A STATISTICAL ANALYSIS OF LAUNCHED PROJECTILES

Paul Bouthellier
Department of Computer Science and Mathematics
University of Pittsburgh-Greensburg
Greensburg, PA 15601
pbouthe@pitt.edu

When a projectile is launched, its angle of launch and initial velocity are actually probability distributions with given means and standard deviations. In this paper we shall study the corresponding trajectory of the projectile as a probability distribution.

As a historical application, we shall look at the accuracy of bombs dropped from WWII planes. The speed of the planes was often only known to be within 10-20 mph and the speed (and direction) of the wind was often estimated [4], [5], [6]. Considering: the speed of the planes, the winds, changes in atmospheric density, the coefficients of drag (given the nonsymmetric shape of the bomb) and the mass of the bombs, it will be shown that a given bomb dropped from 25,000-30,000 feet could miss its target by miles. This explains why it was often necessary to launch 1000 bomber raids against a given target to have any chance of hitting it.

In this talk, we will first create a mathematical model of a 3D trajectory of a projectile launched from a cannon or dropped from a plane. This model will consider the angle and velocity of launch, an atmospheric density model, a 3D vector field representation of the wind, the mass and coefficients of drag of the projectile.

The model will be given as a system of differential equations, and the solution will be approximated by numerical methods then implemented using a program written in Python. The program will be run over a range of values for launch speeds, launch angles, and wind speeds. Using interpolation, we will model the trajectories of a projectile given the above-mentioned parameters.

The equations derived using interpolation will then allow us to create probability distributions for the projectile's trajectory.

All our results will be illustrated by runs of the programs and summarized by Excel charts.

Creating a Model of Flight

To create a realistic model of flight of a projectile, we need at least the following parameters:

- Gravity
- Mass
- Cross-Section
- Shape of Surface
- Initial Velocity Vector
- Coefficient of Drag
- Rotation of Object
- Vector Field of Wind
- Rotation of Earth

In this paper we shall not consider the rotation of the Earth or of the object (the Corolis effect) as the rotation of the Earth will not be a main factor in our problems and the projectiles we consider are not rotating. The Corolis effect can be added to projectiles such as baseballs and golf balls.

The mathematical model of projectiles we are using is covered in detail in [2] and [3].

It consists of:

- Gravity only model
- The effects of drag
- Effects of wind.

The final equations are given as follows:

Where v_w is the vector of wind in the x , y, and z-directions in meters/second.

$$v_w = \begin{bmatrix} v_{wx} \\ v_{wy} \\ v_{wz} \end{bmatrix} \qquad (1)$$

Where v_x, v_y, v_z, a_x, a_y, a_z are the velocities and accelerations in the x, y, and z directions respectively, m is the mass of the object, F_D is the force of drag on the object (this can be broken into x, y, and z components for nonsymmetric objects), and g represents gravity (-9.81 m/s). The differential equations of motion in the x, y, and z directions are given by [2], [3]:

$$
\begin{aligned}
v_{ax} &= v_x + v_{wx} \\
v_{ay} &= v_y + v_{wy} \\
v_{az} &= v_z + v_{wz}
\end{aligned}
\qquad
\begin{aligned}
a_x &= \frac{-F_D v_{ax}}{m v_a} = \frac{dv_x}{dt} \\
a_y &= \frac{-F_D v_{ay}}{m v_a} = \frac{dv_y}{dt} \\
a_z &= -g - \frac{-F_D v_{az}}{m v_a} = \frac{dv_z}{dt}
\end{aligned}
\tag{2}
$$

where

$$
v_a = \sqrt{v_{xa}^2 + v_{ya}^2 + v_{za}^2} \tag{3}
$$

Using the python language, the above differential equations (1)-(3) for the trajectory (x, y, z) of the projectile are approximated using a Runge-Kutta 4 method is given below where:

p0= atmospheric density at sea level
cd=the coefficient of drag
v0=the launch velocity
theta=the launch angle
mass=mass
cs=cross sectional area (can be broken along axes to consider nonsymmetric objects)
h=step size
state= [x, y, z, vx, vy, vz] where x, y, and z are the directions in the x, y, and - directions respectively. (x is forward distance for cannon, y=height, and z=lateral distance. vx, vy, and vy are the corresponding velocities in the x, y, and z directions. All units are in metric units.

