Finding the Optimal Angle for a Projectile by Calculating the Minimum Value of a Single-Variable Function using MATLAB

Yong S. Suh
California State University, Sacramento

If an object is shot with an initial velocity at an angle, then the projectile of the object can be calculated. If the initial velocity is fast enough, the object is to find the initial velocity and angle of the object so that it hits the target on the wall that is standing at a certain distance from the launch. If there is no air resistance, this problem can be easily solved with a few simple mathematical equations in a closed analytic form. The dragging force by the air, however, it is not trivial to solve the problem by hand.

By keeping the same initial velocity, the trajectory of the object can be changed by adjusting the initial inclination angle of the projectile, hence hitting the target. The goal of this activity is to find the best angle of the initial inclination for the object to hit a specific target on the wall given the initial velocity. The graph below illustrates an example of the trajectory of an object launched at the origin (0,0) and hitting the target at (1000,100) on the wall that is 1000 units away from the launch.

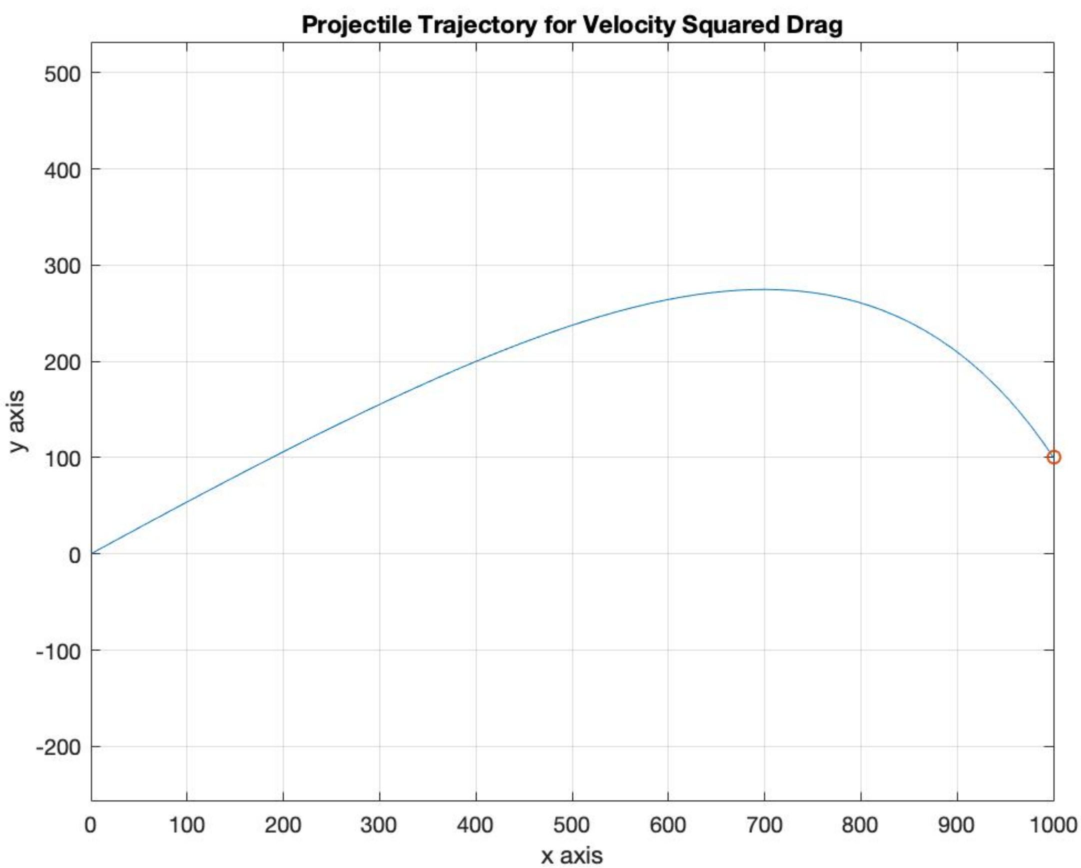

Figure 1. Projectile trajectory

Equations of motion of a projectile

A reasonable model for the projectile motion is to assume that the atmospheric drag is proportional to the square of the velocity. Hence,

