Además, se analizará a este como una partícula.

Comenzando con el MRU, este sistema depende de una función, que dependen de la velocidad para poder desarrollar el movimiento. La velocidad, siendo la razón de cambio en el tiempo.

$$\Delta x = V\Delta t$$

Siendo x la posición en esta única dimensión, V la velocidad inicial que va a ser constante y t el tiempo transcurrido.

Aplicando estos conceptos, se puede hacer un programa que permita calcular cualquier variable teniendo las otras dos, para este se estableció el punto central del sistema de referencias como el punto inicial. Además, permite realizar un gráfico donde se puede ver el desplazamiento del cuerpo en función del tiempo. Ver Anexo 1 para obtener el código.



*Gráfico MRU. Gráfico realizado con una velocidad de 1 unidad de distancia/unidad de tiempo.
Ver Anexo 1 para obtener el código.*

En MRUV, hay que agregar una aceleración, que causa un cambio en la velocidad, a esta para las fórmulas la representaremos como "a". A su vez, se suma una nueva posible formula, ya que ahora podemos calcular la velocidad en un punto, debido a que ahora está varia:

$$\Delta x = Vi\Delta t + \frac{a\Delta t^2}{2}$$

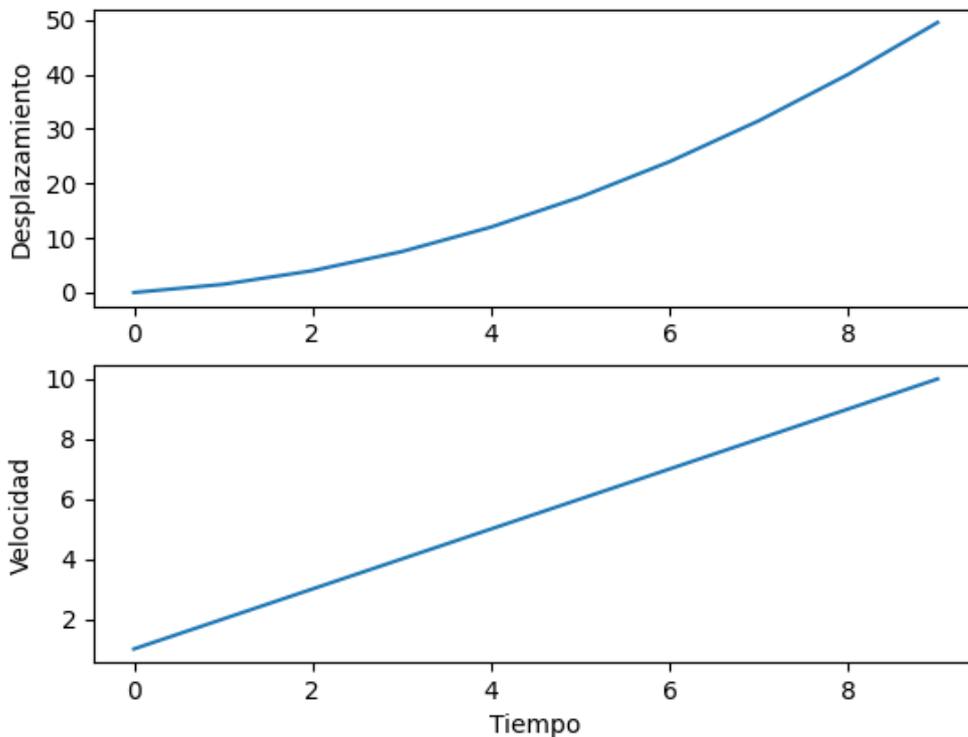
$$Vf = Vi + a\Delta t$$

Además, hay que tener en cuenta que como la velocidad varia tenemos que definir y separar la terminología para esta, "Vi" representa a la velocidad inicial y "Vf" a la velocidad en el momento a calcular. También se limitaron las posibilidades a calcular, debido a que por la cantidad de variables que hay, existen demasiadas formas de calcular cada una de ellas utilizando las demás. Se simplificarán los cálculos posibles del punto anterior, aumentando los gráficos, que es donde mas se busca centrar debido a la velocidad y practicidad que presenta esta metodología.