Python Program for Approximating the Trajectory of a Projectile

```
from math import exp, cos, sin, sqrt, pi

# Constants and initial conditions
p0 = 1.225
cd = 0.4
v0 = 104        # m/s
theta = 0       # degrees
thetarad = theta * pi / 180.0
```

```
vx0 = v0 * cos(thetarad)
vy0 = v0 * sin(thetarad)
vz0 = 0
mass = 227       # kg
lywind = 0
lzwind = 0

h = 0.01        # timestep
n = 5000        # maximum number of steps

# Define the derivative function for the state vector S = [x, y, z, vx, vy, vz]
def f(t, state, lxwind, lywind, lzwind):
    # Unpack state
    x, y, z, vx, vy, vz = state
    # Calculate air density based on altitude y
    p = p0 * exp(-y / 8400)
    # Update cross-sectional areas (originally based on i/100, here using t since t = i*h)
    csy = 0.4 - (t) * 0.007
    csx = 0.04 + (t) * 0.005
    csz = 0.4
    # Compute drag coefficients for each direction
    cy = 0.5 * p * csy * cd
    cx = 0.5 * p * csx * cd
    cz = 0.5 * p * csz * cd
    # Compute effective velocity relative to wind
    v_w = sqrt((vx + lxwind)**2 + (vy + lywind)**2 + (vz + lzwind)**2)

    # The system of ODEs:
    dxdt = vx
    dydt = vy
    dzdt = vz
    dvxdt = - (cx * v_w * (vx + lxwind)) / mass
    dvydt = - (cy * v_w * (vy + lywind)) / mass - 9.81
    dvzdt = - (cz * v_w * (vz + lzwind)) / mass

    return [dxdt, dydt, dzdt, dvxdt, dvydt, dvzdt]

# Loop over different x-direction wind speeds
for lxwind in range(-25, 26, 5):
    t = 0.0
    # Initial state: [x, y, z, vx, vy, vz]
    state = [0.0, 8064.0, 0.0, vx0, vy0, vz0]

    # Lists to store trajectory (if needed)
    xs = [state[0]]
```

```
    ys = [state[1]]
    zs = [state[2]]

    i = 0
    while i < n:
        # Check if the object has reached the ground (y < 0)
        if state[1] < 0:
            print("lxwind is %.2f time is %.2f x is %.2f y is %.2f and z is %.2f" %
                  (lxwind, t, state[0], state[1], state[2]))
            break

        # Compute RK4 increments
        k1 = f(t, state, lxwind, lywind, lzwind)
        state_k2 = [state[j] + (h/2)*k1[j] for j in range(6)]
        k2 = f(t + h/2, state_k2, lxwind, lywind, lzwind)
        state_k3 = [state[j] + (h/2)*k2[j] for j in range(6)]
        k3 = f(t + h/2, state_k3, lxwind, lywind, lzwind)
        state_k4 = [state[j] + h*k3[j] for j in range(6)]
        k4 = f(t + h, state_k4, lxwind, lywind, lzwind)

        # Update state vector using RK4 formula
        state = [state[j] + (h/6) * (k1[j] + 2*k2[j] + 2*k3[j] + k4[j]) for j in range(6)]

        t += h  # update time
        xs.append(state[0])
        ys.append(state[1])
        zs.append(state[2])
        i += 1
```

Distance Probabilities with respect to Initial Angle and Initial Launch Velocity

Consider the following parameters for our model:

```
p=1.225
cs=.0324294 #units in square meters
cd=.3
c=.5*p*cs*cd
v0=300 #in meters per second
m=4 #in kg
```

Note: These parameters are from a contest called “Pumpkin Chunkin”, hence the mass of 4 kg and the cross-sectional area. The objective was to see how far a given machine could launch a pumpkin (distances of almost 5000 feet were attained before the contest was cancelled, people just became too frighted by 4 kg pumpkins flying almost a mile).

