Documentation and Validation of EveryCalc's Trajectory Tool

Thaddeus Hughes
hughes.thad@gmail.com
thaddeus-maximus.github.io

May 27, 2020

Abstract

Hurling projectiles is something we humans really like doing. We've become exceedingly efficient at it. We make sport of it. We can guess at a trajectory pretty easily- but nailing it down, and tweaking it with an engineering mindset is harder. Lots of tools already exist to do this but I want to roll my own so I can add all the physics I want, along with the swept area of a projectile, and stacking the tolerance.

1 Basic Projectile Motion

Consider a ball of mass m, in a vacuum. It has only the force of gravity acting on it. That is to say,

$$\Sigma F_x = m\frac{dv_x}{dt} = 0 \tag{1}$$

$$\Sigma F_y = m\frac{dv_y}{dt} = -mg \tag{2}$$

To figure out the path, we'd also need to know the initial conidtions. Let's also say that the ball was launched at an angle θ from the horizontal at an initial velocity of $\bar{v}_0$, so that the x- and y- velocities would be

$$v_x(0) = \bar{v}_0\ cos(\theta) \tag{3}$$

$$v_y(0) = \bar{v}_0\ sin(\theta). \tag{4}$$

We'll start from a height of y_0 and at $x = -x_0$ from our target.

$$x(0) = -x_0 \tag{5}$$

$$y(0) = -y_0 \tag{6}$$

This is enough to get us a very simple simulation for projectile motion:

$$\frac{dv_x}{dt} = 0 \tag{7}$$
$$\frac{dv_y}{dt} = -g \tag{8}$$
$$\frac{dx}{dt} = v_x \tag{9}$$
$$\frac{dy}{dt} = v_y \tag{10}$$
$$v_x(0) = \bar{v}_0\ cos(\theta) \tag{11}$$
$$v_y(0) = \bar{v}_0\ sin(\theta) \tag{12}$$
$$x(0) = -x_0 \tag{13}$$
$$y(0) = y_0 \tag{14}$$
$$\text{terminate when } x \geq 0 \tag{15}$$

2 Swept Path with Parallel Curves

It's also worth knowing the swept zone that the target travels, since the object of firing a projectile may not be to hit a target per se, but to make it *through* a target. This may seem like a trivial task at first blush; just add on the radius of the ball to the y-direction, but then one realizes the projetile may not be striking the target dead-on. Creating an offset path, or parallel curve is necessary. The upper swept path $(x_u,\, y_u)$ of a ball of radius r can be determined as

$$x_u = x + r(\hat{x} \cdot \hat{N}) \tag{16}$$
$$y_u = y + r(\hat{y} \cdot \hat{N}) \tag{17}$$

Where $\hat{x}$, $\hat{y}$, $\hat{N}$, $\hat{T}$ are unit vectors in the x-, y-, normal, and tangent directions.

$$\text{let } \bar{v} = \sqrt{v_x^2 + v_y^2} \tag{18}$$
$$\hat{x} \cdot \hat{N} = -v_x/\bar{v} \tag{19}$$
$$\hat{x} \cdot \hat{T} = +v_y/\bar{v} \tag{20}$$
$$\hat{y} \cdot \hat{N} = +v_x/\bar{v} \tag{21}$$
$$\hat{y} \cdot \hat{T} = +v_y/\bar{v} \tag{22}$$

The lower path would be found by reversing the direction of r, yielding

$$x_l = x - r(\hat{x} \cdot \hat{N}) \tag{23}$$
$$y_l = y - r(\hat{y} \cdot \hat{N}). \tag{24}$$

3 Aerodynamic Effects

There are multiple aerodynamic forces that can act on a ball.

$$F_{drag} = \frac{1}{2}\ C_{drag}\ \rho\ A\ \bar{v}^2 \tag{25}$$
$$F_{lift} = \frac{1}{2}\ C_{lift}\ \rho\ A\ \bar{v}^2 \tag{26}$$
$$F_{magnus} = \frac{\pi}{3}\ C_{magnus}\ \rho\ A\ \bar{v}\ \omega_{+\mathrm{ccw}} \tag{27}$$

The lift and drag forces are standard equations, but the magnus force equation is derived from this NASA page. There are undoubtedly better models out there, but this is what I have currently.

In vector form the conservation of momentum could be written as

$$m\frac{d\vec{v}}{dt} = -F_{drag}\hat{T} + F_{lift}\hat{N} + F_{magnus}\hat{N} - mg\hat{y} \tag{28}$$

This changes the model, when projected out into the x- and y- components to

$$\frac{dv_x}{dt} = \frac{-F_{drag}(\hat{x}\cdot\hat{T}) + F_{lift}(\hat{x}\cdot\hat{N}) + F_{magnus}(\hat{x}\cdot\hat{N})}{m} \tag{29}$$

$$\frac{dv_y}{dt} = \frac{-F_{drag}(\hat{y}\cdot\hat{T}) + F_{lift}(\hat{y}\cdot\hat{N}) + F_{magnus}(\hat{y}\cdot\hat{N})}{m} - g \tag{30}$$

$$\frac{dx}{dt} = v_x \tag{31}$$

$$\frac{dy}{dt} = v_y \tag{32}$$

$$v_x(0) = \bar{v}_0\ cos(\theta) \tag{33}$$

$$v_y(0) = \bar{v}_0\ sin(\theta) \tag{34}$$

$$x(0) = -x_0 \tag{35}$$

$$y(0) = y_0 \tag{36}$$

$$\text{terminate when } x \geq 0. \tag{37}$$

4 Tolerance Stacking

To determine accuracy, multiple iterations of the simulation can be ran with different permutations of input variables.