Con todo esto definido, la gráfica a continuación muestra las funciones desplazamiento y velocidad en función del tiempo, con una aceleración de 1 y una velocidad inicial de 1. El código capaz de hacer los cálculos antes mencionados y la grafica se encuentran en el Anexo 2.

A tener en cuenta que esto mismo se podría utilizar para trabajar Tiro Vertical en una sola dimensión, debido a que la forma de trabajo analítica y grafica son similares, solo hay que tener en cuenta la gravedad como aceleración.

Analisis del cuerpo en MRUV



*Gráfico MRUV. Gráfico realizado con una velocidad y aceleración de 1 unidad de distancia/unidad de tiempo o unidad de tiempo al cuadrado respectivamente.
Ver Anexo 2 para obtener el código.*

Como ultimo modelo, se presentará el área de Tiro Oblicuo.

Para esta área, hay que tener en cuenta que la cantidad de variables respecto a la anterior es casi el doble, ya que hay 2 dimensiones en las cuales trabajar por separado, por lo que, si bien las formulas a utilizar se mantienen, se tienen en cuenta las variables de ambas dimensiones por separado.

Aun así, este sistema no tiene otra variación respecto al anterior, por lo que, si bien el análisis mental es mas complejo ya que hay que tener en cuenta mas posibilidades, el trabajo analítico es similar. De todas formas, justamente para evitar el trabajo mental imaginando la trayectoria, es donde se implementa Python como herramienta para graficar como se muestra a continuación.

Trayectoria 1:

Velocidad inicial en X = 3 m/s
Velocidad Inicial en Y = 30 m/s
Aceleración en X = 1 m/s²
Aceleración en Y = -9.81 m/s²

Analisis del cuerpo en Tiro Oblicuo

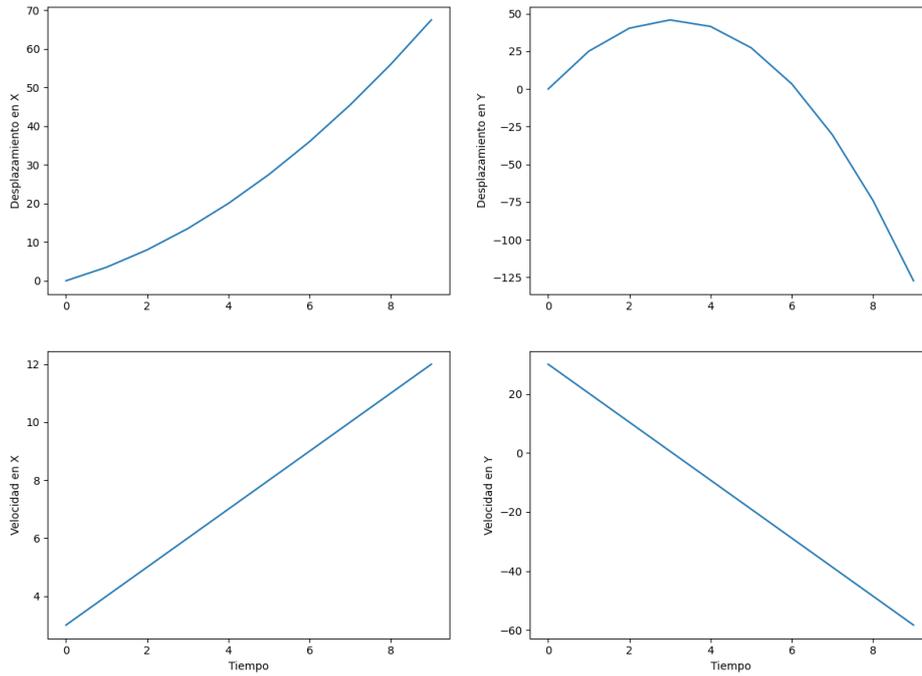


Gráfico Características de Tiro Oblicuo 1.
Ver Anexo 3 para obtener el código.