In Figures 1 and 2 below, our program was run for various launch angles and launch speeds with the given parameters. The results were graphed using Excel.

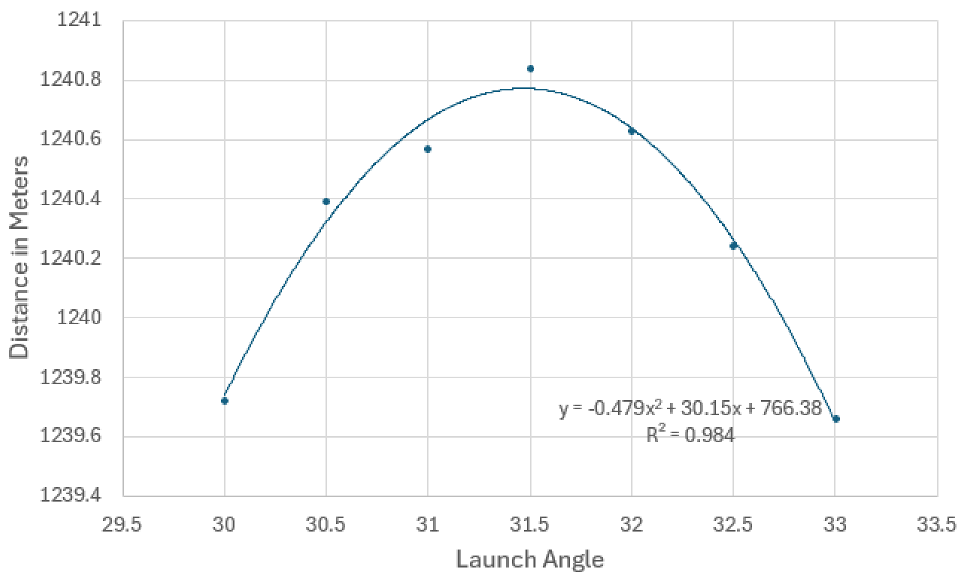

Figure 1

From Figure 1, we can see that the maximum distance occurs for a launch angle of about 31°.

Note: In typical calculus textbooks, it is stated that to obtain a maximum distance for a launched projectile, the launch angle should be 45°, however this does not consider an atmosphere.

Now, the objective may be to launch at exactly 31°, but the actual launch angle is a probability distribution. Say a uniform or normal distribution with a mean of 31°. If there is a 90% chance the launch angle is between 30° to 32° we can see from Figure 1 there is a 90% chance the distance of the projectile will be between 1239.7 and 1240.8 meters. Thus, if we have a probability distribution of the launch angle, we can determine the corresponding probability distribution the distance the projectile travels.

Figure 2 considers the distance the projectile with fly for various launch velocities.

