Original Code - Do Not Change

In [142]:

```python
import numpy as np
import matplotlib.pyplot as plt

# a basic rk4 routine implemented in octave
#  based on an rk4 routine from:
# https://math.okstate.edu/people/yqwang/teaching/math4513_fall11/Notes/rungekutta.pdf

# the code has been modified to use arbitrary function names, return
# output in an array, and have vectors of an arbitrary length for
# its input



def projectile(time, p):
    x = p[0]
    z = p[1]
    vx = p[2]
    vz = p[3]

    # This is a 2trick to set the projectile constants as initial conditions
    coeff = p[4]

    drv = np.zeros(5, np.float)

    # the velocities are the derviative of position
    drv[0] = vx
    drv[1] = vz


    # calcualte the acceleration
    g = -9.8  # m/s - gravity

    ax = -coeff * vx * vx * np.sign(vx)
    az = g  - np.sign(vz) * coeff * vz*vz


    # update the change vector
    drv[2] = ax
    drv[3] = az

    # we don't actually update the projectile constants since they are constant
    drv[4] = 0
```

```
    return drv

def myrungekutta(h, t, nsteps, y0):
    # rk4 routine
    # inputs:
    #   h      - stepsize (dt)
    #   t      - starting time for the integration
    #   nsteps - number of steps to take during the integration
    #   y0     - initial conditions for the equation
    #
    #  outputs:
    #  output:  an array of the output values of system
    y = y0
    # find the number of values in the initial conditions array
    ny = len(y)
    output = np.zeros([nsteps,ny+1], np.float)

    # loop over the array and calculate the position using an rk4
    # ODE integration routine
    for i in range(nsteps):
        k1 = h*projectile(t,y)
        k2 = h*projectile(t+h/2, y+k1/2)
        k3 = h*projectile(t+h/2, y+k2/2)
        k4 = h*projectile(t+h, y+k3)
        y = y + (k1 + 2*(k2+k3) + k4)/6
        t = t + h
        output[i,0] = t
        output[i,1:] = y[:]

    return output

##################

##################
# integration parameters

# initial position and air resistance
x =0
z =0
coeff = 0.005

# inputs are
```

```
#  v- initial velocity of the projectile
#  theta - the angle above the ground
#  h = step size - generally below one
#  tfinal - final time in the simulation

v = float( input("initial velocity? "))
theta = float(input("initial angle? "))
h = float(input("timestep h? "))
tfinal = float(input("t final? "))

#v, theta, h, tfinal = input()
vx = np.cos(theta * np.pi / 180.) * v
vz = np.sin(theta * np.pi / 180.) * v

# number of steps
nsteps = int(tfinal / h) + 1

# initial conditions for a circular projectile of unit size
t = 0.0
p0 = [x, z, vx, vz, coeff]

# define the function we are going to integrate - set up a pointer with the
# appropriate name
```

```
initial velocity? 120
initial angle? 4
timestep h? 0.01
t final? 10
```

In [143]:

```
# actually do the integration
o = myrungekutta(h, t, nsteps, p0)
```