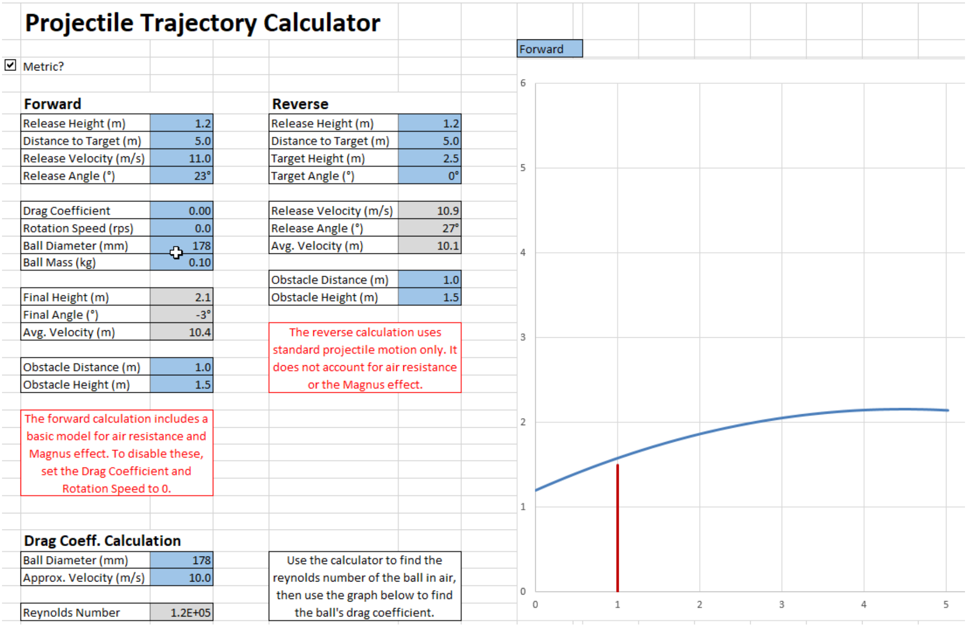
5 Reverse Computation

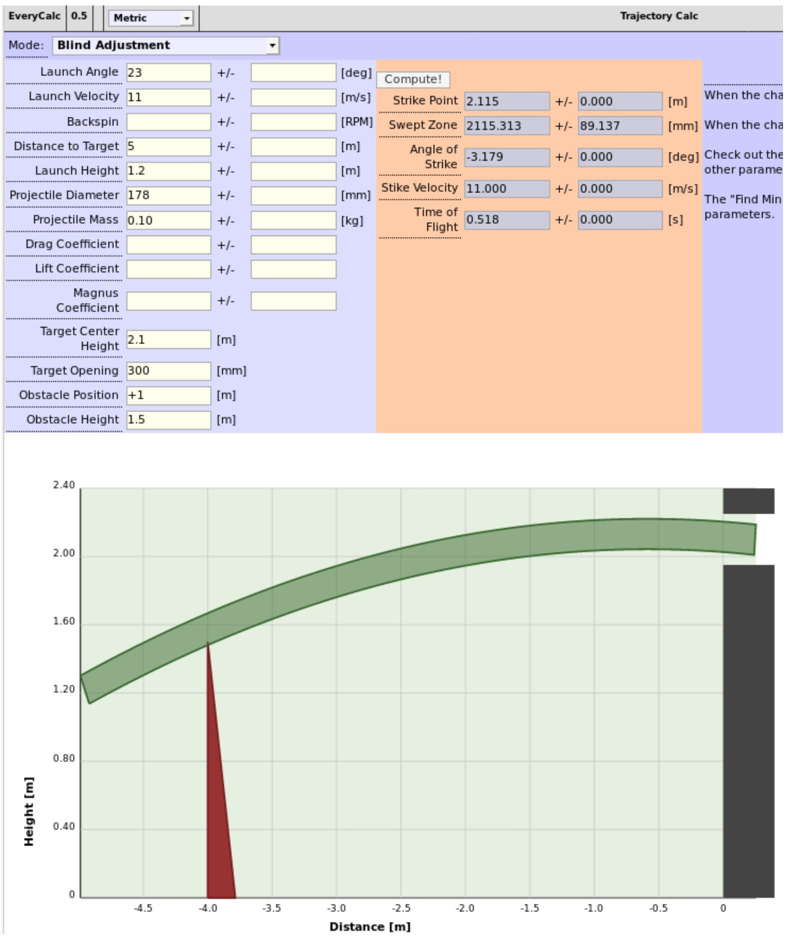
A bisection algorithm is used to solve for the appropriate distance/angle/velocity required to propel the projectile into the target. This has benefits over analytical solutions in that it can be used in conjunction with aerodynamic effects.