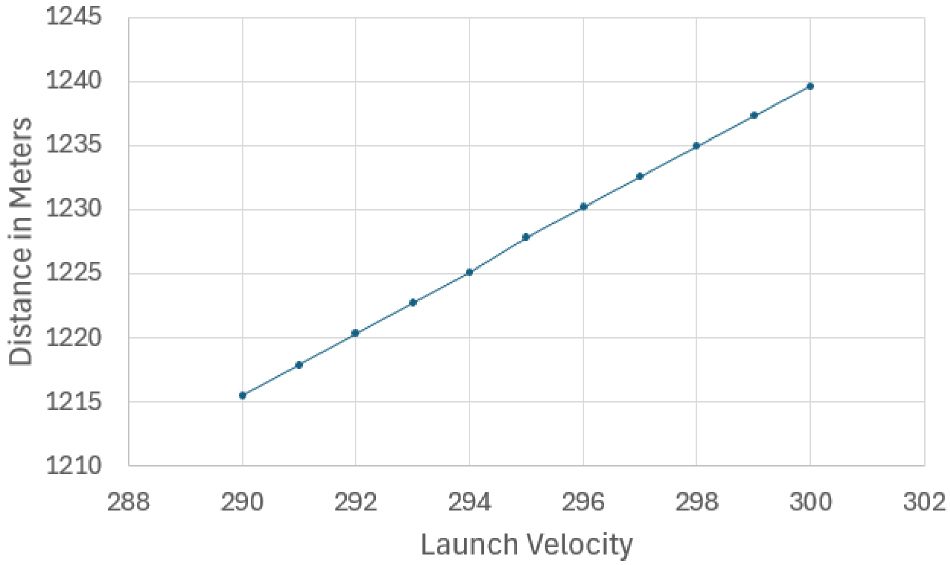

Figure 2

As above, the objective may be to launch the projectile at a certain velocity, however, in reality, what we have is a probability distribution for the launch velocity. Given this, we can find the probability the object flies a certain distance. For example: If there is a 95% chance the launch velocity is between 290 and 300 m/s, the is a 95% chance the distance traveled will be between 1215 and 1240 meters.

Note: We can form joint tables for both launch angle and velocity to form joint probability distributions for the distance the projectile flies based on these parameters.

We shall study this in our next problem.

Why 1000 Bomber Raids in WWII

Reason: A typical bomb dropped from 25,000 feet and had almost no chance of hitting a given target on the ground.

So: Send 500 or 1000 planes and drop 5,000 to 10,000 bombs and hope you hit your target.

Models for WWII Bombers and Bombs [4],[5], [6]

Approximate Speed: 200-240 mph	Variables:

500/1000 pound bombs each: 1 meter long 16’’-18’’ wide Effective Damage Radius: 25 meters 8-10 Bombs per Plane	Height of Plane Speed of Plane When Dropped Wind in the x, y, z Direction Fusing Coriolis Effect (small relative to other variations) Mangus Effect (None)

Table 3

Note: WWII bombers used both airspeed and ground speed in their raids. Airspeed is the speed of the planes relative to the surrounding air and groundspeed is the speed relative to the ground. Here we shall consider speed to the speed at which the bomber was approaching the target [7].

In our model, we shall consider dropping bombs from 8064 meters (about 26000 feet). Hence, we need:

1) an atmospheric model given by:

p=1.225*exp(-y[i]/8400) (4)

where p is the atmospheric density in kg/m^3 and y is the number of meters above sea level, and

2) cross sections along the x- y- and z-axes for our bombs (in meters)

csy=.4-(i/100)*.007
csx=.04+(i/100)*.005 (5)
csz=.4

Note: As the bomb rotates from a horizontal to a vertical position as they fall, we need to adjust the cross-sectional area as the bombs fall.

Default Parameter Values:

v0=104 #in meters per second
theta=0 #angle in degrees
mass=227 #in kg
lxwind=0 #in meters per second
lywind=0
lzwind=0
xn=0
	yn=8064
	zn=0
	p=1.225*exp(-y[i]/8400)
	csy=.4-(i/100)*.007
	csx=.04+(i/100)*.005
	csz=.4

Table 4

Here the initial height is 8064 meters, and the launch angle is 0° as the bombs are dropped out of the planes horizontally.

With these initial values, we get a baseline result x of 3832.90 y is 0 and z is 0 (meters). This says the bomb will travel 3832.90 meters horizontally before it hits the ground.

Wind can Blow Bombs Hundreds of Meters Off Target

Figure 5 below shows what happens when there is wind blowing along the main axis of the plane (and hence the bombs). We see that even small amounts of wind along the x-axis can throw the bombs hundreds of meters off target.

