



# Aproximación de la Trayectoria de un Cohete Multietapa en Python

## Approximation of the Path of Multistage Rocket in Python

J. HERNANDEZ-BARAHONA<sup>1,†</sup>, J. BARAHONA-TAVALERA<sup>2,†</sup>, C. VALDES-SANCHEZ<sup>1,†</sup>

Recibido: 5 de abril 2021 / Aceptado: 29 de mayo de 2021

<sup>1</sup>Escuela de Física, Facultad de Ciencias,  
Universidad Nacional Autónoma de Honduras.  
cfvaldes@unah.edu.hn  
jocsan.hernandez@unah.edu.hn

<sup>2</sup>Instituto Técnico 18 de Noviembre.  
juan.fernando@educatrachos.hn

### RESUMEN

Planteamos la aproximación de la trayectoria de vuelo de un cohete de dos etapas, haciendo uso de un modelo matemático que tiene en consideración la fuerza de gravedad ejercida al cohete por el cuerpo masivo del cual parte, así también como la fuerza de resistencia del aire, la cual aumenta según incrementa la velocidad del cohete.

La aproximación matemática se logró utilizando programación orientada a objetos en el lenguaje de programación de Python. Además se ha utilizado una sencilla interfaz de usuario para que el producto final sea amigable para los interesados en la aeronáutica y no así en la programación. Cabe destacar que el producto final puede ser utilizado por cualquiera con acceso a una computadora y a internet, ya que no se programó para un compilador en particular. Finalmente proponemos que el programa se extienda a cualquier tipo de cohete mediante la modificación del código aquí presentado.

### ABSTRACT

We use an approximation of the flight trajectory for a two-stage rocket, making use of a mathematical model that takes into account the weight of the rocket itself, but also air resistance that will rise as the rockets speeds up.

We achieved the mathematical approximation using object orientated programming in Python language. Also the final product has a GUI so it is available to use for anyone who wants to make use of this product whether they know of programming or not. It is worthnotting that the final product may be used by anyone who has access to internet, since the program was not design for a particular compiler or OS. Finally we propose as future work that the code is extended to any type of rocket.

### PALABRAS CLAVES

Flujo Másico, Etapa, Factor de Carga, Propelente, Python, Impulso, Programación Orientada a Objetos, Empuje, Interfaz de Usuario.

### KEYWORDS

Mass Flow, Stage, Load Factor, Propellant, Python, Specific Boost, Object Oriented Programming, Thrust, GUI.

\* Esta obra está bajo una licencia Creative Commons Reconocimiento - NoComercial 4.0 Internacional

\* This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

## **I | AGRADECIMIENTOS**

Este proyecto es posible debido a uno de los aspectos principales que competen al quehacer universitario, la vinculación o extensión universidad-sociedad. El proyecto es producto directo de un curso libre de la Escuela de Física llamado "Introducción a la programación interactiva en Python". Este artículo no sería posible sin todos los involucrados en la creación, impartición y realización del curso. Agradecimiento infinito para todos ellos.

## **II | INTRODUCCIÓN**

Un cohete tipo lanzador es una máquina que, utilizando un motor de combustión, produce la energía cinética necesaria para la expansión de los gases, que son lanzados a través de un tubo propulsor (propulsión a reacción). Por extensión, el vehículo, generalmente espacial, que presenta motor de propulsión de este tipo es denominado cohete o misil. Normalmente, su propósito es enviar artefactos a la órbita terrestre (especialmente satélites artificiales y sondas espaciales). Un cohete está formado por una estructura, un motor de propulsión a reacción y una carga útil. La estructura sirve para proteger los tanques de combustible y oxidante y la carga útil. En este artículo estaremos modelando un cohete lanzador de dos etapas cuya carga útil será un satélite artificial.