6 Validation Against Other Tools

I'll compare results to AMB's Design Spreadsheet.

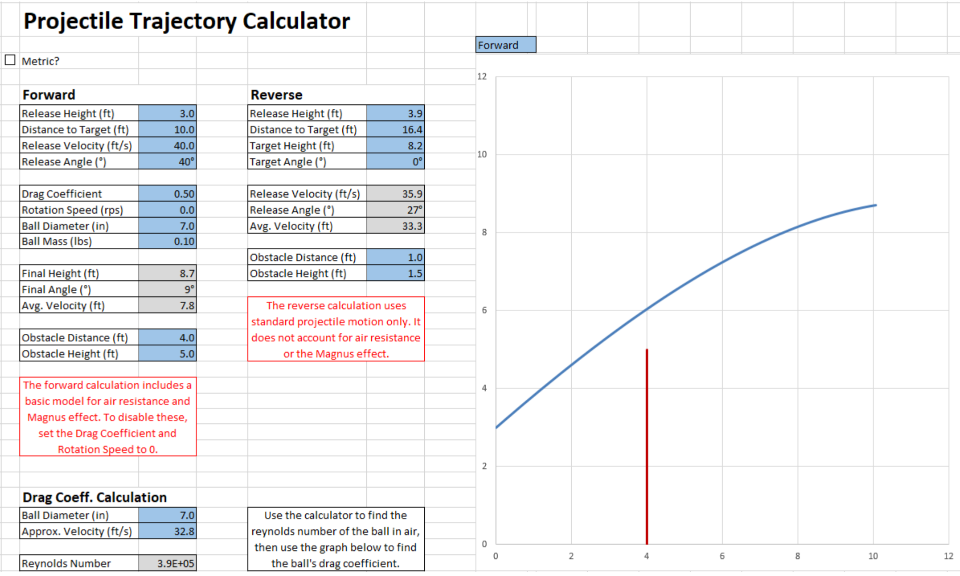
Case A: Metric units, no aero

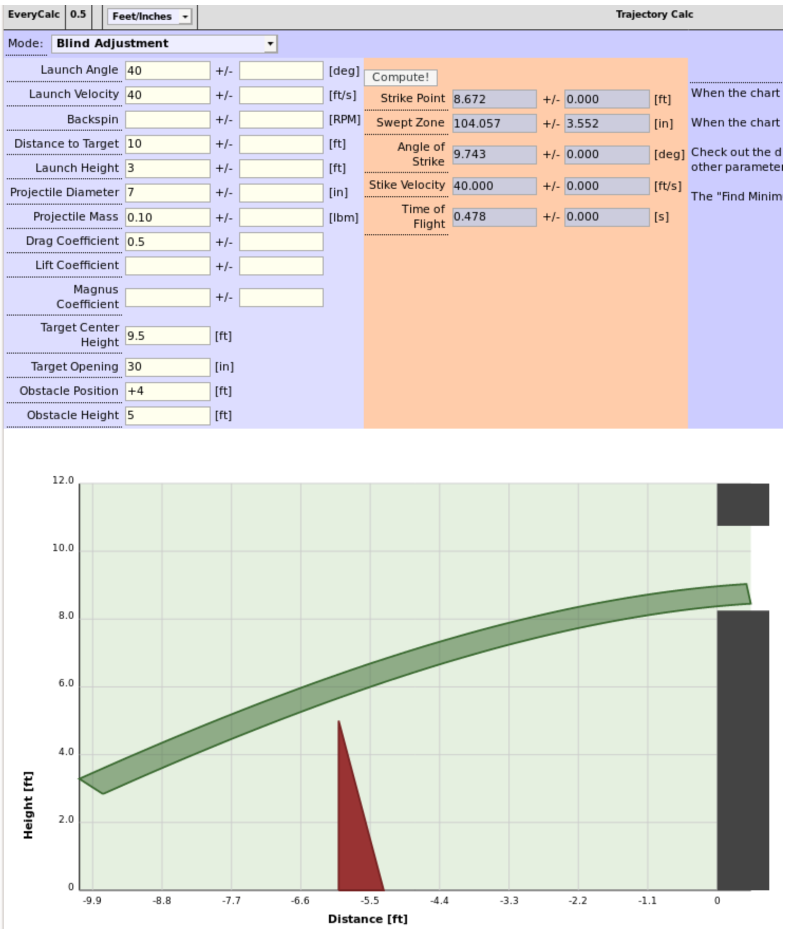




Look only at the "Forward" portions of AMB's sheet. Effectively the same result.

Case B: English units, drag included





Look only at the "Forward" portions of AMB's sheet. Effectively the same result.