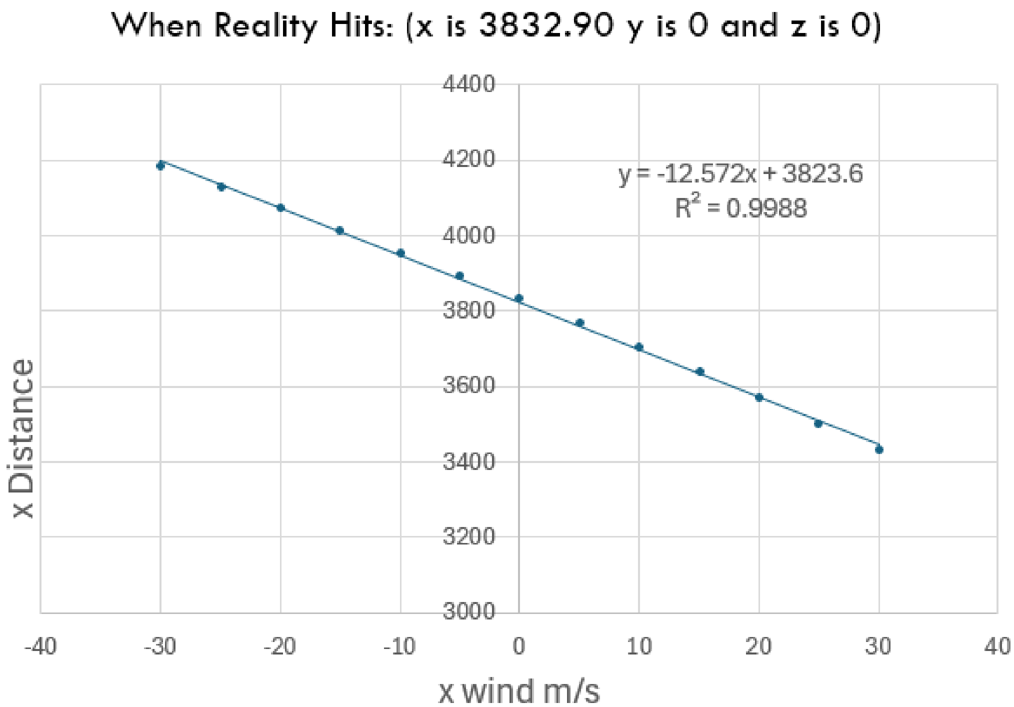

Figure 5- wind vs how far the bomb falls before it hit the ground

Figure 6 below shows what happens when there is wind blowing across the main axis of the plane (and hence the bombs), the z-axis. We see that even small amounts of wind along the z-axis can throw the bombs hundreds of meters left or right off target.

As wind models at the time of WWII were only so accurate, even small amounts of wind and wind gusts could throw the bombs hundreds of meters off target.

