

Proyecto final

MRU, MRUV, Tiro Oblicuo

Victoria González Fuhr

30/06/2023

Introducción

En el siguiente trabajo se van a presentar algunos de los aspectos más importantes de la cinemática en una y dos dimensiones, trabajando con un programa llamado Python para analizar distintas situaciones dónde se involucran estos conceptos.

Marco teórico

La cinemática es una rama de la física que se ocupa del estudio del movimiento de los objetos sin considerar las causas que lo producen. En particular, la cinemática en una y dos dimensiones se encarga de estudiar el movimiento de objetos en línea recta y en un plano. Los conceptos fundamentales incluyen la posición, la velocidad y la aceleración, y se utilizan ecuaciones y fórmulas para describir matemáticamente el movimiento. Estos son fundamentales para comprender el comportamiento de los principales objetos en movimiento y son la base para otros campos de estudio dentro de la física, como la dinámica y la mecánica.

Palabras claves

MRU, MRUV, Tiro oblicuo.

Desarrollo

Para trabajar los conceptos mencionados anteriormente no vamos a tener en cuenta la fuerza de rozamiento y vamos a trabajar a todos los objetos como masas puntuales.

Empezamos trabajando con MRU (movimiento rectilíneo uniforme), podemos ver que éste depende de la velocidad y a su vez, la velocidad depende de la razón de cambio de la posición respecto del tiempo, por lo que su unidad de medida es m/s.

La posición, que se mide en metros, es la ubicación respecto de un punto cero en un sistema de referencia.

El desplazamiento (ΔX) es el cambio de posición entre un instante inicial y otro final. Es decir, $\Delta x = x_f - x_i$

Hay dos ecuaciones que describen a la velocidad, la cual ya definimos anteriormente:

Velocidad promedio (V_p) = $\frac{\Delta x}{\Delta t}$ (ésta es la que vamos a utilizar)

Velocidad instantánea (V) = $\lim_{\Delta t \rightarrow 0} \left(\frac{\Delta x}{\Delta t} \right) = \frac{dx}{dt}$

La ecuación para MRU que se utiliza es: $\Delta x = v\Delta t$

Teniendo en cuenta los conceptos previamente desarrollados podemos recurrir a un programa que calcule alguno de los tres datos teniendo en cuenta los otros

dos, además, podemos ver un gráfico del desplazamiento del cuerpo en función del tiempo transcurrido.

Ver anexo 1 para ver el código y la gráfica.

Ahora que ya vimos la cinemática en una dimensión, podemos verla en dos dimensiones (MRUV).

En MRUV (movimiento rectilíneo uniformemente acelerado) debemos tener en cuenta que hay una aceleración “a” que causa un cambio en la velocidad, es decir que ya no es constante.

La aceleración se define como la razón de cambio de la velocidad a medida que pasa el tiempo y, al igual que la velocidad, la aceleración también se puede calcular matemáticamente de dos maneras.

Además, hay que tener en cuenta que la velocidad varía, entonces debemos definir una velocidad inicial (v_i) y otra final (v_f) al momento de realizar los cálculos.

Aceleración promedio (a_p) = $\frac{\Delta v_p}{\Delta t}$ (ésta es la que vamos a utilizar)

Aceleración instantánea (a) = $\lim_{\Delta t \rightarrow 0} \left(\frac{\Delta v}{\Delta t} \right) = \frac{dv}{dt}$

Las ecuaciones para MRUV son:

$$\Delta x = v_i \Delta t + \frac{1}{2} a \Delta t^2$$

$$v_f = v_i + a \Delta t$$

$$v_f^2 = v_i^2 + 2a \Delta x$$

Las fórmulas anteriores son solo una manera de calcular algunos datos dependiendo de los que datos dispongamos, pero en el programa se encontrarán con otras ecuaciones que derivan de éstas tres.

En el anexo 1 se podrán encontrar con el código y las gráficas de desplazamiento, velocidad y aceleración en función del tiempo.

Por último, se va a presentar el tema de tiro oblicuo que es un caso particular del movimiento de caída libre en el que la velocidad inicial del proyectil forma cierto ángulo con la horizontal, dando como consecuencia una trayectoria parabólica.

Para este tema hay que tener en cuenta que las fórmulas son las mismas que ya se presentaron anteriormente, pero se va a tener en cuenta las variables de las dos dimensiones por separado.