Trayectoria del cuerpo en Tiro Oblicuo

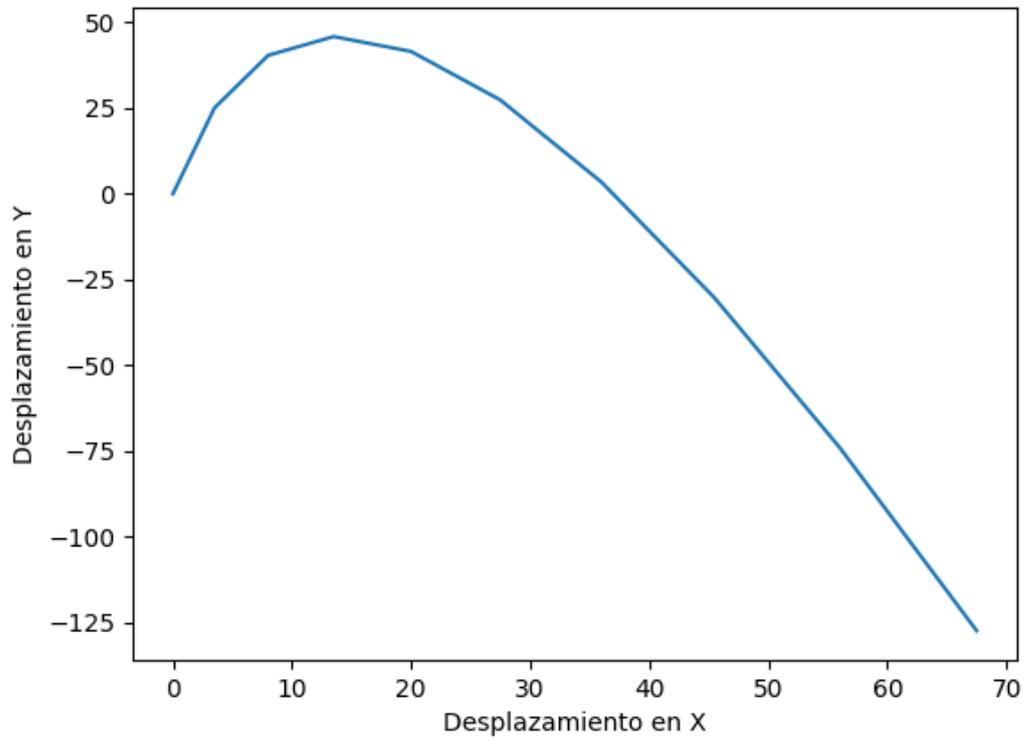


Gráfico Trayectoria de Tiro Oblicuo 1.
Ver Anexo 3 para obtener el código.

Trayectoria 2:

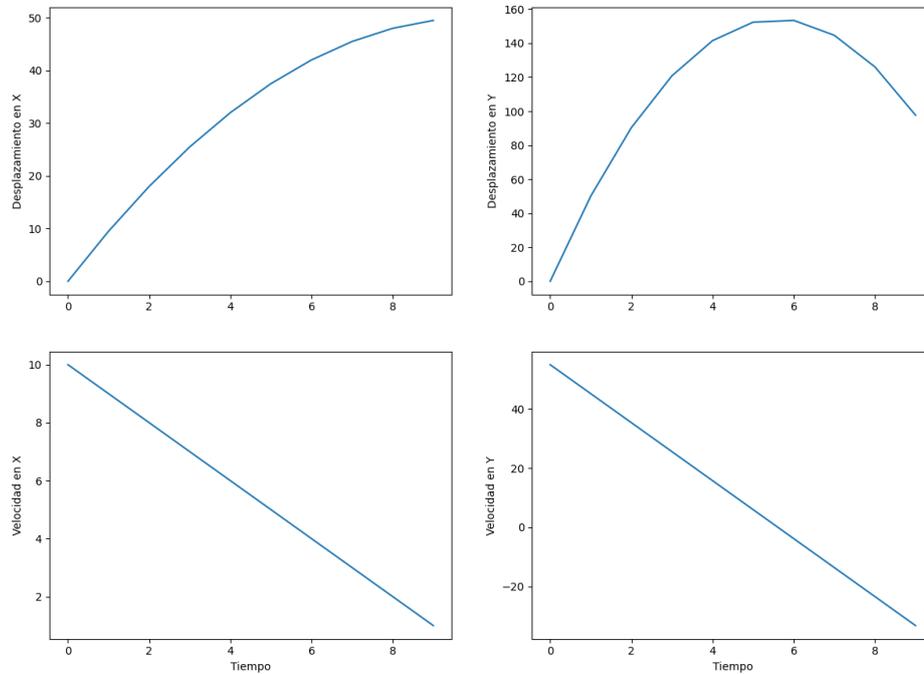
Velocidad inicial en X = 10 m/s

Velocidad Inicial en Y = 55 m/s

Aceleración en X = -1 m/s²

Aceleración en Y = -9.81 m/s²

Analisis del cuerpo en Tiro Oblicuo



*Gráfico Características de Tiro Oblicuo 2.
Ver Anexo 3 para el código.*

Trayectoria del cuerpo en Tiro Oblicuo

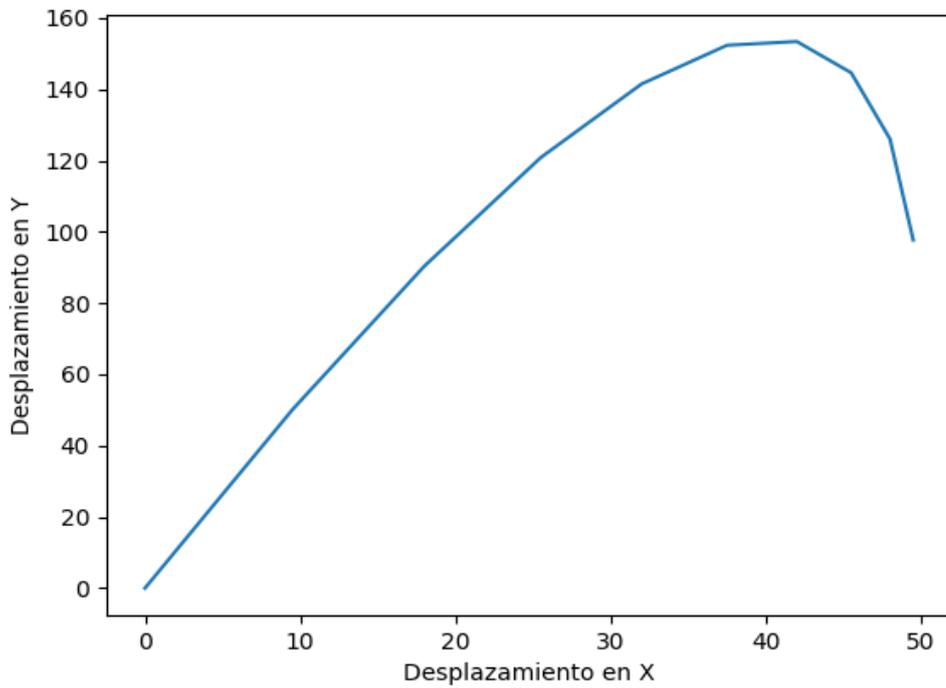


Gráfico Trayectoria de Tiro Oblicuo 2.
Ver Anexo 3 para obtener el código.

Trayectoria 3:

- Velocidad inicial en X = 1 m/s
- Velocidad Inicial en Y = 0 m/s
- Aceleración en X = 0 m/s²
- Aceleración en Y = -9.81 m/s²