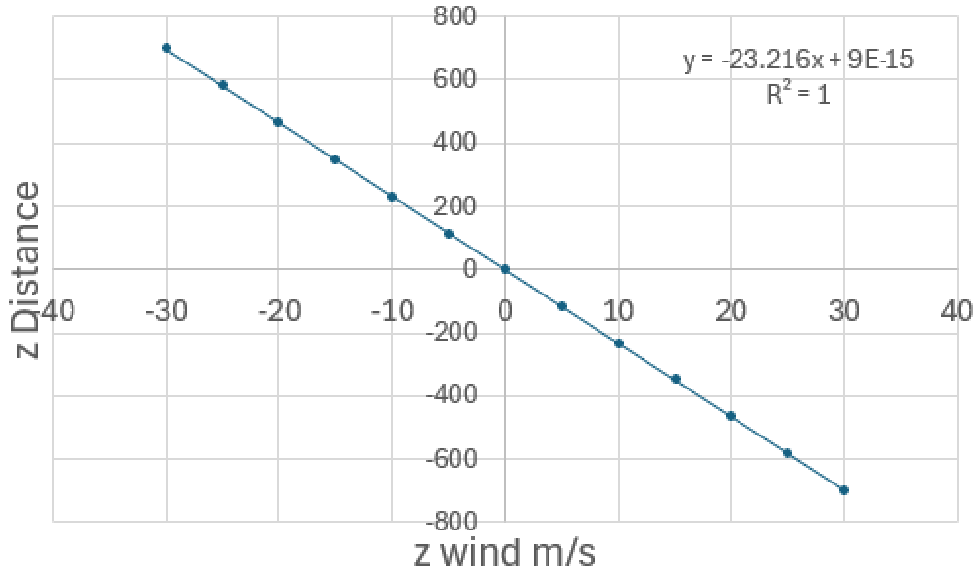

Figure 6

Combining Wind in the x and z Directions
(Benchmark Values): x is 3832.90 y is 0 and z is 0 v0=104 m/s
lxwind=lzwind=0

In table 7 lxwind>0 is a headwind and lxwind<0 is a tailwind. lzwind is the wind on the sides of the bombs.

lxwind	lzwind	x	z
-20.00	-20.00	4072.72	441.80
-20.00	-10.00	4073.09	220.14
-20.00	0.00	4073.51	0
-20.00	10.00	4073.09	-220.14
-20.00	20.00	4072.72	-441.80
-10.00	-20.00	3954.51	452.36
-10.00	-10.00	3955.43	225.57
-10.00	0.00	3955.86	0
-10.00	10.00	3955.43	-225.57
-10.00	20.00	3954.51	-452.36
0.00	-20.00	3831.75	463.46
0.00	-10.00	3832.76	231.17
0.00	0.00	3832.76	0
0.00	10.00	3832.76	-231.17
0.00	20.00	3831.75	-463.46
10.00	-20.00	3703.40	474.67
10.00	-10.00	3704.51	236.82
10.00	0.00	3704.95	0
10.00	10.00	3704.51	-236.83
10.00	20.00	3703.40	-474.67
20.00	-20.00	3569.51	486.13
20.00	-10.00	3570.69	242.59
20.00	0.00	3571.14	0
20.00	10.00	3570.69	-242.59
20.00	20	3569.51	-486.13

Table 7

Small Deviations in the Speed of the Plane and Height can Cause Bombs to Fall Hundreds of Meters Off Target

Figure 8 below shows that a difference of 1 m/s in the plane's speed will cause a 35 meter difference in how far the bomb will fall horizontally. In Figure 9 we see shows that for each 100 meters difference in height the bomb when dropped will cause a 16-meter difference in how far the bomb falls horizontally.