En el anexo 1 se van a incluir algunos gráficos de tiro oblicuo, donde hay que tener en cuenta la posición inicial y final en X e Y, la velocidad inicial y final en X e Y, el tiempo que demora el objeto en caer y la aceleración en Y, donde solo se

contempla la aceleración de gravedad ya que no hay ninguna fuerza externa que me pueda variar el movimiento de la masa puntual.

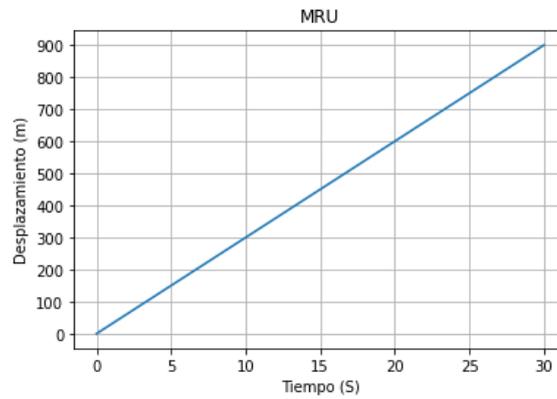
Conclusión

Como se puede ver en el trabajo, la cinemática ayuda a entender de forma simple como es la trayectoria de una masa puntual en diferentes situaciones y variando distintos factores, aunque esto no puede aplicarse a la vida cotidiana porque hay muchos factores que no se están teniendo en cuenta, por ejemplo, la fuerza de rozamiento.

Anexo 1

Gráficos de MRU

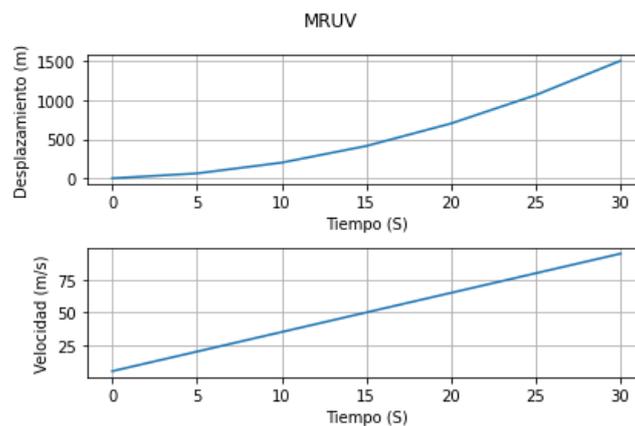
Velocidad = 30 m/s



Gráficos de MRUV

Velocidad inicial = 5 m/s

Aceleración = 3 m/s²



Gráficos de Tiro oblicuo

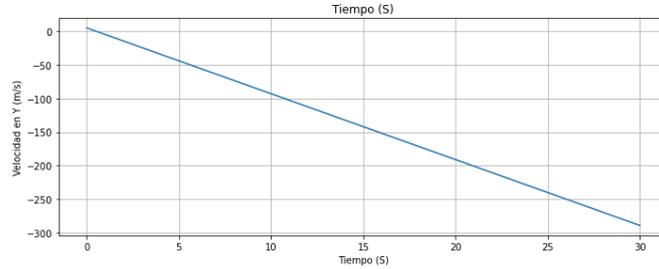
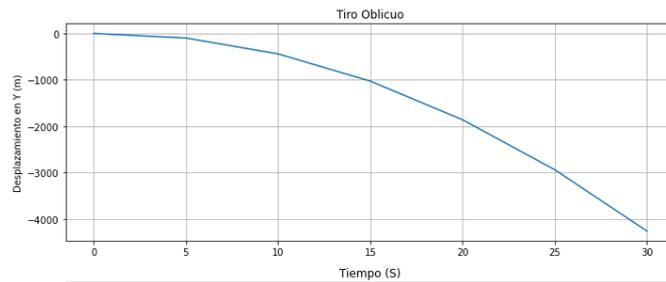
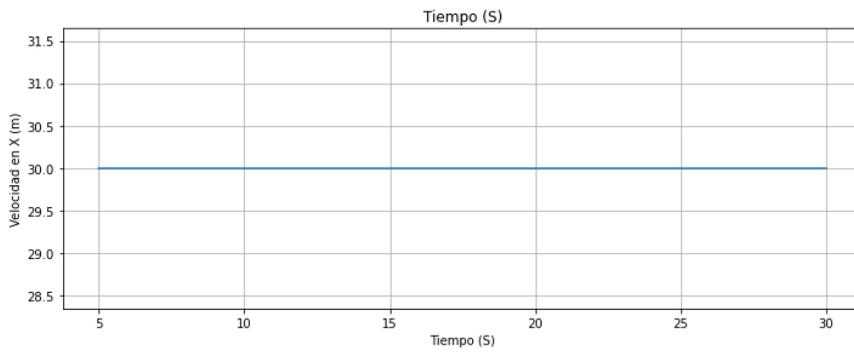
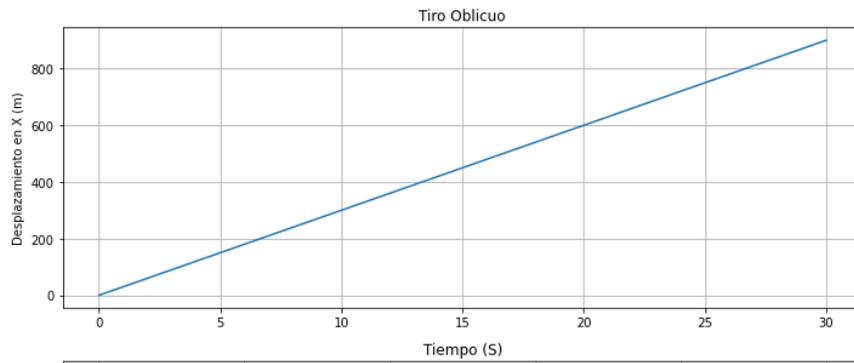
En x:

Velocidad = 30 m/s

En y:

Velocidad = 5 m/s

Aceleración = -9.8 m/s²



Programa

Created on Wed Jun 18 00:56:00 2023

@author: Victoria Gonzalez

```
import matplotlib.pyplot as plt
```

```
import sys
```

```
import numpy as np
```

```
sigue = "s"
```

```
while sigue == "s":
```

```
    print("Menú")
```

```
    print("1- Calcular MRU")
```

```
    print("2- Calcular MRUV")
```

```
    print("3- Calcular Tiro Oblicuo")
```

```
    print("0- Salir")
```

```
opcion = int(input())
```

```
if opcion == 1:
```

```
    print("Ingrese qué variable desea calcular")
```

```
    print("1- Desplazamiento")
```

```
    print("2- Velocidad")
```

```
    print("3- Tiempo")
```

```
    print("4- Ver gráfico del desplazamiento en función del tiempo")
```

```
    print("0- Volver")
```

```
subopcion = int(input())
```

```
if subopcion == 1:
```

```
    xi = float(input("Ingrese la posición inicial en metros: "))
```

```
    v = float(input("Ingrese la velocidad en m/s: "))
```

```
    t = float(input("Ingrese el tiempo en segundos: "))
```

```
    x = xi + v * t
```

```
    print("El desplazamiento es:", x, "m")
```

```
elif subopcion == 2:
```

```
    xi = float(input("Ingrese la posición inicial en metros: "))
```

```
x = float(input("Ingrese el desplazamiento en metros: "))
t = float(input("Ingrese el tiempo en segundos: "))
v = (x - xi) / t
print("La velocidad es:", v, "m/s")
```

```
elif subopcion == 3:
```

```
xi = float(input("Ingrese la posición inicial en metros: "))
x = float(input("Ingrese el desplazamiento en metros: "))
v = float(input("Ingrese la velocidad en m/s: "))
t = (x - xi) / v
print("El tiempo es:", t, "segundos")
```

```
elif subopcion == 4:
```

```
xi = float(input("Ingrese la posición inicial en metros: "))
v = float(input("Ingrese la velocidad del cuerpo en m/s: "))
t = np.arange(0, 31, 5)
x = xi + v * t
fig, ax = plt.subplots()
ax.plot(t, x)
ax.set(xlabel = "Tiempo (S)", ylabel = "Desplazamiento (m)", title =
"MRU")
ax.grid(True)
ax.set_yticks(np.arange(0, np.max(x)+1, 100))
fig.savefig("MRU.png")
plt.show()
```

```
elif opcion == 2:
```

```
print("Ingrese qué variable desea calcular")
print("1- Posición final de la masa puntual")
print("2- Velocidad final de la masa puntual en un momento específico")
print("3- Aceleración de la masa puntual")
```

```
print("4- Gráfico del desplazamiento y la velocidad en función del tiempo")
print("o- Volver")
```

```
subopcion = int(input())
```

```
if subopcion == 1:
```

```
    xi = float(input("Ingrese la posición inicial en metros: "))
    vi = float(input("Ingrese la velocidad inicial de la masa puntual: "))
    t = float(input("Ingrese el tiempo en segundos: "))
    a = float(input("Ingrese la aceleración de la masa puntual en m/s2: "))
    x = xi + (vi * t) + ((a * (t ** 2)) / 2)
    print("El desplazamiento es:", x, "m")
```

```
elif subopcion == 2:
```

```
    vi = float(input("Ingrese la velocidad inicial de la masa puntual: "))
    t = float(input("Ingrese el tiempo en segundos: "))
    a = float(input("Ingrese la aceleración de la masa puntual en m/s2: "))
    vf = vi + (a * t)
    print("La velocidad es:", vf, "m/s")
```

```
elif subopcion == 3:
```

```
    xi = float(input("Ingrese la posición inicial en metros: "))
    x = float(input("Ingrese el valor del desplazamiento en metros: "))
    vi = float(input("Ingrese el valor de la velocidad inicial en m/s: "))
    t = float(input("Ingrese el valor del tiempo en s: "))
    a = (2 * ((x - xi) - vi * t) / (t ** 2))
    print("La aceleración es:", a, "m/s2")
```

```
elif subopcion == 4:
```

```
    xi = float(input("Ingrese el valor de la posición inicial en m: "))
    vi = float(input("Ingrese la velocidad del cuerpo en m/s: "))
```

```

a = float(input("Ingrese la aceleracion del cuerpo en m/s2: "))
t = np.arange(0, 31, 5)
x = xi + (vi * t) + ((a * ((t) ** 2)) / 2)
vf = vi + (a * t)
fig, ax = plt.subplots(2)
ax[0].plot(t, x)
ax[1].plot(t, vf)
ax[0].set(ylabel = "Desplazamiento (m)", xlabel = "Tiempo (S)")
ax[1].set(ylabel = "Velocidad (m/s)", xlabel = "Tiempo (S)")
ax[0].grid(True) # Cuadrícula
ax[1].grid(True) # Cuadrícula
fig.suptitle("MRUV")
fig.tight_layout() #Ajusta los márgenes y evita superposiciones de los
rótulos y otros elementos
fig.savefig("MRUV.png")
plt.show()

elif subopcion == 0:
    continue

elif opcion == 3:
    print("Tiro Oblicuo")
    print("Ingrese qué variable desea calcular")
    print("1- Calcular la posicion del cuerpo en X")
    print("2- Calcular la posicion inicial del cuerpo en Y")
    print("3- Calcular la posicion final del cuerpo en Y")
    print("4- Velocidad de la masa puntual en X")
    print("5- Velocidad inicial de la masa puntual en Y")
    print("6- Velocidad final de la masa puntual en Y")
    print("7- Tiempo que tarda el cuerpo en recorrer la trayectoria en términos
de x")
    print("8- Tiempo que tarda el cuerpo en recorrer la trayectoria en términos
de y")