Análisis del cuerpo en Tiro Oblicuo

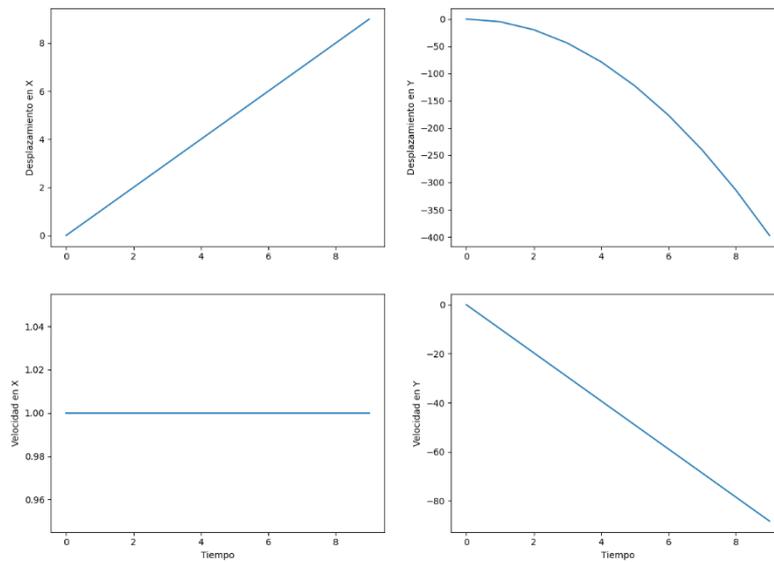
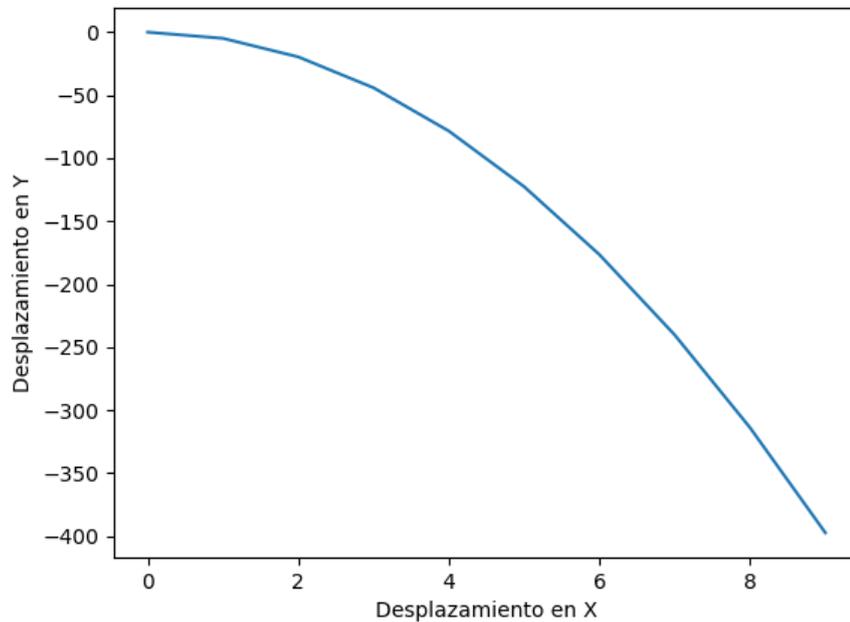


Gráfico Características de Tiro Oblicuo 3.
Ver Anexo 3 para el código.

Trayectoria del cuerpo en Tiro Oblicuo



*Gráfico Trayectoria de Tiro Oblicuo 2.
Ver Anexo 3 para obtener el código.*

Conclusión

Como se puede observar, la cinemática permite analizar de forma simplificada la trayectoria de los cuerpos en ambientes ideales. Aun así, hay muchos factores que no se tienen en cuenta para esta, como los rozamientos contra las superficies o el aire, como el viento, etc. Por lo que puede llegar a ser una buena herramienta teórica pero imposible de aplicar a la práctica debido que requiere condiciones imposibles de conseguir en la cotidianidad.