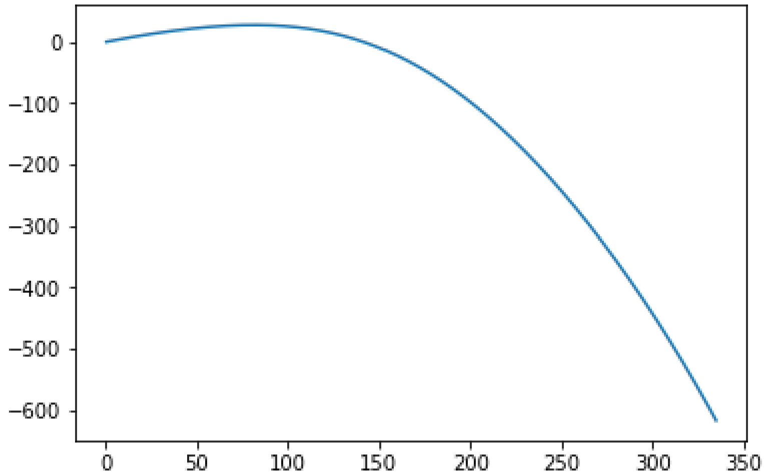
In [145]:

```
o[5]
```

Out[145]:

```
array([6.00000000e-02, 7.05649867e+00, 4.84005977e-01, 1.15557741e+02,
       7.76324581e+00, 5.00000000e-03])
```

In [9]:
```
# optional plotting routines
plt.plot( o[:,1], o[:,2])
plt.show()
```



In [10]:
```
o
```

Out[10]:
```
array([[ 1.00000000e-02,  4.32544627e-01,  2.49354288e-01,
          4.32077227e+01,  2.48709112e+01,  5.00000000e-03],
       [ 2.00000000e-02,  8.64155799e-01,  4.97419292e-01,
          4.31145786e+01,  2.47421430e+01,  5.00000000e-03],
       [ 3.00000000e-02,  1.29483753e+00,  7.44198207e-01,
          4.30218352e+01,  2.46136930e+01,  5.00000000e-03],
       ...,
       [ 1.99900000e+01,  3.34593759e+02, -6.15956972e+02,
          8.12717331e+00, -4.42364059e+01,  5.00000000e-03],
       [ 2.00000000e+01,  3.34675014e+02, -6.16399337e+02,
          8.12387210e+00, -4.42365625e+01,  5.00000000e-03],
       [ 2.00100000e+01,  3.34756236e+02, -6.16841703e+02,
          8.12057358e+00, -4.42367185e+01,  5.00000000e-03]])
```

Modified Code

In [146]:

```
import numpy as np
import matplotlib.pyplot as plt
import time             # added time. Used in testing and gave me an idea of runtime

# a basic rk4 routine implemented in octave
#  based on an rk4 routine from:
# https://math.okstate.edu/people/yqwang/teaching/math4513_fall11/Notes/rungekutta.pdf

# the code has been modified to use arbitrary function names, return
# output in an array, and have vectors of an arbitrary length for
# its input



def projectile(time, p):
    x = p[0]
    z = p[1]
    vx = p[2]
    vz = p[3]

    # This is a 2trick to set the projectile constants as initial conditions
    coeff = p[4]

    drv = np.zeros(5, np.float)

    # the velocities are the derviative of position
    drv[0] = vx
    drv[1] = vz


    # calcualte the acceleration
    g = -9.8  # m/s - gravity

    ax = -coeff * vx * vx * np.sign(vx)
    az = g  - np.sign(vz) * coeff * vz*vz


    # update the change vector
    drv[2] = ax
    drv[3] = az

    # we don't actually update the projectile constants since they are constant
```

```
    drv[4] = 0

    return drv

def myrungekutta(h, t, nsteps, y0):
    # rk4 routine
    # inputs:
    #   h      - stepsize (dt)
    #   t      - starting time for the integration
    #   nsteps - number of steps to take during the integration
    #   y0     - initial conditions for the equation
    #
    #  outputs:
    #  output:  an array of the output values of system
    y = y0
    # find the number of values in the initial conditions array
    ny = len(y)
    output = np.zeros([1,ny+1], np.float)
    YCheck = 'False'

    # loop over the array and calculate the position using an rk4
    # ODE integration routine
    for i in range(nsteps):
        while YCheck == 'False':
            k1 = h*projectile(t,y)
            k2 = h*projectile(t+h/2, y+k1/2)
            k3 = h*projectile(t+h/2, y+k2/2)
            k4 = h*projectile(t+h, y+k3)
            y = y + (k1 + 2*(k2+k3) + k4)/6
            t = t + h

            output[i,0] = t
            output[i,1:] = y[:]

            if output[:,2][i] <= 0:     # added this to get only successful launches
                YCheck = 'True'

    return output

##################

##################
# integration parameters
```

```
# initial position and air resistance
x =0
z =0
coeff = 0.005
```