$$\dot{v}_x = -cvv_x \qquad (1)$$
$$\dot{v}_y = -g - cvv_y \qquad (2)$$
$$\dot{x} = v_x\,,\ \dot{y} = v_y \qquad (3)$$
$$v = \sqrt{v_x^2 + v_y^2} \qquad (4)$$

where g is the gravity constant and c is a ballistic coefficient depending on such physical properties as the projectile shape and air density. The dot indicates the time derivative of the variable.

We want to know the changes of velocities (v_x, v_y), vertical location (y), and time (t) with respect to the horizontal distance (x).

For $\frac{dt}{dx}$:

$$dx = v_x dt,\ \ \frac{dt}{dx} = \frac{1}{v_x} \qquad (5)$$

Applying the chain rule, the derivatives of v_x, v_y and y with respect to x can be easily found.

For $\frac{dv_x}{dx}$

$$\frac{dv_x}{dx} = \frac{dv_x}{dt}\frac{dt}{dx} = (-cvv_x)\frac{1}{v_x} = -cv \qquad (6)$$

For $\frac{dv_y}{dx}$, applying the chain rule:

$$\frac{dv_y}{dx} = \frac{dv_y}{dt}\frac{dt}{dx} = (-g - cvv_y)\frac{1}{v_x} \qquad (7)$$

For $\frac{dy}{dx}$:

$$\frac{dy}{dx} = \frac{dy}{dt}\frac{dt}{dx} = v_y\frac{1}{v_x} = \frac{v_y}{v_x} \qquad (8)$$

For a given distance to the wall $L = x_w$, we want to find the height (y) of the projectile at $x = x_w$ by integrating the equation (8).

MATLAB Implementation

Set the gravitational constant, coefficient of drag as global variables so that they can be shared by all functions.

Implement a function, trajectory(), that takes an initial angle, initial velocity, drag coefficient, gravitation constant, and the distance to the wall as its input, and output the hit position on the wall, and the time it reaches the wall.

```
[y,x,t] = trajectory( angle, vinit, gravity, cdrag, xfinal)
```

Construct an array, $z = [v_x;\ v_y;\ y;\ t]$. Then the first order differential equation of z is:

$$\frac{dz}{dx} = \frac{[-cvv_x; -(g+cvv_y);\ v_y;\ 1]}{v_x} \qquad (9)$$

Implement a function, `project_equation(x,z)` and output [x,z].

Use the MATLAB ode45() function to solve for z.

```
   [x, z] = ode45(@project_equation, [0, xfinal], z0, opt) (10)
```

where the initial value z0 is

```
     z0 = [vinit * cos(angle); vinit * sin(angle); 0; 0] (11)
```

In function project_equation(), it should implement Equation (9) and returns z. For a scalar x and a vector z = [$v_x(x)$; $v_y(x)$; y; t], the function returns a column vector for x and column vectors of z of which each row corresponds to the row of x vector.
Once y vectors are obtained, plot the (x,y) values on a graph as shown in Figure 1.

Finding the Best Angle

Given an inclination angle of the projectile, we can measure the error between the target and the computed y. We need to find the best angle that minimizes the error. The error is defined by the square of the difference:

$$Error = (y(x_w) - y_{target})^2 \qquad (12)$$

MATLAB Implementation

Create a function, `runtraj(vinit)`, in which the MATLAB function fminbnd() is called to minimize the error. The input to the function is the initial velocity, v(0). The bound of the single variable is [(atan(yfinal/xfinal) * 180/pi, 75]. The input function to fminbnd() is Equation (12).

Reference

Howard B. Wilson, Louis H. Turcotte , David Halpern, “Advanced Mathematics and Mechanics Applications Using MATLAB”, 3rd Edition, Chapman & Hall/CRC , 2003