

物理期末專題

飛彈飛行模擬

B11901073 林禹融
B11901148 李承彥

概述

基本原理講解
參數設定

程式講解

模擬結果分析

基本原理講解與參數設定

火箭受力

- 重力
- 科氏力
- 氣動力
- 引擎推力

參數

- 比衝
- 空氣密度
- 飛彈尺寸
- 球座標轉換

飛彈受力

重力

$$\vec{F} = \frac{GMm}{r^2} \hat{r}$$



科氏力

$$\vec{F}_c = -2m(\vec{\omega} \times \vec{v})$$












飛彈受力

空氣阻力

升力

○
$$\vec{F} = -\frac{1}{2}\rho c A v^2 \hat{v}$$

Shape		Drag Coefficient
Sphere	→ 	0.47
Half-sphere	→ 	0.42
Cone	→ 	0.50
Cube	→ 	1.05
Angled Cube	→ 	0.80
Long Cylinder	→ 	0.82
Short Cylinder	→ 	1.15
Streamlined Body	→ 	0.04
Streamlined Half-body	→ 	0.09

$$F = \rho V g$$

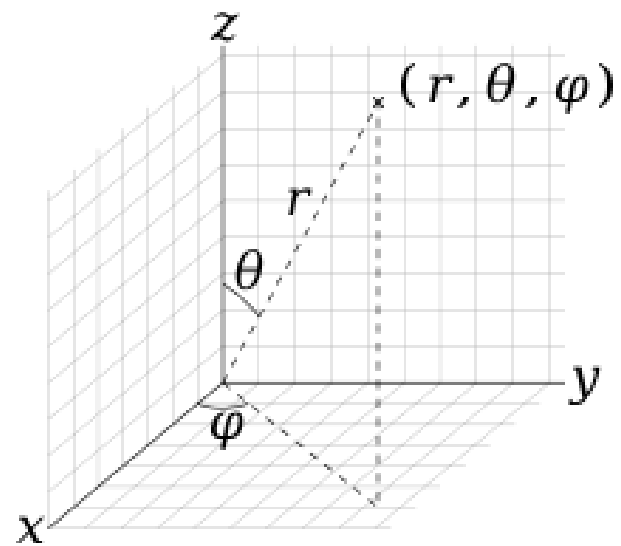


飛彈受力

火箭推力

$$\vec{F} = C * \hat{v}$$

球座標轉換



名詞解釋—比衝

燃燒1牛頓重的物質以產生
1牛頓的推力可持續時間

比衝高→效率好

參數設定

飛彈設定

- 雄三飛彈
- 質量：1,500公斤(彈頭重150公斤、引擎325公斤、燃料1025公斤)
- 假設燃料燃燒 kg/s = 飛彈損失重量 kg/s
- 長度：6.1公尺
- 直徑：46 cm

參數設定

引擎

參考引擎

- 第一代猛禽火箭
- 質量：2000kg
- 推力：1814kN
- 比衝：330



引擎模型

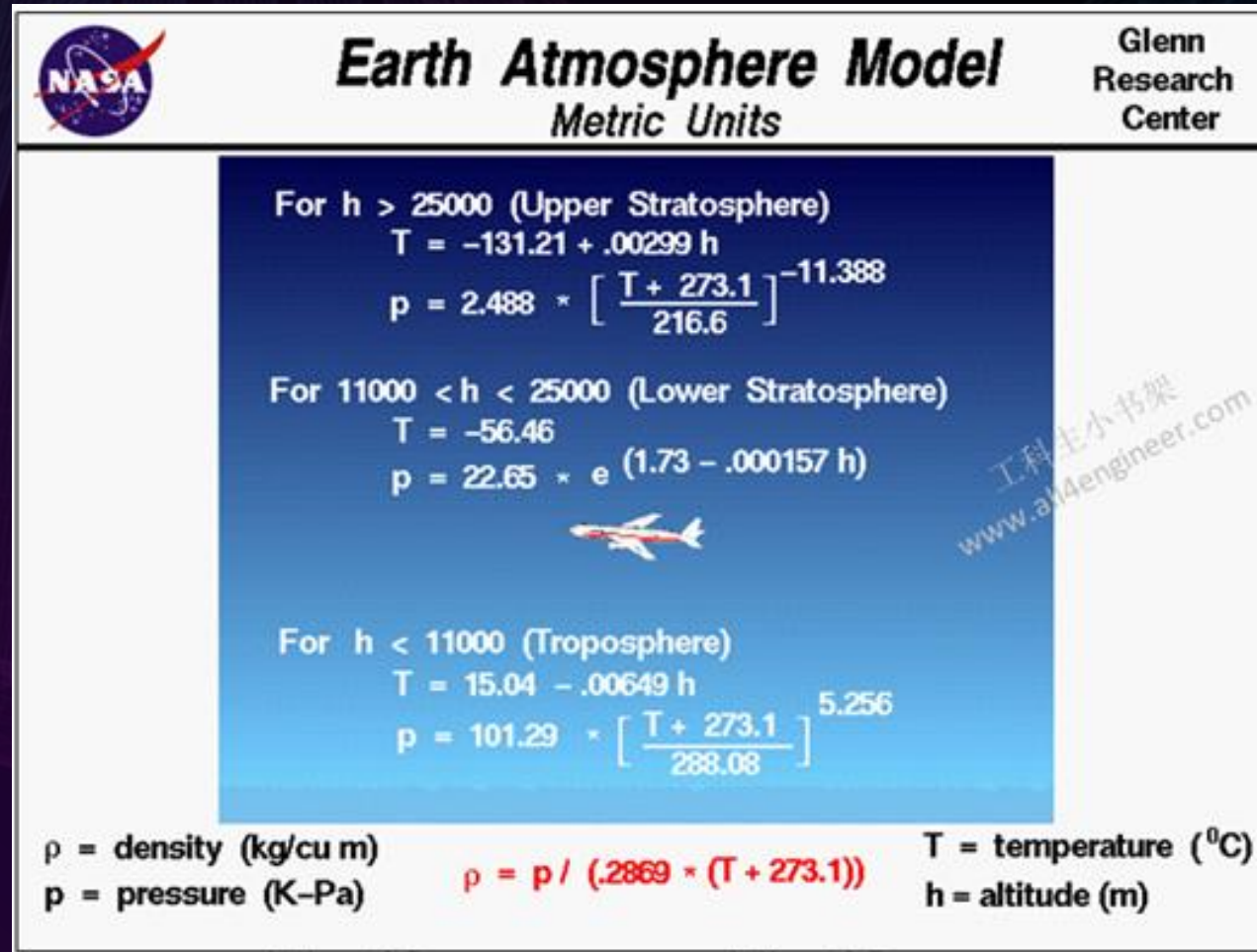
- 質量：275kg
- 推力：66447N
- 比衝：330
- 加速時間：
 $1025 * 9.8 * 330 / 66447 = 49.89s$
- 1025*9.8牛頓重燃料產生
66447N推力可持續約50s