```

```

print("9- Gráficos de la posición en x y la velocidad en x en función del
tiempo")

print("10- Gráficos de la posición en y y la velocidad en y en función del
tiempo")

print("0- Volver")

subopcion = int(input())

if subopcion == 1:
    xi = float(input("Ingrese la posición inicial en x de la masa puntual en m:
"))
    vx = float(input("Ingrese la velocidad de la masa puntual en x: "))
    t = float(input("Ingrese el tiempo en segundos: "))
    x = xi + vx * t
    print("La posición final en x es:", x, "m")

elif subopcion == 2:
    yf = float(input("Ingrese la posición final en y de la masa puntual: "))
    vi = float(input("Ingrese la velocidad inicial en y de la masa puntual: "))
    t = float(input("Ingrese el tiempo en segundos: "))
    a = float(input("Ingrese la aceleración de la gravedad en m/s2: "))
    yi = yf - vi * t + 0.5 * a * (t ** 2)
    print("La posición inicial en y es:", yi, "m")

elif subopcion == 3:
    yi = float(input("Ingrese la posición inicial en y de la masa puntual: "))
    vi = float(input("Ingrese la velocidad inicial en y de la masa puntual: "))
    t = float(input("Ingrese el tiempo en segundos: "))
    a = float(input("Ingrese la aceleración de la gravedad en m/s2: "))
    yf = yi + vi * t + 0.5 * a * (t ** 2)
    print("La posición final en y es:", yf, "m")

elif subopcion == 4:

```

```

    xi = float(input("Ingrese la posición inicial en x de la masa puntual en m:
"))
    x = float(input("Ingrese la posición final en x de la masa puntual en m: "))
    t = float(input("Ingrese el tiempo en segundos: "))
    v = (x - xi)/t
    print("La velocidad en x es:", v, "m/s")