Testing: (Manual)

-------------------High Angle

initial velocity? 380

initial angle? 89

timestep h? 0.01

t final? 20

-------------------Middle Angle

initial velocity? 39

initial angle? 30

timestep h? 0.01

t final? 20

-------------------Low Angle

initial velocity? 191

initial angle? 1

timestep h? 0.01

t final? 20

-------------------Conclusions

It is safe to say that the angle is between 0 and 90. Found the velocities needed for 1 and 89.

It might also be safe to say that the maximum velocity is 380 and t final is at most ~19.43

In [125]:
```
# actually do the integration
start_time = time.time()
o = myrungekutta(h, t, nsteps, p0)
end_time = time.time()

print("Runtime: " + str(end_time - start_time))
```

```
Runtime: 64.49391603469849
```

In [171]:

```
# enough math; do brute force

# inputs are
#  v- initial velocity of the projectile
#  theta - the angle above the ground
#  h = step size - generally below one
#  tfinal - final time in the simulation

h = float(0.0001)
tfinal = float(20.00)

# number of steps; irrelevant. choose something small
nsteps = int(tfinal / h) + 1

t = 0.0

aResults = []
xResults = []
tResults = []

start_time = time.time()
for a in range(1, 90):
    for xv in np.arange(1, 381, 0.25):
        theta = a
        v = xv

        #v, theta, h, tfinal = input()

        vx = np.cos(theta * np.pi / 180.) * v
        vz = np.sin(theta * np.pi / 180.) * v

        # initial conditions for a circular projectile of unit size

        p0 = [x, z, vx, vz, coeff]

        o = myrungekutta(h, t, nsteps, p0)

        if o[:,1] >= 99 and o[:,1] <= 100:
            print("Success! Velocity = " + str(v) + " Angle = " + str(theta) + " Time = " + str(o[:,0]) + " X
= " + str(o[:,1]) + " Y = " + str(o[:,2]) + " Current Runtime = " + str(round((time.time() - start_time), 2
)))
            xResults.append(v)
```

```
            aResults.append(theta)
            tResults.append(float(o[:,0]))
            break
print("\n Total Runtime: " + str(time.time() - start_time))
```

```
Success! Velocity = 190.0 Angle = 1 Time = [0.6758] X = [99.17230993] Y = [-6.68784922e-05] Current Runtime =
88.14
Success! Velocity = 134.5 Angle = 2 Time = [0.9553] X = [99.18883566] Y = [-9.40040729e-05] Current Runtime =
171.6
Success! Velocity = 110.0 Angle = 3 Time = [1.17] X = [99.25827496] Y = [-0.00018385] Current Runtime = 256.8
7
Success! Velocity = 95.25 Angle = 4 Time = [1.3485] X = [99.01956172] Y = [-0.00040665] Current Runtime = 33
8.04
Success! Velocity = 85.5 Angle = 5 Time = [1.5102] X = [99.32348357] Y = [-0.0002058] Current Runtime = 419.5
3
Success! Velocity = 78.25 Angle = 6 Time = [1.6553] X = [99.43720193] Y = [-0.00022122] Current Runtime = 49
9.82
Success! Velocity = 72.5 Angle = 7 Time = [1.7856] X = [99.238448] Y = [-2.75466184e-05] Current Runtime = 58
1.02
Success! Velocity = 68.0 Angle = 8 Time = [1.9099] X = [99.31041773] Y = [-0.00040892] Current Runtime = 662.
16
Success! Velocity = 64.25 Angle = 9 Time = [2.0256] X = [99.26976873] Y = [-0.00097568] Current Runtime = 74
4.06