換算方式：

雄三飛彈推力 = 猛禽火箭推力X噴氣口面積比X燃燒室體積比

參數設定

空氣密度



程式講解

設定常數

```
You, now | 2 authors (Andy Li and others)
1  from vpython import *
2  import math
3  import cv2
4
5  # 為了方便觀察將飛彈的形體放大10000倍
6  original_mass = 1500 #飛彈原本質量
7  radius = 10000*0.23 #飛彈半徑
8  length = 6.1 #飛彈長度
9  length_of_warhead = 10000*length*3/14.4 #彈頭長度
10 length_of_missile_body = 10000*length*11.4/14.4 #飛彈本體長度
11 cross_section = radius**2*pi/100000000 #飛彈橫截面
12 volume = 1/3*(cross_section*length_of_warhead+cross_section*length_of_missile_body)/10000 #飛彈體積
13 pushing_force = 1814*1000*(275/2000)**(5/3) #推進力量值
14 acceleration_time = 1025*9.8*330/pushing_force #加速時間
15
```

You, now • Uncommitted changes

設定常數

```
mass_of_earth = 5.9722*10**24 #地球質量
radius_of_earth = 6.371 * 10**6 #地球半徑
gravitational_constant = 6.6743 * 10 **(-11) #重力常數
longitude = pi*121/180#東經
latitude = pi*23.5/180#北緯

#座標系
origin_x = radius_of_earth * cos(latitude) * cos(longitude)
origin_y = radius_of_earth * sin(latitude)
origin_z = -radius_of_earth * cos(latitude) * sin(longitude)

drag_force_constant = 0.5 #拖曳力常數

center_of_earth = vector(0,0,0) #地球球心
rocket = vector(origin_x,origin_y,origin_z) #火箭座標
theta = 76.5*pi/180#火箭發射仰角
deviated_angle=pi/2#火箭發射方向，從x軸向逆時針旋轉
```


設定常數

```
#計算火箭發射位置的坐標系
x = norm(cross(vector(0,1/sin(latitude)*radius_of_earth,0),rocket))
y = norm(cross(rocket,x))
z = norm(cross(x,y))
v = norm(norm(cos(deviated_angle)*x+sin(deviated_angle)*y)+z*tan(theta))#火箭發射方向
h = 0 #火箭離地表高度
omega = vector(0,2*pi/86400,0) #地球自轉角速度

g = 9.8#重力加速度

#模擬火箭發射
scene = canvas(width=500, height=500, center=vec(0, -0.2, 0), background=vec(0.8,0.8,0.8),align="left")
earth=sphere(pos=vec(0,0,0),radius=radius_of_earth, texture={'file':textures.earth})
sim=sphere(pos=rocket,radius=0.001*radius_of_earth,make_trail=True)#標示起始發射點
scene.center=rocket
```

計算受力

```
def mass(t):#燃料燃燒，質量隨時間降低
    if t < acceleration_time:
        return original_mass-1025*t/acceleration_time
    else:#燃燒停止
        return 475

def rho(h):#計算空氣密度
    if h > 25000:
        T = -131.21 + 0.00299*h
        p = 2.488 * ((T+273.1)/216.6)**(-11.388)
    elif 25000 >= h > 11000:
        T = -56.46
        p = 22.65*e**(1.73-0.000157*h)
    else:
        T = 15.04-0.00649*h
        p = 101.29*((T+273.1)/288.08)**(5.256)
    rho = p/(0.2869*(T+273.1))
    return rho
```

計算受力

```
def push(direction):#計算火箭推力
    push = pushing_force*norm(direction)
    return push

def lift(rho,vec):#計算浮力
    return rho*volume*g*norm(vec)

def drag_force(rho,v):#計算拖曳力
    return -0.5*rho*drag_force_constant*mag2(v)*cross_section*norm(v)

def Coriolis_force(mass,omega,v):#計算科氏力
    return -2*mass*cross(omega,v)

def gravity(mass,h,vec):#計算重力
    return - gravitational_constant*mass_of_earth*mass/(h+radius_of_earth)**2*norm(vec)
```


座標系轉換

```
def trans(f_vec):#座標轉換成經緯度
    zero_vec=vector(1,0,0)
    new_vec=vector(f_vec.x,0,f_vec.z)
    cos_angle=dot(new_vec,zero_vec)/(mag(new_vec)*mag(zero_vec))
    latitude=asin(f_vec.y/radius_of_earth)*180/pi
    if f_vec.z<0:
        longitude=acos(cos_angle)*180/pi
        return "Eastern", longitude, latitude
    else:
        longitude=acos(cos_angle)*180/pi
        return "Western", longitude, latitude

t = 0
dt = 0.005
distance_traveled=0#火箭移動距離
projected_distance=0#火箭投影地球的移動距離
```

判斷飛彈是否落地

```
while mag(rocket-center_of_earth)-radius_of_earth >= -100:#火箭插入地球之前
    rate(10000)
    m = mass(t)#計算質量
    h = mag(rocket)-radius_of_earth#計算離地高度
    RHO = rho(h)#計算空氣密度
    F = lift(RHO,rocket) + drag_force(RHO,v) + Coriolis_force(m,omega,v) + gravity(m,h,rocket)#計算火箭受力
    if t < acceleration_time: F += push(v)#燃料燃燒殆盡前仍有推力
    a = F/m
    v += a * dt
    rocket += v * dt
    distance_traveled+=mag(v*dt)#ds
    # 計算投影
    cos_phi=dot(rocket,v*dt)/(mag(rocket)*mag(v*dt))
    sin_phi=math.sqrt(1-cos_phi**2)
    projected_distance+=sin_phi*mag(v*dt)
    scene.center=rocket
    t += dt

    rocket_3D.pos=rocket
    rocket_3D.axis=norm(v)*length_of_missile_body
    rocket_head.pos=rocket+norm(v)*length_of_missile_body
    rocket_head.axis = norm(v)*length_of_warhead
```