Anexos

Anexo 1, programa MRU

```
import matplotlib.pyplot as plt

import numpy as np

import sys

guia = 0

while guia == 0:

    print("Ingrese 1 para calcular la posicion del cuerpo")

    print("Ingrese 2 para calcular el tiempo al que llego a esa posicion")

    print("Ingrese 3 para calcular la velocidad del cuerpo")

    print("Ingrese 4 para presentar la grafica de desplazamiento en funcion del tiempo")

    guia = int(input())

    if guia == 1:

        print("Ingrese la velocidad del cuerpo")

        v = float(input())

        print("Ingrese el tiempo")

        t = float(input())

        x = v*t

        print("El objeto se traslado ", x, "unidades de su posicion inicial")

        guia = 0

    elif guia == 2:

        print("Ingrese la velocidad del cuerpo")

        v = float(input())

        print("Ingrese la distancia respecto al inicio")

        x = float(input())

        t = x/v

        print("El objeto tardo ", t, "unidades en llegar a la posicion")

        guia = 0

    elif guia == 3:

        print("Ingrese la distancia que recorrio el cuerpo respecto al inicio")

        x = float(input())
```

```
print("Ingrese el tiempo que tardo en recorrer esta distancia")

t = float(input())

v = x/t

print("El objeto tiene una velocidad ", v)

guia = 0

elif guia == 4:

    print("Ingrese la velocidad del cuerpo")

    v = float(input())

    t = np.arange(0.0, 10.0)

    x = v*t

    fig, ax = plt.subplots()

    ax.plot(t, x)

    ax.set(xlabel='Tiempo', ylabel='Desplazamiento',

           title='Grafica de desplazamiento en funcion del tiempo')

    ax.grid()

    fig.savefig("MRU")

    plt.show()

    guia = 0

    sys.exit()
```

Anexo 2, MRUV.

```
import matplotlib.pyplot as plt
import numpy as np
import sys
guia = 0
while guia == 0:
    print("Ingrese 1 para calcular la posicion del cuerpo")
    print("Ingrese 2 para calcular la velocidad del cuerpo en un momento especifico")
    print("Ingrese 3 para presentar las graficas en funcion del tiempo")
    guia = int(input())
    if guia == 1:
        print("Ingrese la velocidad del cuerpo")
        vi = float(input())
        print("Ingrese el tiempo")
        t = float(input())
        print("Ingrese la aceleracion del cuerpo")
        a = float(input())
        
$$x = (vi*t) + ((a*((t)**2))/2)$$

        print("El objeto se traslado ", x, "unidades de su posicion inicial")
        guia = 0
    elif guia == 2:
        print("Ingrese la velocidad inicial del cuerpo")
        vi = float(input())
        print("Ingrese el tiempo en el punto a calcular")
        t = float(input())
        print("Ingrese la aceleracion del cuerpo")
        a = float(input())
        
$$vf = vi + (a*t)$$

        print("El objeto tiene una velocidad ", vf)
        guia = 0
    elif guia == 3:
```

```
print("Ingrese la velocidad del cuerpo")
vi = float(input())
print("Ingrese la aceleracion del cuerpo")
a = float(input())
t = np.arange(0.0, 10.0)
x = (vi*t)+((a*((t)**2))/2)
vf = vi+(a*t)
fig, ax = plt.subplots(2)
ax[0].plot(x)
ax[1].plot(vf)
ax[0].set(ylabel = "Desplazamiento")
ax[1].set(ylabel = "Velocidad", xlabel = "Tiempo")
fig.suptitle("Análisis del cuerpo en MRUV")
fig.savefig("MRUV")
plt.show()
guia = 0
sys.exit()
```

Anexo 3, Tiro Oblicuo.

```
import matplotlib.pyplot as plt
import numpy as np
import sys
guia = 0
while guia == 0:
    print("Ingrese 1 para calcular la posicion del cuerpo")
    print("Ingrese 2 para calcular la velocidad del cuerpo en un momento especifico")
    print("Ingrese 3 para presentar las graficas en funcion del tiempo")
    print("Ingrese 4 para presentar la animacion de movimiento del cuerpo")
    guia = int(input())
if guia == 1:
    print("Ingrese X o Y respecto a la dimension a trabajar")
    guia2 = input()
    if guia2 == "X":
        print("Ingrese la velocidad del cuerpo en el eje X")
        vix = float(input())
        print("Ingrese el tiempo")
        t = float(input())
        print("Ingrese la aceleracion del cuerpo en el eje X")
        ax = float(input())
        
$$x = (vix*t) + ((ax*((t)**2))/2)$$

        print("El objeto se traslado ", x, "unidades de su posicion inicial")
    elif guia2 == "Y":
        print("Ingrese la velocidad del cuerpo en el eje Y")
        viy = float(input())
        print("Ingrese el tiempo")
        t = float(input())
        print("Ingrese la aceleracion del cuerpo en el eje Y")
        ay = float(input())
        
$$y = (viy*t) + ((ay*((t)**2))/2)$$