Success! Velocity = 61.25 Angle = 10 Time = [2.1403] X = [99.6099348] Y = [-0.0006279] Current Runtime = 825.
54
Success! Velocity = 58.5 Angle = 11 Time = [2.2431] X = [99.43281215] Y = [-0.00069716] Current Runtime = 90
7.82
Success! Velocity = 56.25 Angle = 12 Time = [2.3467] X = [99.61938165] Y = [-0.0008031] Current Runtime = 99
0.74
Success! Velocity = 54.25 Angle = 13 Time = [2.4451] X = [99.69814169] Y = [-5.60387193e-07] Current Runtime
= 1076.91
Success! Velocity = 52.25 Angle = 14 Time = [2.5294] X = [99.08273288] Y = [-0.00076599] Current Runtime = 11
63.5
Success! Velocity = 50.75 Angle = 15 Time = [2.6244] X = [99.33520903] Y = [-0.00087918] Current Runtime = 12
49.09
Success! Velocity = 49.25 Angle = 16 Time = [2.7085] X = [99.07711681] Y = [-0.00023803] Current Runtime = 13
36.54
Success! Velocity = 48.0 Angle = 17 Time = [2.7959] X = [99.14594544] Y = [-0.00100971] Current Runtime = 142
6.42
Success! Velocity = 47.0 Angle = 18 Time = [2.8887] X = [99.62338002] Y = [-0.00012346] Current Runtime = 151
6.1
Success! Velocity = 46.0 Angle = 19 Time = [2.9741] X = [99.76381006] Y = [-0.00104568] Current Runtime = 160
4.77
Success! Velocity = 45.0 Angle = 20 Time = [3.052] X = [99.58295802] Y = [-0.00101814] Current Runtime = 169
2.7
Success! Velocity = 44.0 Angle = 21 Time = [3.1226] X = [99.1030365] Y = [-0.00146766] Current Runtime = 178
3.39
Success! Velocity = 43.25 Angle = 22 Time = [3.2033] X = [99.21623805] Y = [-0.00078822] Current Runtime = 18
```

```
71.26
Success! Velocity = 42.5 Angle = 23 Time = [3.2783] X = [99.09276876] Y = [-0.00132333] Current Runtime = 196
0.85
Success! Velocity = 42.0 Angle = 24 Time = [3.3663] X = [99.64608268] Y = [-0.00159141] Current Runtime = 205
0.94
Success! Velocity = 41.25 Angle = 25 Time = [3.4305] X = [99.09157791] Y = [-0.00015589] Current Runtime = 21
40.43
Success! Velocity = 40.75 Angle = 26 Time = [3.5093] X = [99.26284091] Y = [-0.00127187] Current Runtime = 22
30.82
Success! Velocity = 40.25 Angle = 27 Time = [3.5838] X = [99.25964072] Y = [-0.00119791] Current Runtime = 23
21.3
Success! Velocity = 39.75 Angle = 28 Time = [3.6541] X = [99.09075028] Y = [-0.00066497] Current Runtime = 24
16.64
Success! Velocity = 39.5 Angle = 29 Time = [3.7421] X = [99.71828952] Y = [-0.00071581] Current Runtime = 251
6.39
Success! Velocity = 39.0 Angle = 30 Time = [3.8048] X = [99.24820765] Y = [-0.00012429] Current Runtime = 261
0.09
Success! Velocity = 38.75 Angle = 31 Time = [3.8865] X = [99.60338997] Y = [-0.00139509] Current Runtime = 27
08.52
Success! Velocity = 38.5 Angle = 32 Time = [3.9652] X = [99.82945448] Y = [-0.0001784] Current Runtime = 280
5.35
Success! Velocity = 38.25 Angle = 33 Time = [4.0412] X = [99.93569966] Y = [-0.00148776] Current Runtime = 29
05.16
Success! Velocity = 38.0 Angle = 34 Time = [4.1143] X = [99.9213167] Y = [-0.0010863] Current Runtime = 3008.
55
Success! Velocity = 37.75 Angle = 35 Time = [4.1846] X = [99.79129824] Y = [-0.00027821] Current Runtime = 31
22.25
Success! Velocity = 37.5 Angle = 36 Time = [4.2522] X = [99.55040351] Y = [-0.00044235] Current Runtime = 323
5.74