顯示結果

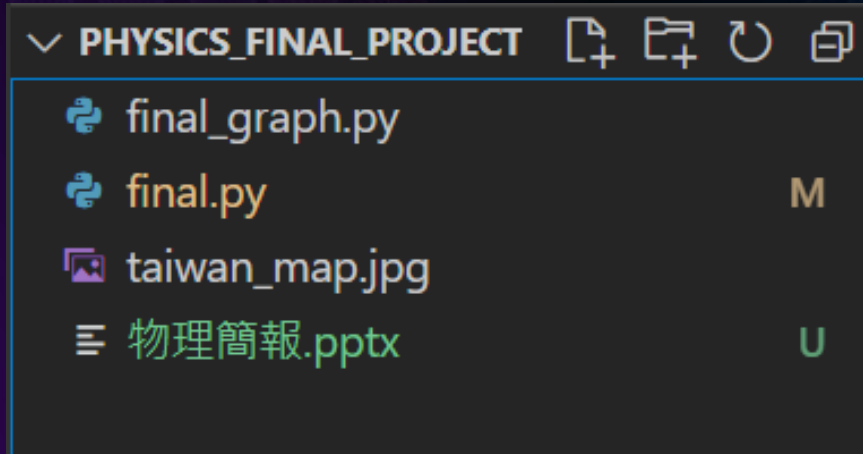
○

```
we,new_long,new_lat=trans(rocket)
# scene.center = rocket
print(acceleration_time)
print("time:",t)
print("distance traveled(km): ",round(distance_traveled/1000,2))#總射程
print("distance traveled(projected on the surface of earth)(km): ",round(projected_distance/1000,2))
print("original position: ",(origin_x,origin_y,origin_z))
print("final position: ",rocket)
print(f'longitude = {new_long}, latitude = {new_lat}')
```


繪畫飛彈落地點

```
img = cv2.imread('taiwan_map.jpg')
rows,cols,ch = img.shape #寬671km，長799km，附近經度1度 = 101km，附近緯度1度 = 111km
position = (int(img.shape[1]//2+(new_long-121)*101/671*img.shape[1]),int(img.shape[0]//2-(new_lat-23.5)*111/799*img.shape[0]))
cv2.circle(img,position,5,(0,0,255),thickness=-1)
cv2.namedWindow('image')
cv2.imshow('image',img)#顯示落點圖
cv2.waitKey(0)
```

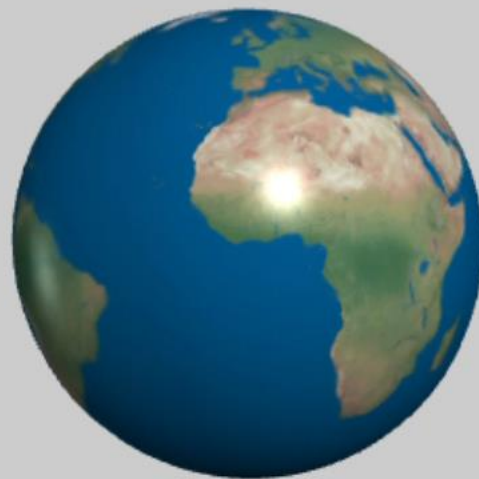
繪畫飛彈落地點



```
longitude = pi*121/180#東經  
latitude = pi*23.5/180#北緯
```

模擬結果分析

飛彈飛行動畫



飛彈飛行結果

```
time: 153.73999999993268  
distance traveled(km): 141.28  
distance traveled(projected on the surface of earth)(km): 133.81  
original position: (-3009156.1690410296, 2540430.3181227436, -5008076.871323472)  
final position: <-3.12302e+06, 2.53934e+06, -4.9383e+06>  
longitude = 122.3095799329062, latitude = 23.489345501677562
```

