Rahul Jayaraman

Dr. Eric R. Nelson

ATCS: Numerical Methods

30 October 2014

Projectile Motion: Can We Drag Ourselves Out Of Erroneous Models?

**Introduction**

Numerical methods can prove very useful to simulate many physical phenomena that may not have a closed-form solution, such as projectile motion and the N-body problem. In this paper, we specifically focus on the applications of numerical methods to analyze projectile motion with drag forces in both the x- and y-dimensions.

We first implemented an algorithm for Euler’s method, an SN-order method to solve first-order differential equations. Our implementation involves the use of a method that uses a while loop and takes in four parameters: a starting x-value, an ending x-value, a starting y-value, and an array of coefficients for the differential equation, which can take various forms, including trigonometric, polynomial, and exponential.

Through the use of Euler’s method applied to an arbitrary acceleration function (that may or may not involve appropriate drag forces) given a starting velocity, we identified a velocity function. At the time that the projectile hit the ground (from a position function), we obtained a velocity and compared that to the starting velocity, as both should be the same. We also used similar logic for the y-velocity, albeit with reversed signs; we *made an assumption* that the cannon was launched directly from ground level.

Of course, our analysis of the forces must have some constants previously defined. Our projectile in this project was a 5.5 kg cannonball, with a radius of 10.5 cm (diameter 21 cm) and drag coefficient of 0.45; the cannonball was fired at a 45o angle from the horizontal. Our step size for the Euler’s method function was 0.1, and the value of *g* (the gravitational field strength on Earth) that we used in our force analysis was 9.81 (m/s2).

**We define divergence from our model as differing more than 5%. For instance, if the value we obtain from our adiabatic analysis is not within 5% of the value we obtain from an analysis without drag forces, then we will call that a “divergent value.”**

**Baseline: No Air Resistance**

In this model, the projectile faces no air resistance; the only force that it is subject to is gravity. There *is* a closed-form solution for this, and we will use this to verify our numerical analysis results. We expect that the velocity once it hits the ground should be the same as the start velocity (variable *v*). The x-velocity, without drag (or any forces), is obviously the same. However, when we apply Euler’s method with the appropriate initial conditions, we see that as the starting velocity increases, the accuracy of our approximation decreases. By the time we hit |*v*| = 100 m/s, we see that our approximation is 1% off from the actual value predicted by the model. ***This, however, is not a result of drag forces at play. It is merely a result of our simulation and the lack of machine precision when doing a multitude of “piggy-backing” calculations, i.e. those that rely on previous approximations***. This does not diverge.

**Constant Air Density**

This model assumes a constant air density, or ρ. Air has a density of 1.225 kg/m3 AMSL and at 15o C, according to Wikipedia. Plugging this back

**Adiabatic Air Density**

**Isothermal Air Density**