Success! Velocity = 37.25 Angle = 37 Time = [4.3171] X = [99.20136536] Y = [-0.00102486] Current Runtime = 33
46.3
Success! Velocity = 37.25 Angle = 38 Time = [4.4056] X = [99.7421113] Y = [-0.00029765] Current Runtime = 345
8.25
Success! Velocity = 37.0 Angle = 39 Time = [4.4656] X = [99.18644383] Y = [-0.00076536] Current Runtime = 357
1.57
Success! Velocity = 37.0 Angle = 40 Time = [4.5502] X = [99.5293834] Y = [-0.00204138] Current Runtime = 368
8.18
Success! Velocity = 37.0 Angle = 41 Time = [4.6329] X = [99.7730436] Y = [-0.00116374] Current Runtime = 380
7.17
Success! Velocity = 37.0 Angle = 42 Time = [4.7138] X = [99.91997093] Y = [-0.00023362] Current Runtime = 392
5.94
Success! Velocity = 37.0 Angle = 43 Time = [4.793] X = [99.97254815] Y = [-0.00146097] Current Runtime = 404
6.62
```

```
Success! Velocity = 37.0 Angle = 44 Time = [4.8703] X = [99.92815516] Y = [-0.00048661] Current Runtime = 416
9.06
Success! Velocity = 37.0 Angle = 45 Time = [4.9459] X = [99.79068883] Y = [-0.00169645] Current Runtime = 429
6.75
Success! Velocity = 37.0 Angle = 46 Time = [5.0196] X = [99.5575385] Y = [-0.00056507] Current Runtime = 442
3.66
Success! Velocity = 37.0 Angle = 47 Time = [5.0916] X = [99.23240259] Y = [-0.00155376] Current Runtime = 455
0.71
Success! Velocity = 37.25 Angle = 48 Time = [5.1915] X = [99.79032216] Y = [-0.0012901] Current Runtime = 468
0.29
Success! Velocity = 37.25 Angle = 49 Time = [5.2601] X = [99.27273217] Y = [-1.88047383e-05] Current Runtime
= 4806.92
Success! Velocity = 37.5 Angle = 50 Time = [5.3572] X = [99.62880028] Y = [-0.00118166] Current Runtime = 493
8.16
Success! Velocity = 37.75 Angle = 51 Time = [5.4529] X = [99.87832003] Y = [-0.00189889] Current Runtime = 50
71.49
Success! Velocity = 37.75 Angle = 52 Time = [5.5166] X = [99.0687906] Y = [-0.00018748] Current Runtime = 520
8.9
Success! Velocity = 38.0 Angle = 53 Time = [5.6093] X = [99.10924827] Y = [-0.00045745] Current Runtime = 535
5.72
Success! Velocity = 38.25 Angle = 54 Time = [5.7006] X = [99.0400579] Y = [-0.00213853] Current Runtime = 550
9.27
Success! Velocity = 38.75 Angle = 55 Time = [5.8212] X = [99.78046282] Y = [-0.00254628] Current Runtime = 56
73.91
Success! Velocity = 39.0 Angle = 56 Time = [5.9095] X = [99.47357547] Y = [-0.00168088] Current Runtime = 584
5.04
Success! Velocity = 39.25 Angle = 57 Time = [5.9963] X = [99.05104488] Y = [-0.00188443] Current Runtime = 60
10.26
Success! Velocity = 39.75 Angle = 58 Time = [6.1125] X = [99.39937552] Y = [-0.00220398] Current Runtime = 61
67.2
Success! Velocity = 40.25 Angle = 59 Time = [6.2271] X = [99.60250242] Y = [-0.00208808] Current Runtime = 63
32.45
Success! Velocity = 40.75 Angle = 60 Time = [6.34] X = [99.65595048] Y = [-0.00025473] Current Runtime = 650
8.27
Success! Velocity = 41.25 Angle = 61 Time = [6.4513] X = [99.55759486] Y = [-0.00083688] Current Runtime = 66
97.25
Success! Velocity = 41.75 Angle = 62 Time = [6.5609] X = [99.30264486] Y = [-0.00241366] Current Runtime = 68
91.83
Success! Velocity = 42.5 Angle = 63 Time = [6.6991] X = [99.70390042] Y = [-0.00184914] Current Runtime = 709