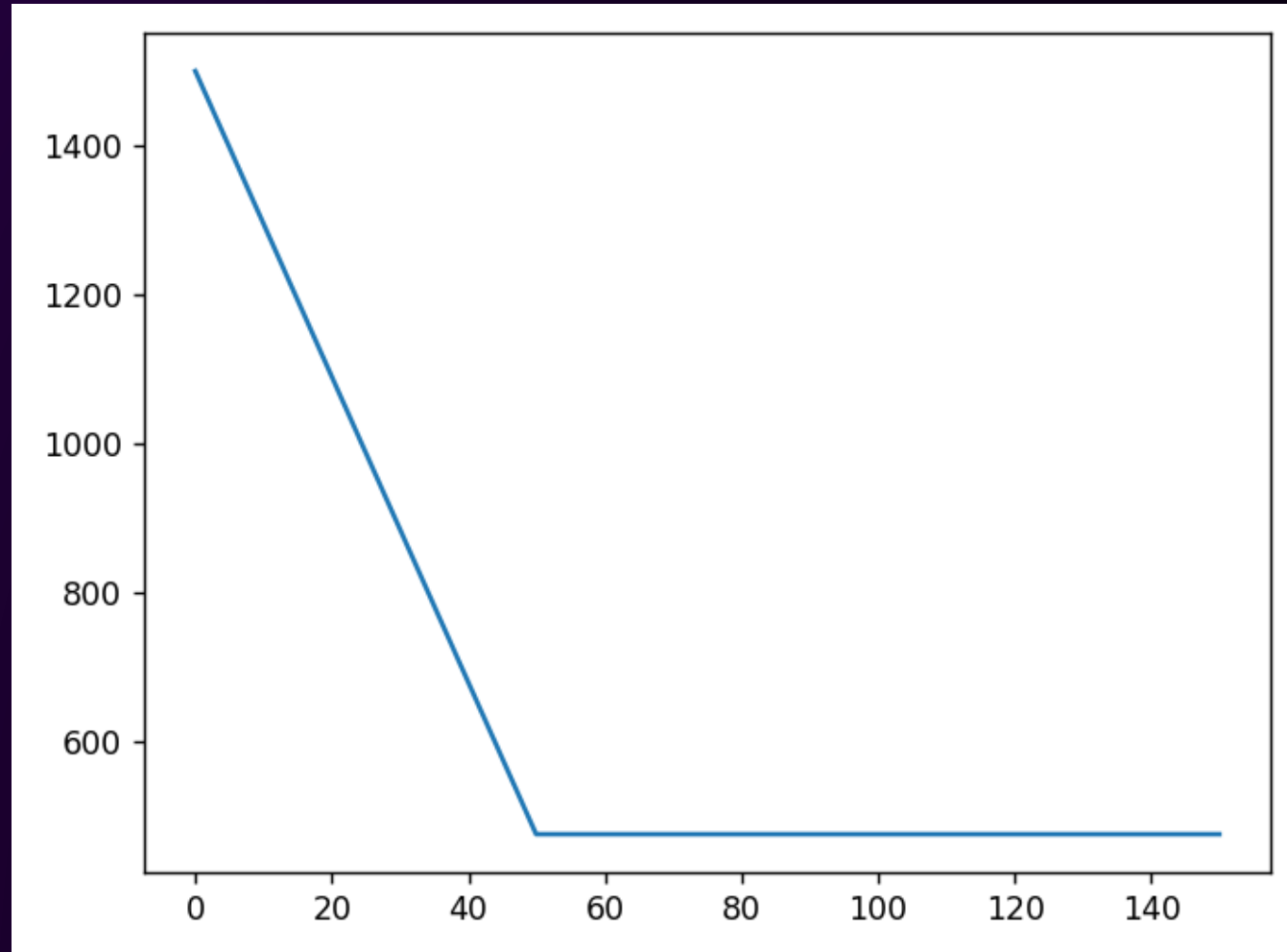
飛彈落點



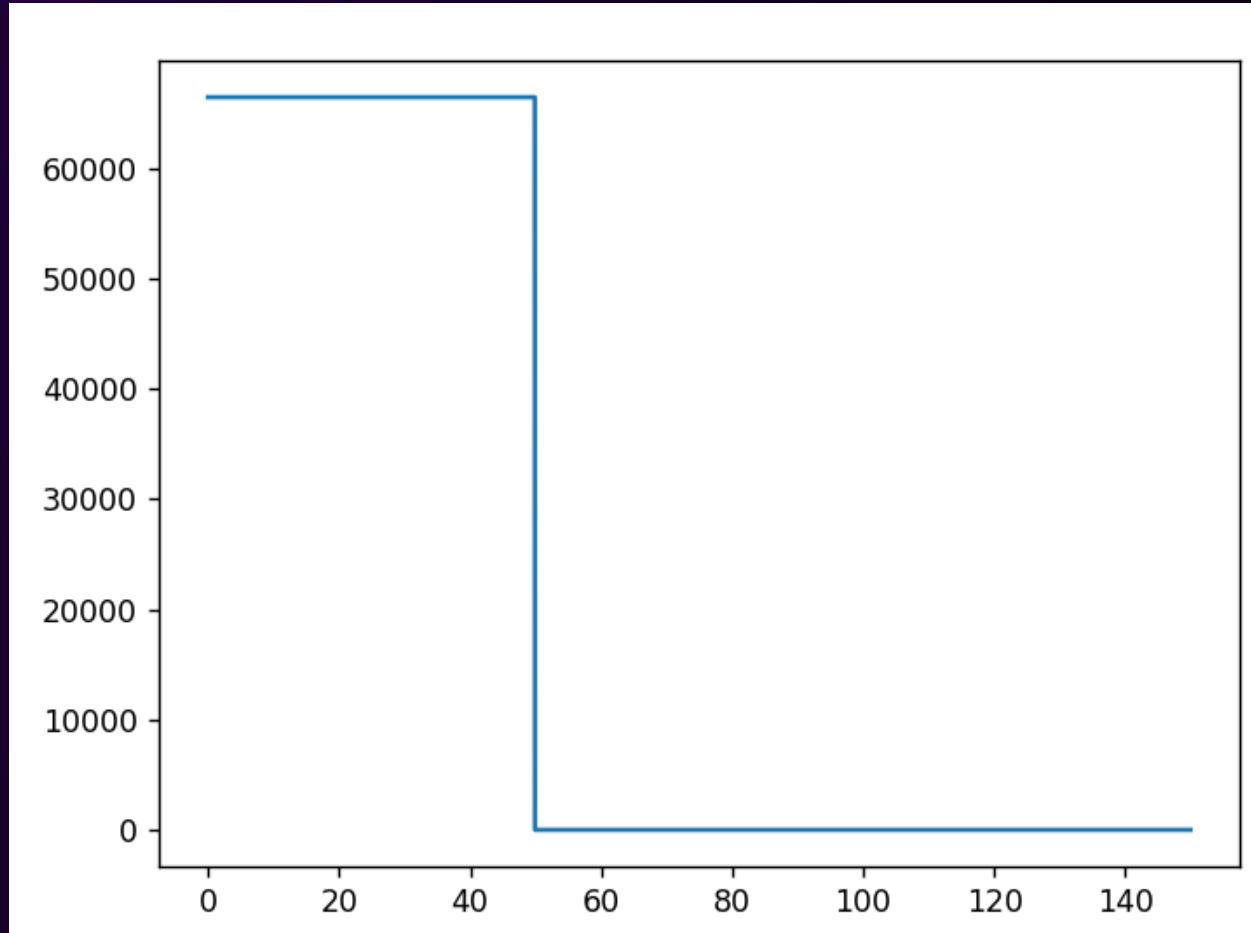
飛彈射程範圍



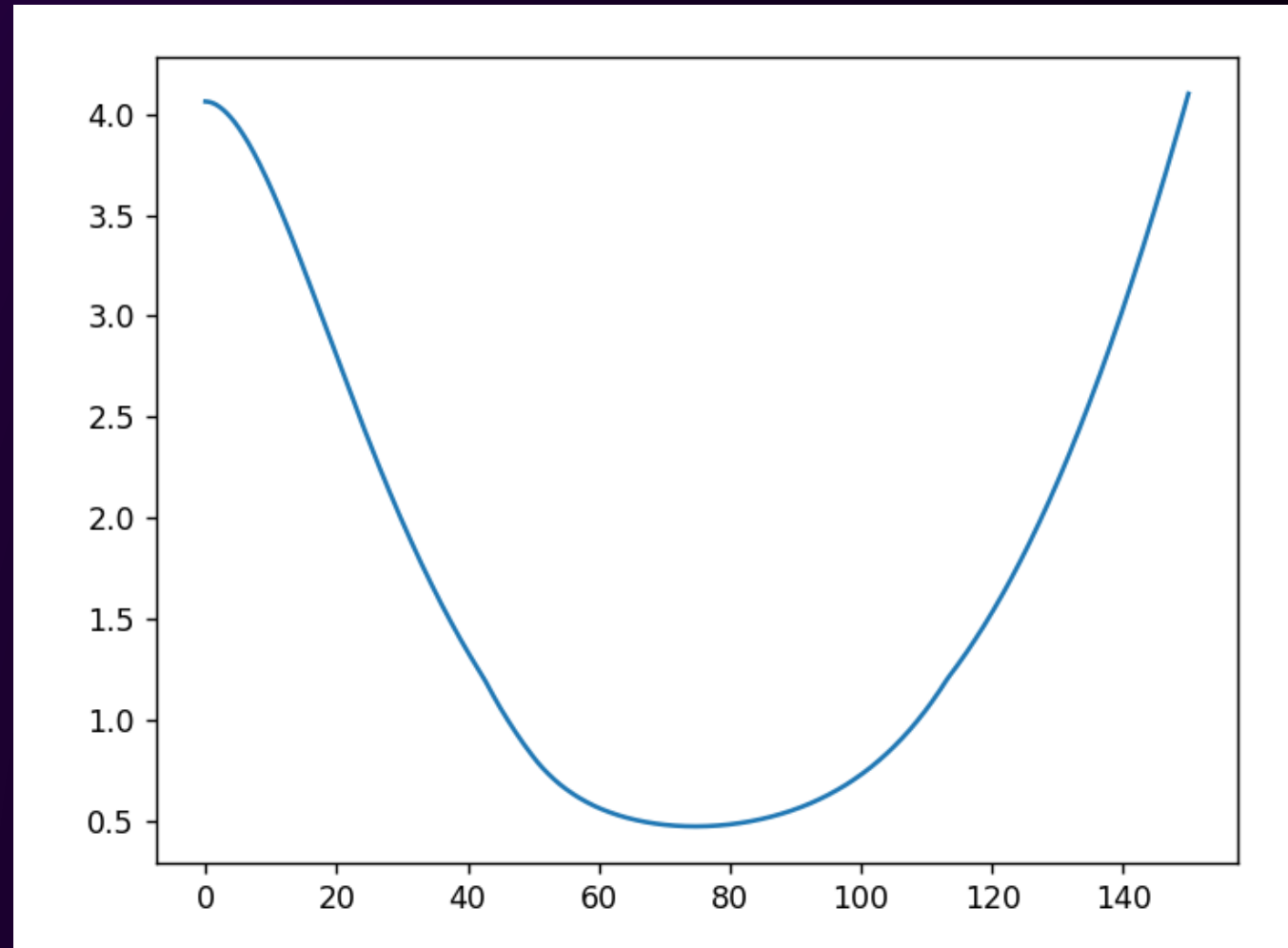
質量對時間的關係



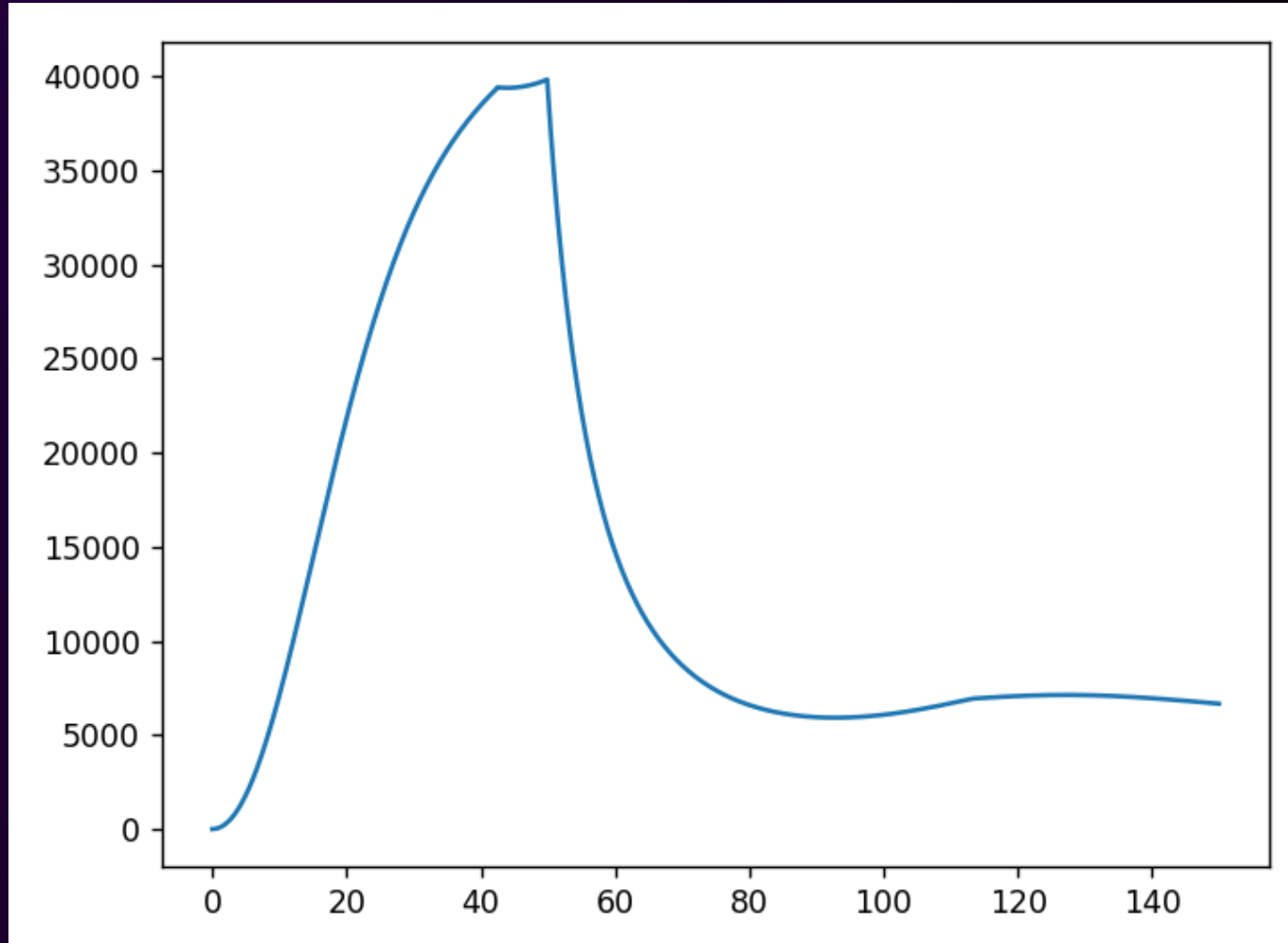
推力對時間的關係



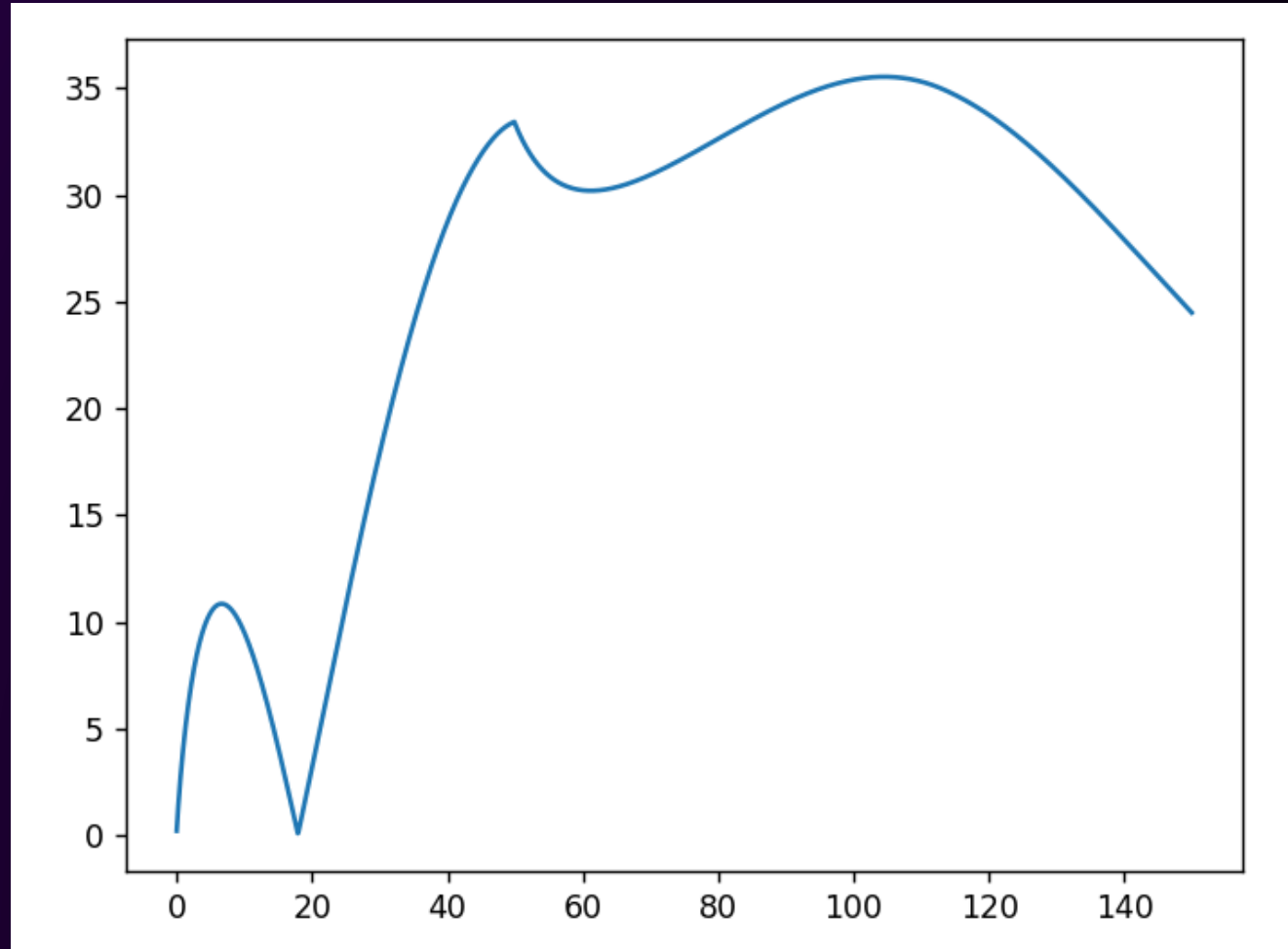
升力對時間的關係



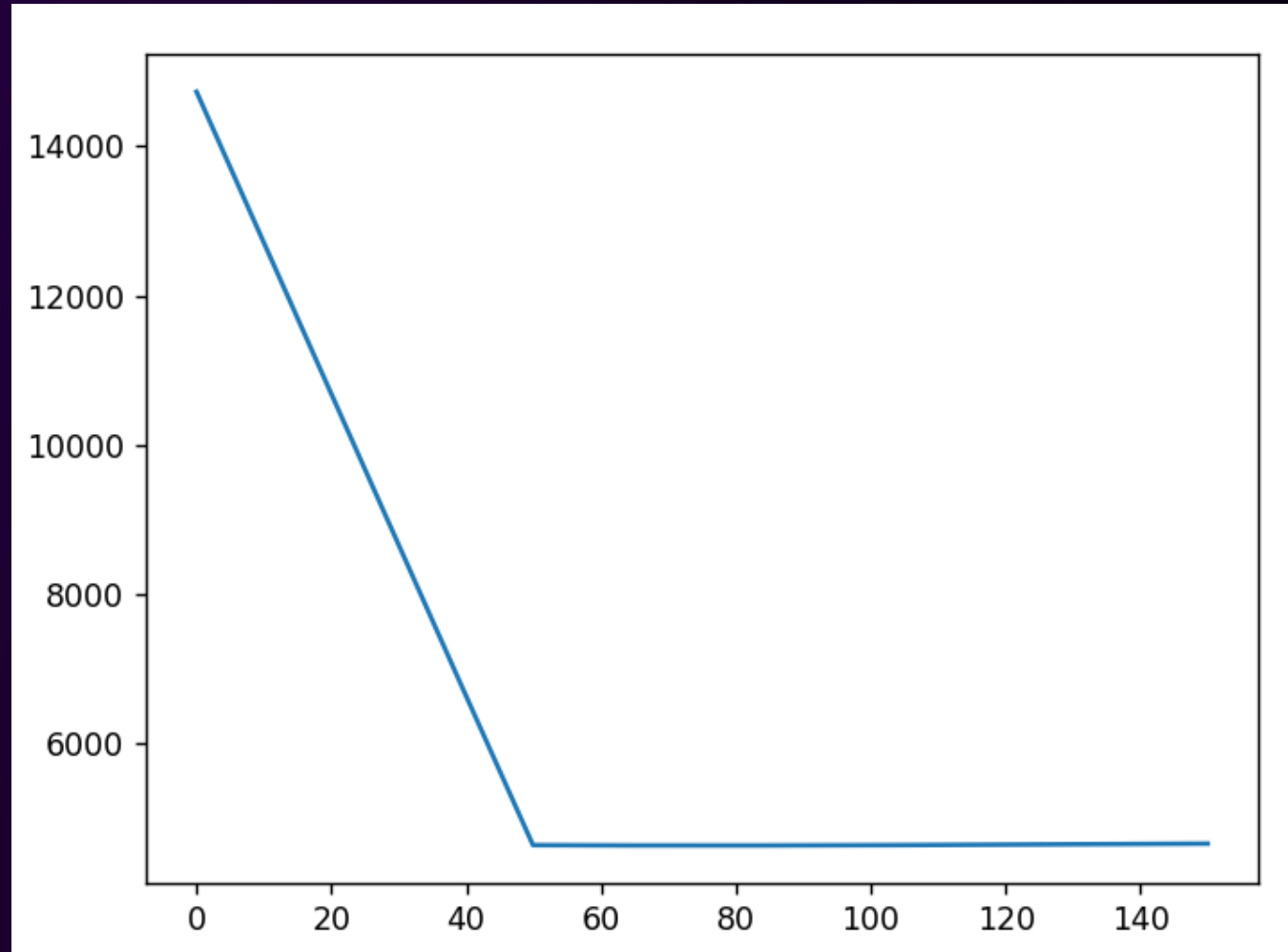
