Charles Wszalek **EGR 310**

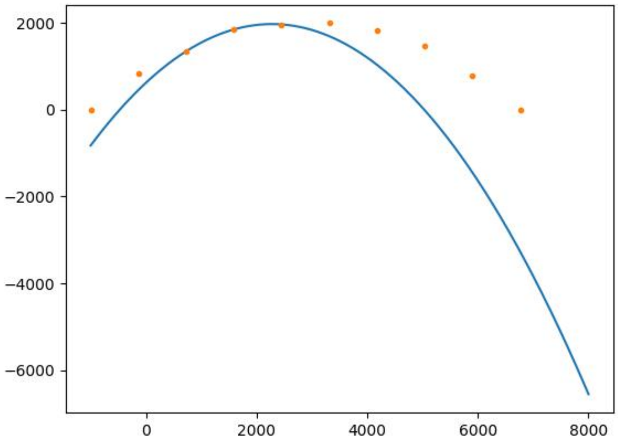
PYTHON CODE

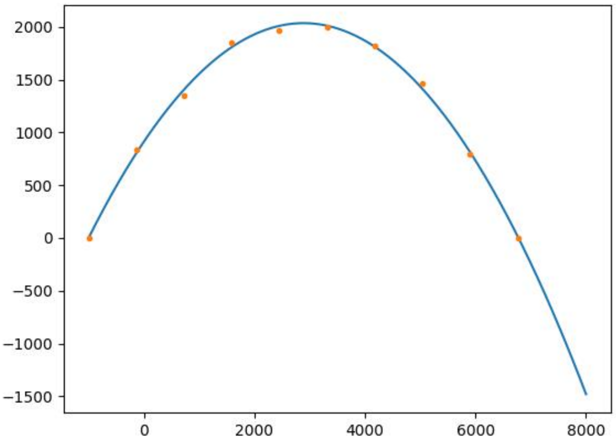
```
import numpy as np
import matplotlib.pyplot as plt
import scipy.optimize
from lib.Header import *


points = [[-1000, 0],
          [-137, 837],
          [725, 1349],
          [1588, 1853],
          [2451, 1969],
          [3313, 1998],
          [4176, 1820],
          [5039, 1468],
          [5901, 799],
          [6764, 0]]

[[x1, b1],
 [x2, b2],
 [x3, b3]] = [[725, 1349],
              [1588, 1853],
              [2451, 1969]]

xrange = np.linspace(-1000,8000,8000)

def L1(x):
    return ((x-x2) * (x-x3)) / ((x1-x2) * (x1-x3))

def L2(x):
    return ((x-x1) * (x-x3)) / ((x2-x1) * (x2-x3))

def L3(x):
    return ((x-x1) * (x-x2)) / ((x3-x1) * (x3-x2))

def ylag(x):
    return L1(x) * b1 + L2(x) * b2 + L3(x) * b3

plt.figure()
plt.plot(xrange, ylag(xrange))
plt.plot([coord[0] for coord in points], [coord[1] for coord in points], '.')
SAVE(1)
plt.show()

'''
# scipy.optimize.curve_fit allows you to put in a fit function and adjusts parameters
def fitfn(x, a, b, c):
    return a*x**2 + b*x + c
```

```
[[AA, BB, CC], covar] = scipy.optimize.curve_fit(fitfn, [coord[0] for coord in points], [coord[1] for coord in poi
'''
params = np.polyfit([coord[0] for coord in points], [coord[1] for coord in points], 2)

plt.figure()
# plt.plot(xrange, fitfn(xrange, AA, BB, CC))
plt.plot(xrange, np.polyval(params, xrange))
plt.plot([coord[0] for coord in points], [coord[1] for coord in points], '.')
SAVE(2)
plt.show()

PDF("Radar_Tracking.py", "Radar_Tracking.pdf")
```

OUTPUT

Plots





Prints