## **III | DESCRIPCIÓN DEL MODELO MATEMÁTICO**

Colocar un satélite en órbita implica un desafío en el que va implícito mucho cálculo, la fuerza de atracción de la Tierra con un objeto que busca escapar de su influencia gravitacional, además de las fuerzas climatológicas que influye en el objetivo del cohete de colocar un satélite en la órbita terrestre, usando las leyes de Newton empezamos con la mezcla de combustible y comburente, estos se queman en una cámara de combustión generando gases expelidos por las boquillas de la tobera con tanta fuerza que supera la fuerza peso del cohete, impulsando así el cohete hacia arriba con una aceleración constante, a medida que el combustible disminuye, la velocidad aumenta, permitiendo que el cohete tenga velocidad de escape terrestre, en su largo recorrido el cohete debe deshacerse de lastre por tanto se utiliza un cohete a dos etapas, donde la primera etapa impulsa el cohete y la otra etapa orienta al satélite en una órbita elíptica con respecto a la tierra. La misión del vuelo de un cohete se puede definir en tres fases bien marcadas:

- i) La etapa de despegue y vuelo propulsado con sucesiva liberación de etapas hasta alcanzar el corte. El corte se refiere al punto en la trayectoria del cohete donde la propulsión del motor ya no es necesaria y este se apaga.
- ii) La etapa correspondiente a la trayectoria de inercia, en esta parte es cuando el cohete se ve sujeto a las fuerzas de arrastre y la fuerza gravitatoria que buscan evitar que el cohete se libere de la órbita terrestre.
- iii) La etapa del inicio de trayectoria orbital de la carga útil.

El problema aquí abordado se asocia a la fase número uno, en la cual hay que buscar la trayectoria desde el punto de lanzamiento hasta la inyección de la carga útil (Canaval, 2014).

Para lograr analizar dicha trayectoria partiremos de la conservación del momento lineal, donde plantearemos la igualdad de las fuerzas externas que afecten nuestro cohete como ser la gravedad y la fuerza de resistencia aerodinámica las cuales serán iguales al cambio en el momento del sistema, por la segunda ley de Newton, obteniendo como resultado una ecuación diferencial ordinaria de primer orden, luego adaptaremos este modelo al lenguaje de Python utilizando Programación Orientada a Objetos logrando así obtener una predicción para la velocidad y posición del cohete para todo tiempo.

## 1 | Momento Lineal

Partiremos del análisis del movimiento del cohete donde debemos examinar el momento del sistema (cohete y combustible expulsado). Como el cohete está consumiendo combustible, su masa,  $m$ , disminuirá en función del tiempo. En el instante  $t$  el momento sería:

$$P(t) = m(t)v(t)$$

Con un cambio infinitesimal de tiempo ( $t + dt$ ), la masa del cohete cambiará a  $(m + dm)$ , siendo así el cambio en su momento lineal  $(m + dm)(v + dv)$ . Considerando que el combustible expulsado en el tiempo  $dt$  tiene una masa de  $-dm$ , ya que es la misma cantidad que se perdió; pero por nuestra convención es de signo opuesto y una velocidad de  $v - v_{esc}$  relativa al suelo, entonces el momento total de nuestro sistema es:

$$\begin{aligned} P(t + dt) &= (m + dm)(v + dv) - dm(v - v_{esc}) \\ P(t + dt) &= mv + mdv + dmv + dmdv - dmv + dm v_{esc} \\ P(t + dt) &= mv + mdv + dm v_{esc} \\ P(t + dt) - P &= mdv + dm v_{esc} \end{aligned}$$

Despreciando los términos pequeños, finalmente el momento de nuestro sistema en términos del cambio del momento con respecto al tiempo sería el siguiente:

$$dP = mdv + dm v_{esc} \quad (1)$$

## 2 | Ecuaciones de movimiento

En la segunda ley de Newton consideramos únicamente 2 fuerzas, la de arrastre y el peso del cohete, cualquier otra interacción que pudiera existir será despreciable en comparación a estas dos, bajo estas suposiciones:

$$\begin{aligned} \vec{F}_{ext} &= ma \\ \vec{F}_{ext} &= \frac{d}{dt}(vm) \\ \vec{F}_{ext} &= \frac{d}{dt}P \end{aligned}$$

$$\vec{F}_{ext} dt = dP \quad (2)$$

Juntando 1 y 2:

$$(-bv - mg)dt = mdv + dm v_{esc}$$

$$\frac{dv}{dt} + \frac{b}{m}v = \frac{-v_{esc}}{m} \frac{dm}{dt} - g \quad (3)$$