9.37
Success! Velocity = 43.0 Angle = 64 Time = [6.8049] X = [99.10508698] Y = [-0.00079903] Current Runtime = 731
1.66
Success! Velocity = 43.75 Angle = 65 Time = [6.939] X = [99.12147031] Y = [-0.00136176] Current Runtime = 752
```

```
9.71
Success! Velocity = 44.75 Angle = 66 Time = [7.1006] X = [99.69472264] Y = [-0.0023656] Current Runtime = 775
3.56
Success! Velocity = 45.5 Angle = 67 Time = [7.2298] X = [99.26417333] Y = [-0.00117972] Current Runtime = 798
3.05
Success! Velocity = 46.5 Angle = 68 Time = [7.3858] X = [99.33810408] Y = [-0.00111995] Current Runtime = 821
8.88
Success! Velocity = 47.5 Angle = 69 Time = [7.5388] X = [99.13979057] Y = [-0.00262444] Current Runtime = 845
0.29
Success! Velocity = 48.75 Angle = 70 Time = [7.7169] X = [99.34590366] Y = [-0.00130148] Current Runtime = 87
01.16
Success! Velocity = 50.0 Angle = 71 Time = [7.8911] X = [99.21621698] Y = [-0.00288541] Current Runtime = 897
1.29
Success! Velocity = 51.5 Angle = 72 Time = [8.0884] X = [99.3789065] Y = [-0.00096898] Current Runtime = 925
7.53
Success! Velocity = 53.0 Angle = 73 Time = [8.2807] X = [99.13137916] Y = [-0.00176109] Current Runtime = 955
4.35
Success! Velocity = 54.75 Angle = 74 Time = [8.4939] X = [99.05060668] Y = [-0.001716] Current Runtime = 988
0.14
Success! Velocity = 56.75 Angle = 75 Time = [8.7261] X = [99.04214223] Y = [-0.00111491] Current Runtime = 10
230.58
Success! Velocity = 59.0 Angle = 76 Time = [8.9753] X = [99.00507521] Y = [-0.00238003] Current Runtime = 106
08.83
Success! Velocity = 61.75 Angle = 77 Time = [9.2627] X = [99.34471671] Y = [-0.000961] Current Runtime = 1100
1.46
Success! Velocity = 64.75 Angle = 78 Time = [9.5608] X = [99.37182021] Y = [-0.00041144] Current Runtime = 11
425.05
Success! Velocity = 68.25 Angle = 79 Time = [9.8888] X = [99.41369344] Y = [-0.00190613] Current Runtime = 11
890.86
Success! Velocity = 72.25 Angle = 80 Time = [10.2412] X = [99.23593251] Y = [-0.00101656] Current Runtime = 1
2417.13
Success! Velocity = 77.25 Angle = 81 Time = [10.6509] X = [99.37259923] Y = [-0.00229878] Current Runtime = 1
2990.2
Success! Velocity = 83.25 Angle = 82 Time = [11.105] X = [99.34390827] Y = [-0.00076154] Current Runtime = 13
647.65
Success! Velocity = 90.75 Angle = 83 Time = [11.6234] X = [99.27473113] Y = [-0.0002263] Current Runtime = 14
414.93
Success! Velocity = 100.5 Angle = 84 Time = [12.2294] X = [99.22635869] Y = [-0.00271321] Current Runtime = 1
5345.75
Success! Velocity = 113.75 Angle = 85 Time = [12.954] X = [99.19853885] Y = [-0.00316546] Current Runtime = 1
6462.06
Success! Velocity = 132.75 Angle = 86 Time = [13.841] X = [99.04290389] Y = [-0.00137885] Current Runtime = 1
7887.42
```

```
Success! Velocity = 163.25 Angle = 87 Time = [14.9996] X = [99.03328437] Y = [-0.00372588] Current Runtime =
19882.4
Success! Velocity = 220.75 Angle = 88 Time = [16.6331] X = [99.02609854] Y = [-0.000711] Current Runtime = 22
934.8
Success! Velocity = 378.25 Angle = 89 Time = [19.4068] X = [99.00723705] Y = [-0.00093249] Current Runtime =
30392.06

 Total Runtime: 30392.064562797546
```