拖曳力對時間的關係



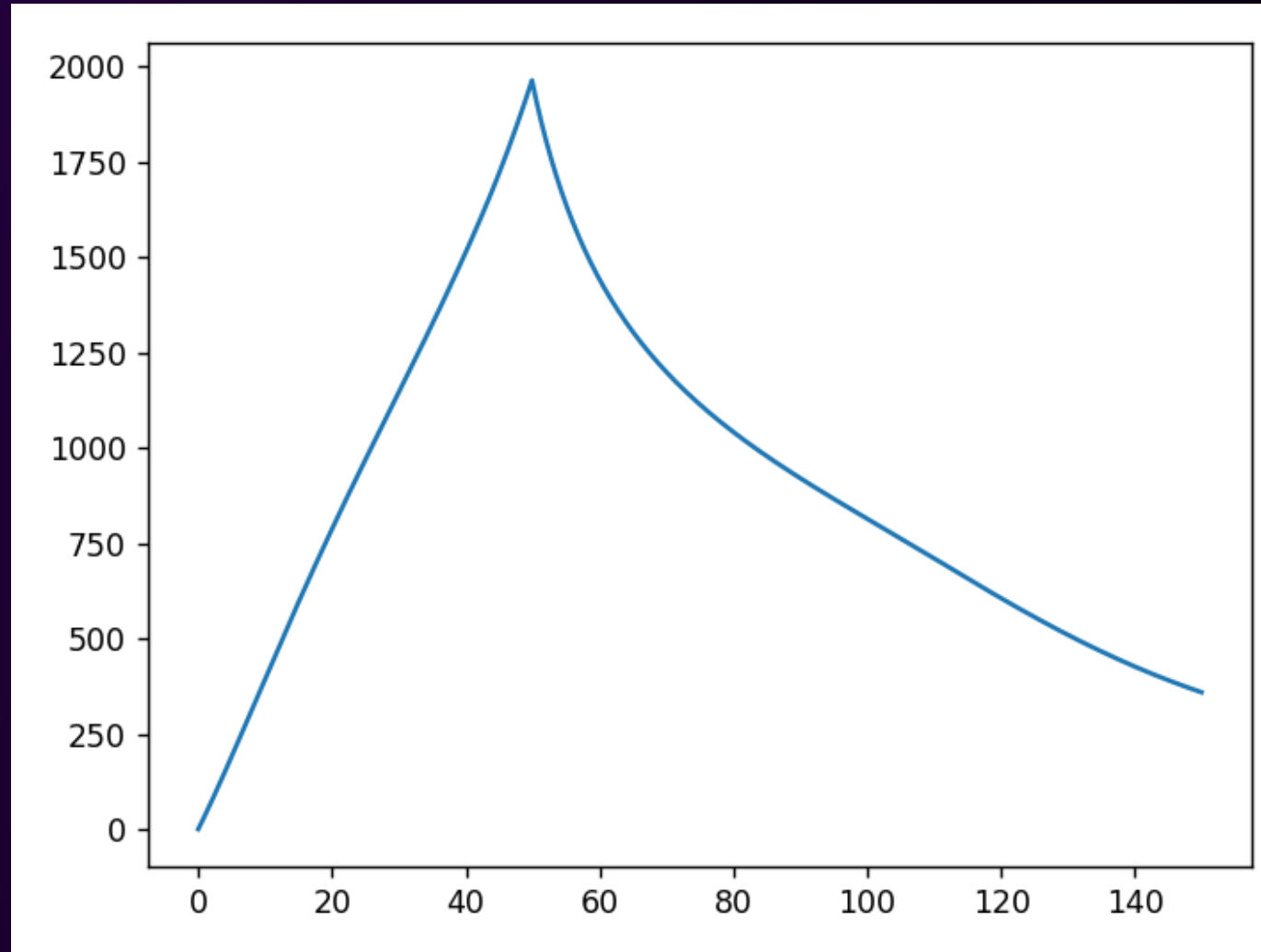
科氏力對時間的關係



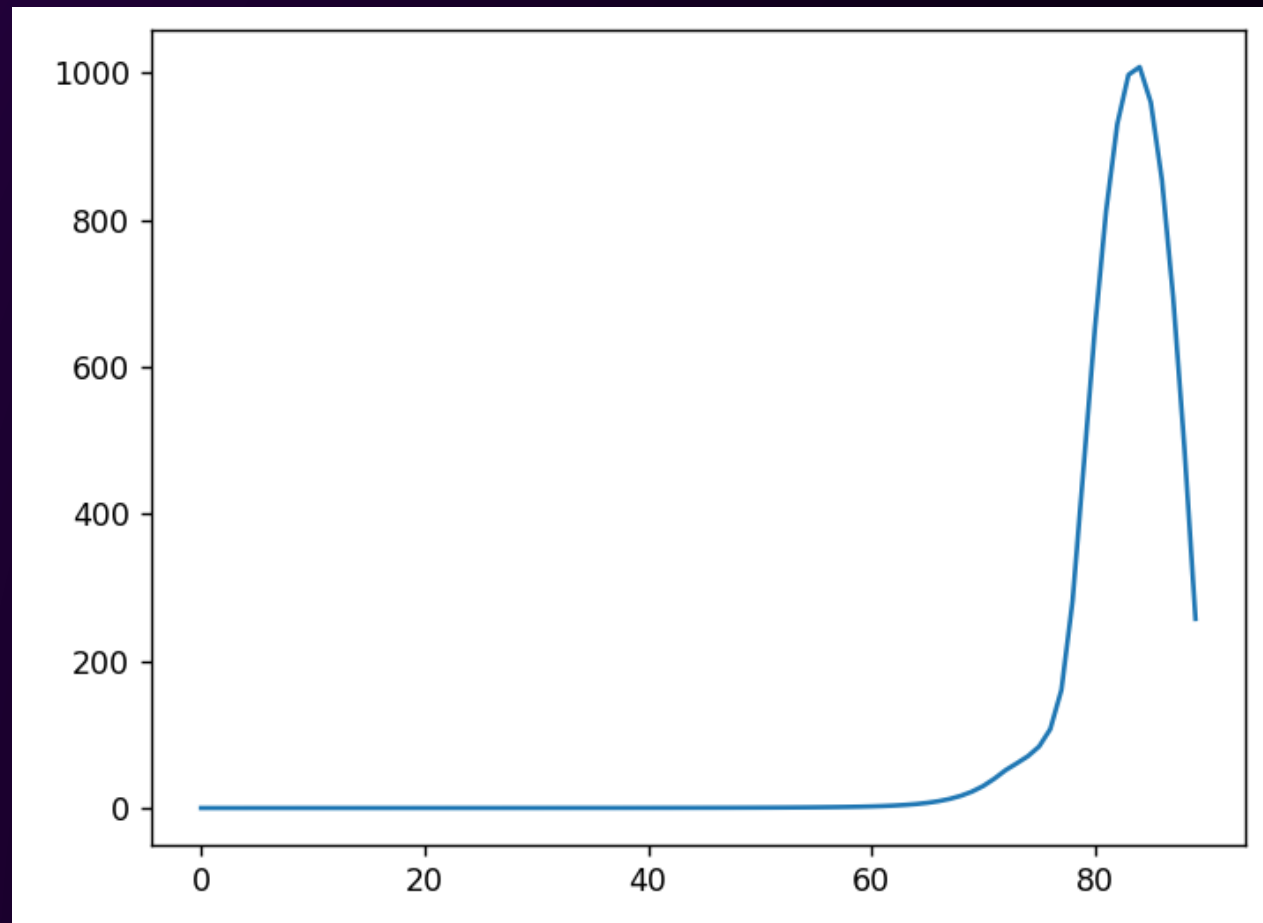
重力對時間的關係



速率對時間的關係



飛行距離(投影至地球)對角度的關係



參考資料

1. 行政院國家科學委員會補助專題研究計畫-飛彈攔截系統之設計及性能評估

[https://ir.nctu.edu.tw/bitstream/11536/102835/3/972221E009046MY3\(%E7%AC%AC3%E5%B9