El resultado es una ecuación diferencial lineal de primer orden cuya solución es:

$$v(t) = v_{esc} \left[ 1 - m_0^{-\frac{b}{D}} (m_0 - Dt)^{\frac{b}{D}} \right] + \frac{g}{D-b} \left[ (m_0 - Dt) - m_0^{1-\frac{b}{D}} (m_0 - Dt)^{\frac{b}{D}} \right] + v_0 m_0^{-\frac{b}{D}} (m_0 - Dt)^{\frac{b}{D}} \quad (4)$$

De manera que hemos encontrado la ecuación de la velocidad del cohete . Ahora utilizando el hecho de que  $v = dy/dt$  y sustituyendo en (3) obtenemos la ecuación de movimiento para el cohete, es decir una ecuación diferencial para  $y$ , cuya solución está dada por:

$$y = y_0 + \frac{D}{b} v_{esc} t + \left[ \frac{v_{esc} \left( \frac{D^2}{b} - D \right) - v_0(D-b) + gm_0}{D^2 - b^2} \right] \times \left[ (m_0 - Dt)^{1+\frac{b}{D}} m_0^{-\frac{b}{D}} - m_0 \right] - \frac{g}{2(D-b)D} [(m_0 - Dt)^2 - m_0^2] \quad (5)$$

Ahora tenemos una ecuación que describe la posición vertical del cohete en un momento  $t$ . Así hemos logrado nuestro objetivo, a saber: hemos encontrado una aproximación matemática que describe la posición del cohete en un tiempo dado (Kaff, 2015).

Cabe mencionar que la ecuación es engorrosa a todas luces, he aquí la importancia de la programación: hacer estos cálculos con papel y lápiz supone una tarea hercúlea, Python puede hacer el cálculo por nosotros en cuestión de decimas de segundos.

#### IV | SIMULACIÓN DE LA TRAYECTORIA EN PYTHON

Para simular la trayectoria del cohete se ha utilizado programación orientada a objetos en el lenguaje de alto nivel conocido como Python.

Para lograrlo creamos una clase a la que se denomina cohete, en ella están comprimidas todas las características de un cohete de dos etapas. Entre estos atributos están la masa inicial (Martínez, 2017), un método que calcula el flujo másico y por supuesto el cálculo de su posición.

De los mayores logros de este trabajo es que la simulación se lleva a cabo exitosamente con apenas 102 línea de código (espacios y comentarios incluidos). Aunque el cuerpo del programa en realidad tiene apenas 4 líneas.

Finalmente para la parte de interfaz de usuario el cohete se dibuja como un punto y la tierra como un círculo.