```

```

elif subopcion == 5:

```

```

    vfy = float(input("Ingrese la velocidad final en m/s: "))
    a = float(input("Ingrese la aceleracion de la gravedad de m/s²: "))
    t = float(input("Ingrese el tiempo en segundos: "))
    viy = vfy-a*t
    print("La velocidad final en y es:", viy, "m/s")

```

```

elif subopcion == 6:

```

```

    viy = float(input("Ingrese la velocidad inicial en m/s: "))
    a = float(input("Ingrese la aceleracion de la gravedad de m/s²: "))
    t = float(input("Ingrese el tiempo en segundos: "))
    vfy = viy+a*t
    print("La velocidad final en y es:", vfy, "m/s")

```

```

elif subopcion == 7:

```

```

    v = float(input("Ingrese la velocidad promedio en x en m/s: "))
    xo = float(input("Ingrese la posición inicial en x de la masa puntual en m:
"))
    x = float(input("Ingrese la posición final en x de la masa puntual en m: "))
    t = (x-xo)/v
    print("El tiempo que tardara en recorrer esa trayectoria es:", t,
"segundos")

```

```

elif subopcion == 8:

```

```

    viy = float(input("Ingrese la velocidad inicial en m/s: "))

```

```

vif = float(input("Ingrese la velocidad final en m/s: "))
a = float(input("Ingrese la aceleracion de la gravedad de m/s2: "))
vfy = viy+a*t
t = (vfy-viy)/a
print("El tiempo que tardara en recorrer esa trayectoria es:", t,
"segundos")

elif subopcion == 9:
    xi = float(input("Ingrese la posicion inicial en metros: "))
    vx = float(input("Ingrese la velocidad de la masa puntual en el eje x en
m/s: "))
    t = np.arange(0, 31, 5)
    x = xi + vx * t
    vx = (x - xi) / t

    fig, axs = plt.subplots(2)
    fig.set_size_inches(10, 10)

    axs[0].plot(t, x)
    axs[0].set_ylabel("Desplazamiento en X (m)")
    axs[0].set_title("Tiro Oblicuo")
    axs[0].grid(True)

    axs[1].plot(t, vx)
    axs[1].set_xlabel("Tiempo (S)", ylabel="Velocidad en X (m)")
    axs[1].set_title("Tiempo (S)")
    axs[1].grid(True)
    plt.tight_layout()
    fig.savefig("TOX.png")
    plt.show()

```

```

elif subopcion == 10:
    yi = float(input("Ingrese el valor de la posición inicial en metros: "))
    viy = float(input("Ingrese la velocidad de la masa puntual en el eje y en
m/s: "))
    ay = float(input("Ingrese la aceleracion de la gravedad en m/s2: "))
    t = np.arange(0, 31, 5)
    yf = yi + (viy * t) + ((ay * ((t) ** 2)) / 2)
    vfy = viy + (ay * t)

    fig, axs = plt.subplots(2)
    fig.set_size_inches(10, 8)

    axs[0].plot(t, yf)
    axs[0].set_ylabel("Desplazamiento en Y (m)")
    axs[0].set_title("Tiro Oblicuo")
    axs[0].grid(True)

    axs[1].plot(t, vfy)
    axs[1].set_xlabel("Tiempo (S)", ylabel="Velocidad en Y (m/s)")
    axs[1].set_title("Tiempo (S)")
    axs[1].grid(True)

    plt.tight_layout()
    fig.savefig("TOY.png")
    plt.show()

elif opcion == 0:
    print("Programa finalizado, gracias por su atención ;)")
    sys.exit()

```

Referencias:

- Presentación de cinemática en una y dos dimensiones de Juan Pablo Umazano del año 2021.
- SERWAY, R.; JEWET, J. 2009. Física para ciencias e ingeniería con Física Moderna. Vol I. México. Cengage Learning Editores.
- <https://www.electrontools.com/Home/WP/tiro-oblicuo-formulas-y-ejercicios-resueltos/>
- <https://plotly.com/python/>
- <https://aprendeconalf.es/docencia/python/manual/matplotlib/>
- <https://interactivechaos.com/es/manual/tutorial-de-matplotlib/la-funcion-subplots>
- <https://www.youtube.com/watch?v=voIKt9lmozE>
- <https://www.youtube.com/watch?v=QPpHUBphDgo&t=24s>
- <https://www.lifeder.com/tiro-parabolico-oblicuo/>