%B4\).PDF](https://ir.nctu.edu.tw/bitstream/11536/102835/3/972221E009046MY3(%E7%AC%AC3%E5%B9%B4).PDF)

2. 火箭方程

<https://zh.wikipedia.org/zh-tw/%E7%81%AB%E7%AE%AD%E6%96%B9%E7%A8%8B>

3. 空氣阻力係數

<https://zh.m.wikipedia.org/zh-tw/%E9%98%BB%E5%8A%9B%E4%BF%82%E6%95%B8>

4. 猛禽發動機

<https://zh.m.wikipedia.org/zh-tw/%E7%8C%9B%E7%A6%BD%E7%81%AB%E7%AE%AD%E7%99%BC%E5%8B%95%E6%A9%9F>

5. 阻力

<https://zh.m.wikipedia.org/zh-tw/%E9%98%BB%E5%8A%9B>

參考資料

6. 雄三飛彈 <https://zh.m.wikipedia.org/zh-tw/%E9%9B%84%E9%A2%A8%E4%B8%89%E5%9E%8B%E5%8F%8D%E8%89%A6%E9%A3%9B%E5%BD%88>

7. NASA 空氣壓力

<https://reurl.cc/WqAvk5>

8. 雄三飛彈介紹

http://211.75.138.103/formosa/blog_echo.php?TopicID=2475

9. Google Map

<https://www.google.com.tw/maps/place/23%C2%B030'00.0%22N+121%C2%B000'00.0%22E/@23.550292,119.3454887,7.83z/data=!4m5!3m4!1s0x0:0x797d62877d113945!8m2!3d23.5!4d121?hl=zh-TW&authuser=0>