## V | PRODUCTO FINAL

A continuación está el código con el que se logró la simulación, donde como es convención en Python se contraen las clases y sus métodos:

```
#Importando modulos
import math, os
import SimpleGUICS2Pygame.simpleguics2pygame as simplegui
os.system("clear")

# Constantes
# Coeficiente de rozamiento del aire
b = 0.2
# gravedad de la Tierra
g = 9.8
# impulso especifico
isp = 225
# Carga util que lleva el cohete
mUtil = 5000

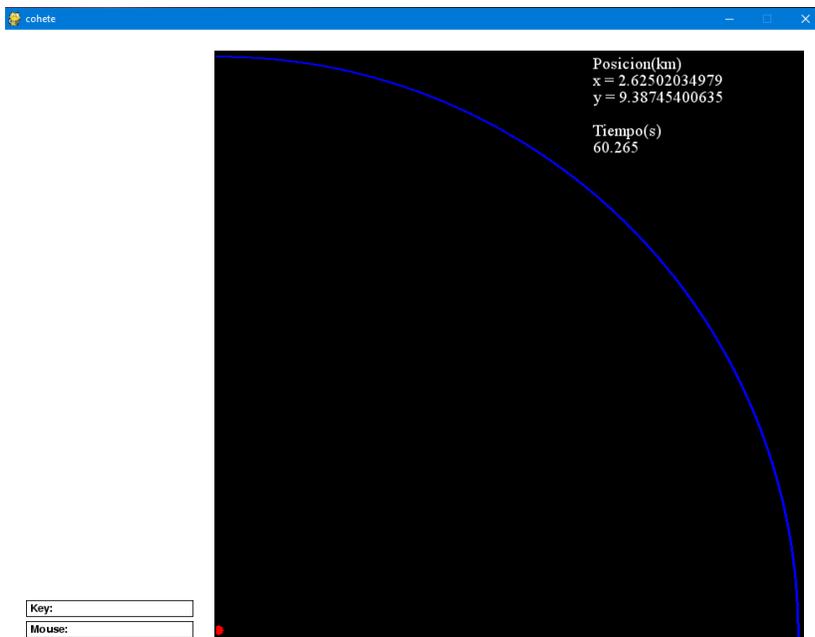
# Variables globales
BANDERA = True
#Masas de las distintas etapas en Kg
MP,MP2 = 50000,50000
MS,MS2 = 5000,5000
#Parametros iniciales
P = 0
POSICION =[0,0]
TIEMPO = 0
V = 0
VO = 0
Y = 0
YO = 0

class Cohete:
    #Atributos iniciales del cohete
    def __init__(self,MS,MS2,MP,MP2, isp, mUtil):
    #Calculo de masa inicial
    def masa_inicial(self,MS1,MS2,MP1,MP2, mUtil):
    #Calculos relevantes a la expulsion de gases
    def gases_tobera(self):
    #Calculo del flujo masico
    def flujo_masico(self,MP, isp):
    #Calculo de la posicion
    def posicion(self, TIEMPO):
#Dibujo en interfaz de usuario
def dibujar(lienzo):
```

```
# Cuerpo del programa
# creacion de cohetes
cohete_1 = Cohete(MS,MS2,MP,MP2, isp , mUtil)

marco = simplegui.create_frame(" cohete ", 700, 700)
marco.set_draw_handler(dibujar)
marco.start()
```