In [208]:

```
aResults
```

```
Out[208]: [1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 10,
 11,
 12,
 13,
 14,
 15,
 16,
 17,
 18,
 19,
 20,
 21,
 22,
 23,
 24,
 25,
 26,
 27,
 28,
 29,
 30,
 31,
 32,
 33,
 34,
 35,
 36,
 37,
 38,
 39,
 40,
 41,
 42,
 43,
```

```
44,
45,
46,
47,
48,
49,
50,
51,
52,
53,
54,
55,
56,
57,
58,
59,
60,
61,
62,
63,
64,
65,
66,
67,
68,
69,
70,
71,
72,
73,
74,
75,
76,
77,
78,
79,
80,
81,
82,
83,
84,
85,
86,
```

87,
88,
89]

In [209]:
```
xResults
```

```
Out[209]:  [190.0,
            134.5,
            110.0,
            95.25,
            85.5,
            78.25,
            72.5,
            68.0,
            64.25,
            61.25,
            58.5,
            56.25,
            54.25,
            52.25,
            50.75,
            49.25,
            48.0,
            47.0,
            46.0,
            45.0,
            44.0,
            43.25,
            42.5,
            42.0,
            41.25,
            40.75,
            40.25,
            39.75,
            39.5,
            39.0,
            38.75,
            38.5,
            38.25,
            38.0,
            37.75,
            37.5,
            37.25,
            37.25,
            37.0,
            37.0,
            37.0,
            37.0,
            37.0,
```

```
37.0,
37.0,
37.0,
37.0,
37.25,
37.25,
37.5,
37.75,
37.75,
38.0,
38.25,
38.75,
39.0,
39.25,
39.75,
40.25,
40.75,
41.25,
41.75,
42.5,
43.0,
43.75,
44.75,
45.5,
46.5,
47.5,
48.75,
50.0,
51.5,
53.0,
54.75,
56.75,
59.0,
61.75,
64.75,
68.25,
72.25,
77.25,
83.25,
90.75,
100.5,
113.75,
132.75,
```

```
163.25,
220.75,
378.25]
```