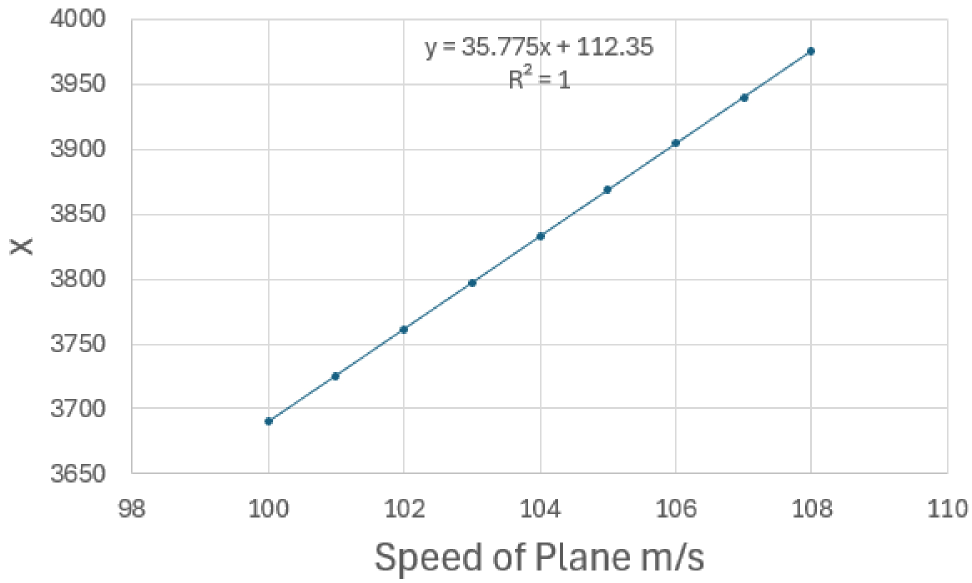

Figure 8

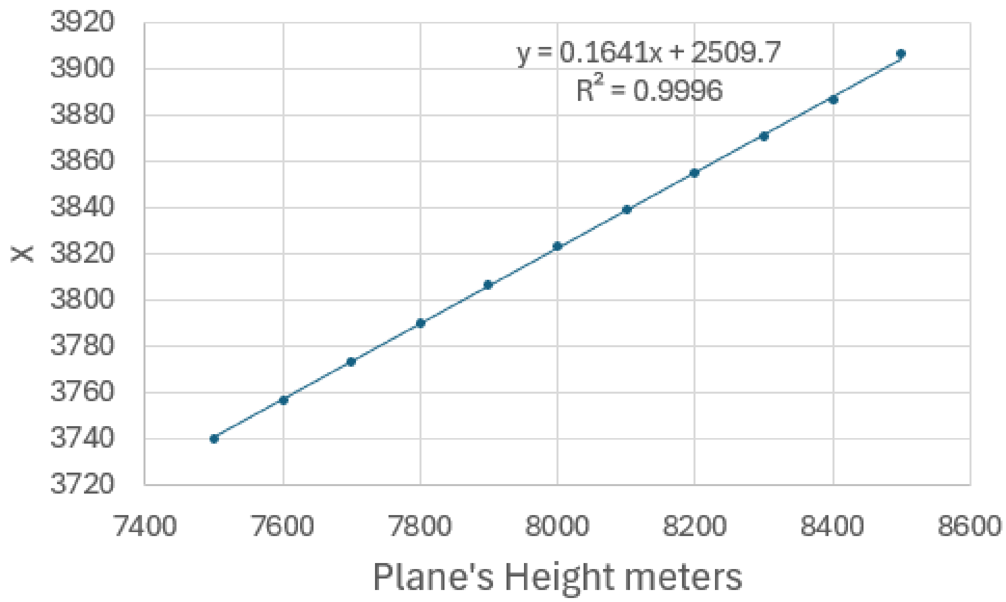

Figure 9

A topic for a future paper is that of studying various bombing formations and the effects when dropping 5000-10000 bombs over a target-and how winds and other errors caused massive distribution in the paths of the falling bombs.