```

```
    print("El objeto se traslado ", y, "unidades de su posicion inicial")
    guia = 0
elif guia == 2:
    print("Ingrese X o Y respecto a la dimension a trabajar")
    guia3 = input()
    if guia3 == "X":
        print("Ingrese la velocidad del cuerpo en el eje X")
        vix = float(input())
        print("Ingrese el tiempo")
        t = float(input())
        print("Ingrese la aceleracion del cuerpo en el eje X")
        ax = float(input())
        vfx = vix+(ax*t)
        print("El objeto tiene una velocidad ", vfx)
    elif guia3 == "Y":
        print("Ingrese la velocidad del cuerpo en el eje Y")
        viy = float(input())
        print("Ingrese el tiempo")
        t = float(input())
        print("Ingrese la aceleracion del cuerpo en el eje Y")
        ay = float(input())
        vfy = viy+(ay*t)
        print("El objeto tiene una velocidad ", vfy)
    guia = 0
elif guia == 3:
    print("Ingrese la velocidad del cuerpo en el eje X")
    vix = float(input())
    print("Ingrese la velocidad del cuerpo en el eje Y")
    viy = float(input())
    print("Ingrese la aceleracion del cuerpo en el eje X")
    ax = float(input())
```

```

print("Ingrese la aceleracion del cuerpo en el eje Y")
ay = float(input())
t = np.arange(0.0, 10.0)
x = (vix*t)+((ax*((t)**2))/2)
vfx = vix+(ax*t)
y = (viy*t)+((ay*((t)**2))/2)
vfy = viy+(ay*t)
fig, ax = plt.subplots(2, 2)
fig.set_size_inches(14.5, 10.5)
ax[0, 0].plot(x)
ax[1, 0].plot(vfx)
ax[0, 1].plot(y)
ax[1, 1].plot(vfy)
ax[0, 0].set(ylabel = "Desplazamiento en X")
ax[1, 0].set(ylabel = "Velocidad en X", xlabel = "Tiempo")
ax[0, 1].set(ylabel = "Desplazamiento en Y")
ax[1, 1].set(ylabel = "Velocidad en Y", xlabel = "Tiempo")
fig.suptitle("Análisis del cuerpo en Tiro Oblicuo")
fig.savefig("Tiro Oblicuo info")
plt.show()
guía = 0
sys.exit()
elif guía == 4:
print("Ingrese la velocidad del cuerpo en el eje X")
vix = float(input())
print("Ingrese la velocidad del cuerpo en el eje Y")
viy = float(input())
print("Ingrese la aceleracion del cuerpo en el eje X")
ax = float(input())
print("Ingrese la aceleracion del cuerpo en el eje Y")
ay = float(input())

```

```
t = np.arange(0.0, 10.0)
x = (vix*t)+((ax*((t)**2))/2)
y = (viy*t)+((ay*((t)**2))/2)
fig, ax = plt.subplots()
ax.plot(x, y)
ax.set(ylabel = "Desplazamiento en Y", xlabel = "Desplazamiento en X")
fig.suptitle("Trayectoria del cuerpo en Tiro Oblicuo")
fig.savefig("Tiro Oblicuo trayectoria")
plt.show()
```

Referencias.

https://matplotlib.org/stable/tutorials/introductory/sample_plots.html

<https://interactivechaos.com/es/manual/tutorial-de-matplotlib/la-funcion-subplots>

<https://www.uaeh.edu.mx/scige/boletin/prepa2/n2/m5.html>

<https://dspace.ups.edu.ec/bitstream/123456789/5642/1/Manual%20de%20cinematica%20y%20dinamica.pdf>

<https://www.mineduc.gob.gt/DIGECADE/documents/Telesecundaria/Recursos%20Digitales/10%20Recursos%20Digitales%20TS%20licencia%20CC%20BY-SA%203.0/01%20CIENCIAS%20NATURALES/U9%20pp%20191%20Cinem%C3%A1tica.pdf>