In [210]:
```
tResults
```

```
Out[210]: [0.6757999999999419,
 0.9552999999999111,
 1.1699999999998876,
 1.348499999999868,
 1.51019999999985,
 1.6552999999998341,
 1.7855999999998198,
 1.909899999999806,
 2.02559999999985,
 2.1403000000000922,
 2.243100000000309,
 2.346700000000528,
 2.4451000000007355,
 2.5294000000009134,
 2.624400000001114,
 2.7085000000012913,
 2.7959000000014758,
 2.8887000000016716,
 2.974100000001852,
 3.052000000002016,
 3.122600000002165,
 3.2033000000023355,
 3.2783000000024938,
 3.3663000000026795,
 3.430500000002815,
 3.5093000000029813,
 3.5838000000031385,
 3.654100000003287,
 3.7421000000034725,
 3.804800000003605,
 3.8865000000037773,
 3.9652000000039433,
 4.04120000000392,
 4.11430000000375,
 4.184600000003586,
 4.252200000003429,
 4.317100000003277,
 4.405600000003071,
 4.465600000002931,
 4.550200000002734,
 4.632900000002541,
 4.713800000002353,
 4.793000000002168,
```

```
4.870300000001988,
4.945900000001812,
5.01960000000164,
5.091600000001472,
5.1915000000012395,
5.26010000000108,
5.357200000000853,
5.45290000000063,
5.516600000000482,
5.609300000000266,
5.700600000000053,
5.821199999999772,
5.909499999999566,
5.996299999999364,
6.112499999999093,
6.227099999998826,
6.339999999998563,
6.451299999998303,
6.560899999998048,
6.699099999997726,
6.804899999997479,
6.938999999997167,
7.10059999999679,
7.229799999996489,
7.3857999999961255,
7.538799999995769,
7.716899999995354,
7.891099999994948,
8.088399999994488,
8.28069999999404,
8.493899999993543,
8.726099999993002,
8.975299999992421,
9.262699999991751,
9.560799999991056,
9.888799999990292,
10.24119999998947,
10.650899999988516,
11.104999999987458,
11.62339999998625,
12.229399999984837,
12.953999999983148,
13.840999999981081,
```

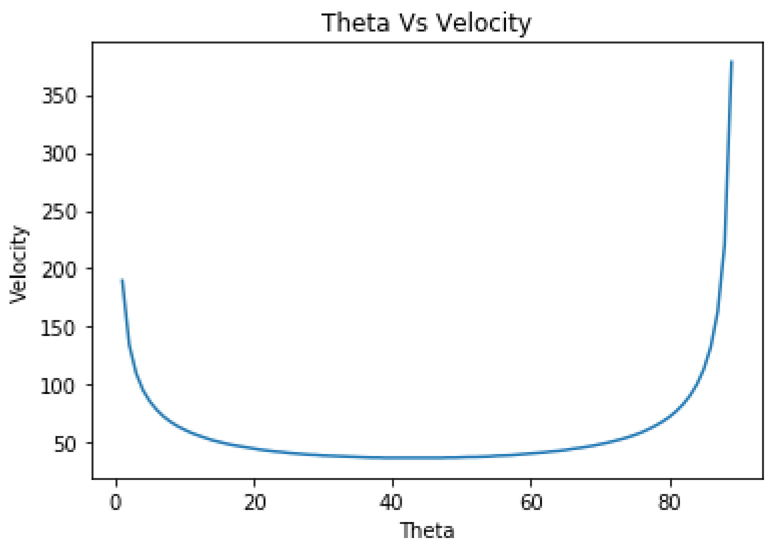
```
14.999599999978381,
16.633099999974576,
19.40679999996811]
```

In [214]:
```
with open("aResults.txt", "w") as file:
    for row in aResults:
        file.write('\n' + str(row))
```

In [215]:
```
with open("xResults.txt", "w") as file:
    for row in xResults:
        file.write('\n' + str(row))
```

In [216]:
```
with open("tResults.txt", "w") as file:
    for row in tResults:
        file.write('\n' + str(row))
```

In [179]:
```
plt.title('Theta Vs Velocity')
plt.ylabel('Velocity')
plt.xlabel('Theta')
plt.plot(aResults, xResults)
plt.savefig('ThetaV')
plt.show()
```



In [180]:
```
plt.title('Theta Vs Time')
plt.ylabel('Time')
plt.xlabel('Theta')
plt.plot(aResults, tResults)
plt.savefig('ThetaT')
plt.show()
```