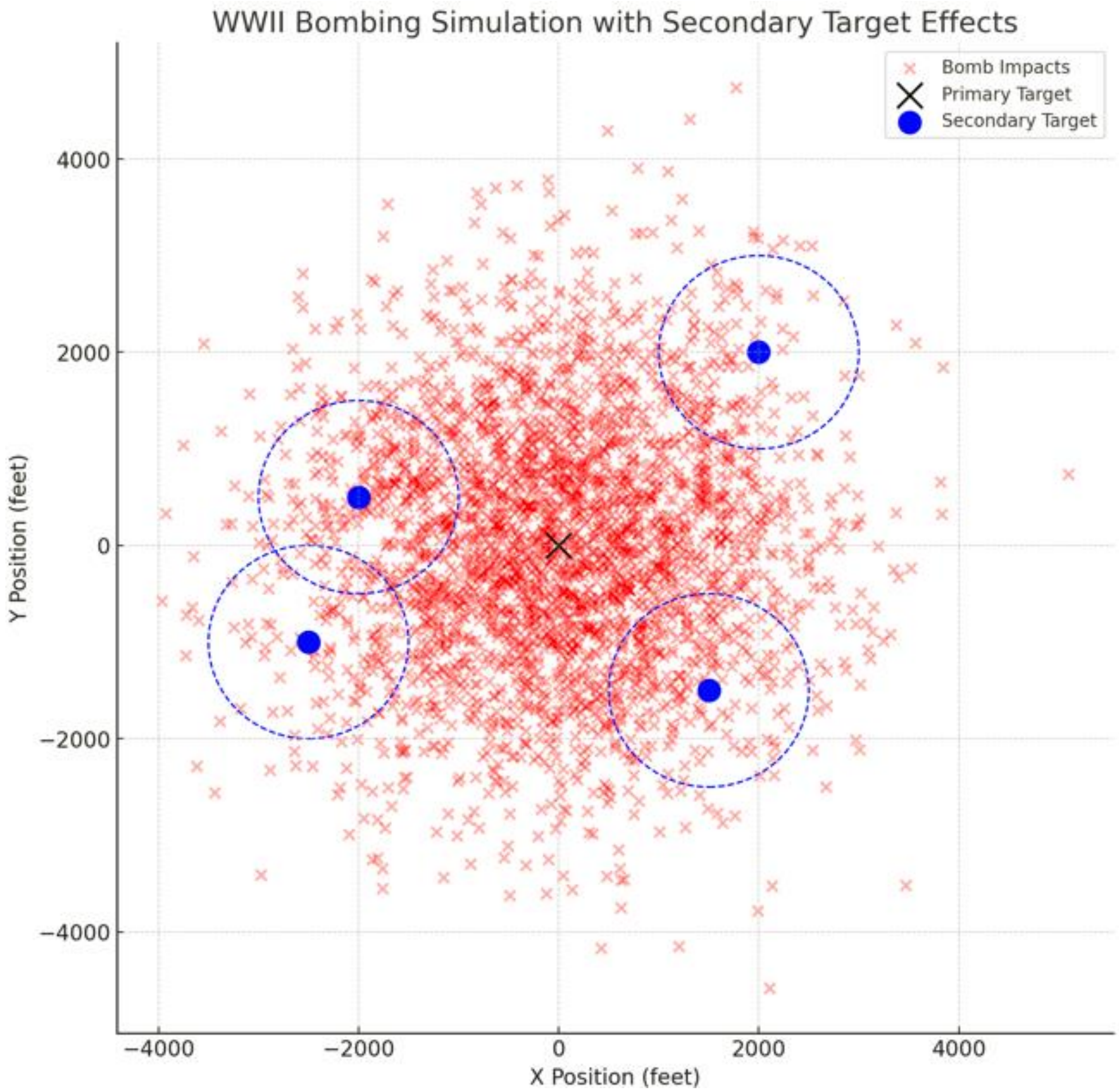

Figure 10

Summary

When creating mathematical models, one needs to consider how changes in the values of parameters can change the outcome. Sometimes, even small changes in inputs can cause large changes in output. If we can find probability distributions for the values of parameters, we can often create corresponding probability distributions for the outcomes.

References:

[1] James Stewart, Calculus 7th Edition, Brooks-Cole, Belmont, CA, 2012.
[2] “The Effect of Wind on the Trajectory of Golf Balls” Proceedings of the 29th ICTCM https://www.pearson.com/content/dam/one-dot-com/one-dot-com/us/en/files/Paul-Bouthellier-ICTCM29.pdf
[3] Grant Palmer, Physics for Game Programmers, Apress, New York, NY 2005
[4] “General-purpose bombs”, e.wikipedia.org/wiki/General-purpose_bomb.htm
[5] “The Development and Trials of a Modified 500 lb Bomb”, apps.dtic.mil/sti/tr/pdf/AD0071692.pdf
[6] “B17| Crew, Range, and Bomb Load”, https://www.britannica.com/technology/B-17.htm.
[7] What's the Difference Between Airspeed and Ground Speed? https://science.howstuffworks.com/transport/flight/modern/airspeed-vs-groundspeed.htm