## 1 | Funcionamiento



*Figura 1: Programa simulando 60 s de recorrido*

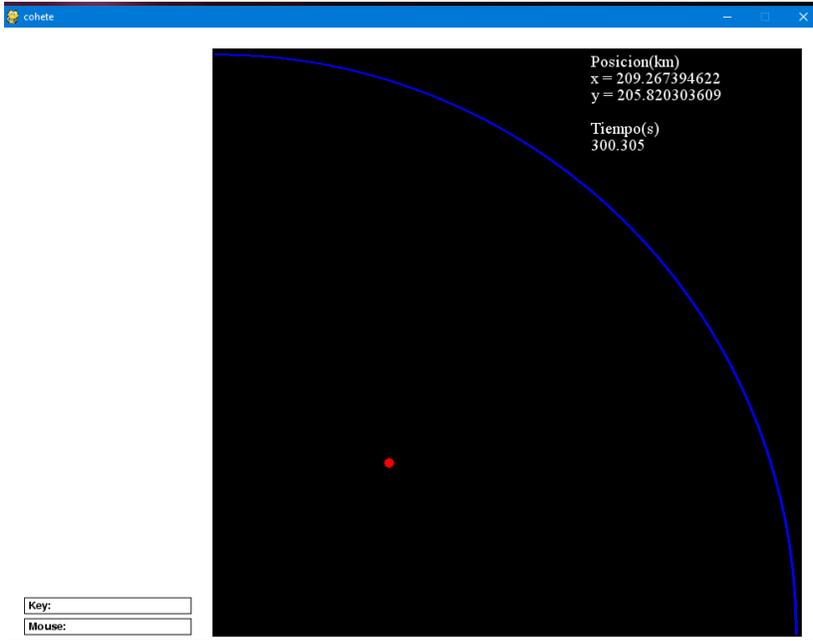


Figura 2: Programa simulando 300 s de recorrido

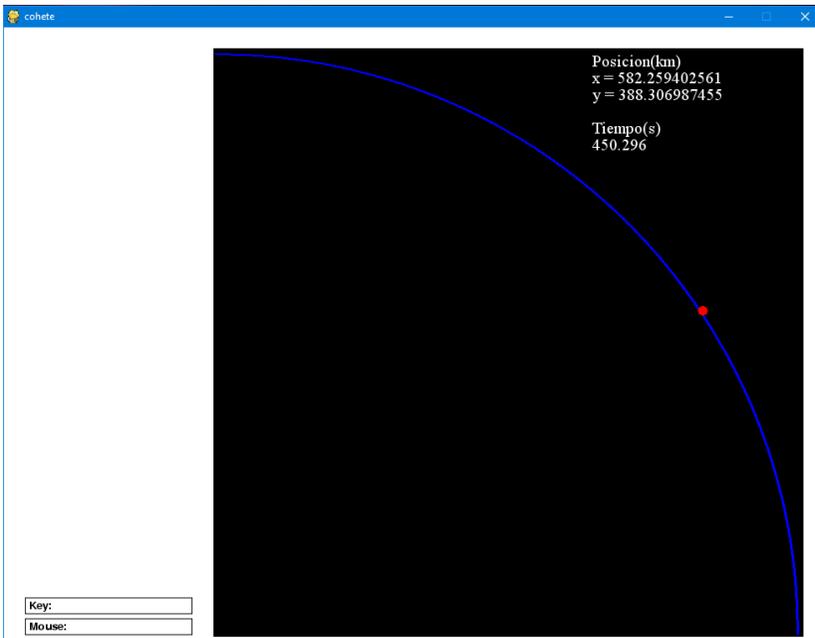


Figura 3: Programa simulando 450s final del recorrido

## VI | COMENTARIOS FINALES

La trayectoria de vuelo de un cohete es un fenómeno físico en el que influyen muchas variables, lo que hace complicado obtener una solución analítica, pero es posible obtener una aproximación idealizando ciertos fenómenos como ser la Fuerza de empuje de la atmósfera, el arrastre, la sustentación y factores como la densidad del aire. Dejando de lado lo anterior obtuvimos una simulación del cohete considerando la fuerza gravitatoria producto del cuerpo masivo del cual partió nuestro objeto y la resistencia del aire que se opone a la velocidad de este, obtenido así un recorrido en tiempo real.

## VII | CONCLUSIONES Y TRABAJO A FUTURO

Se logró simular correctamente la trayectoria de vuelo de un cohete de dos etapas, como se logra observar en las figuras 1, 2 y 3, si bien el modelo es un tanto simplificado podría servir para hacer pruebas previas de cuánta carga útil puede llevar un cohete de este tipo así como demás parámetros.

Se buscará hacer la simulación más interactiva e incluir al menos un factor extra para hacer la simulación más realista. Así mismo se buscará que en el programa un usuario pueda introducir los diferentes parámetros de un cohete para que dicho usuario pueda utilizar como aproximación la simulación de nuestro programa.

También se dibujará la trayectoria del cohete como una línea continua para que sea más fácil para el usuario seguir la trayectoria de la misma, así como la posibilidad de dibujar los ejes coordenados, además se incluirá un código de colores para indicar cada etapa del cohete. Esperamos entregar este proyecto en las próximas ediciones de esta misma revista.

## I | REFERENCIAS

- Canaval, T. E., D. (2014). Optimización de la trayectoria de fase propulsada de un cohete lanzado. *Universidad de San Buenaventura Bogotá*.
- Kaff, A. (2015). Obtención y solución a la ecuación de movimiento de un cohete, actuando sobre él fuerzas externas del campo gravitacional y el rozamiento del aire proporcional a la velocidad. *Revista Mexicana de Física*, 6-10.
- Martínez. (2017). Ventajas de un cohete multietapa para el envío de cargas a diferentes órbitas terrestres. *Revista Ingeniería, Matemáticas y Ciencias de la Información*, 4-8.