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 with drag 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; this equation 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 obtained a set of velocity data. At the time that the projectile hit the ground (obtained from position data and back-calculated), we compared the magnitude of the impact y-velocity to the starting velocity's magnitude. Comparing these values to a non-drag, closed-form model forms the basis of this paper.

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 45^{o} 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/s^2). In our methods, we used two separate while loops – as the acceleration during upward flight is different from that of downward flight.

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

Assumption: In this model, the projectile faces no air resistance; the only force that it is subject to is gravity.

Results: 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*). 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 predicted one. ***This, however, is not a result of drag forces. It is merely a result of our simulation and the lack of machine precision when doing a multitude of successive calculations, i.e. those reliant on previous values***.

Analysis and Divergence: This model, of course, cannot diverge – it is our baseline!

Constant Air Density

Assumptions: This model assumes a constant air density, or ρ, of 1.225 kg/m^3 AMSL at 15^{o} C (Wikipedia).

Results and Analysis: Plugging this value into our algorithm, we observe a significant departure from the predicted velocity. The model starts to diverge when the muzzle velocity magnitude is 22 m/s, or about 49.2 miles per hour (Fig 1) – about the speed of a car on an expressway. More specifically, the impact y-velocity without drag is 15.54 m/s, while the impact y-velocity with constant drag is 14.66 m/s – a deviation of 5.66%, which is above our limit of 5% – indicating the start of divergence from our model.

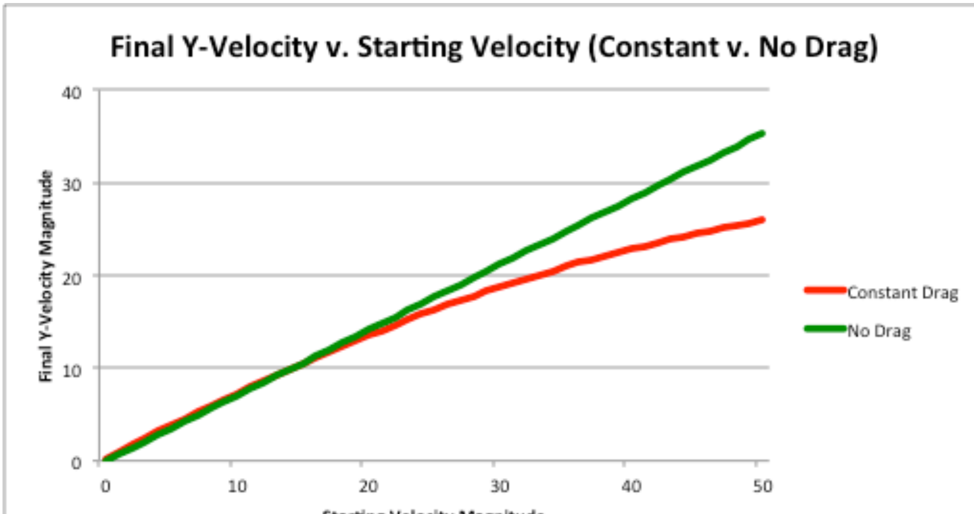

Fig 1

This is as expected; we know that the model for drag is a power-law dependent on velocity. As the velocity increases by a factor of 2, for instance, the drag force increases by a factor of 4 ($F_D \propto v^2$), thus slowing the object down. As a direct result, we observe an increased acceleration upon impact as starting velocity increases.

<u>Adiabatic Air Density</u>

Assumptions: For this model, we assume that the temperature was 15^o C, or 288 K, at all altitudes attained by the cannonball. We also assume an initial air density ρ of 1.225 kg/m^3 AMSL as well as an adiabatic density model for the atmosphere ($\rho = \rho_o(1 - \frac{ry}{T_o})^{2.5}, r = 6.5 * 10^{-3}$), in which there is negligible heat transfer with air movement.

Results and Analysis: We can see that there is not a significant departure from the constant drag model for the altitude that we normally fire cannons at; indeed, at a magnitude of 50 m/s (112 mph), the adiabatic and constant-drag models only differ by .006% -- well within our divergence limit. In fact, using an adiabatic model results in a slightly higher impact velocity due to the slightly lessened drag at "higher" altitudes, such as the peak of the cannonball's flight. As we increase our altitude *significantly* (by thousands of meters), density becomes negligible, and we observe results that are closer to the no-drag model. Shifting our focus back to the ground, we can see in Fig 2 that the adiabatic model starts to diverge from the no-drag model at a starting velocity magnitude of 22 m/s.

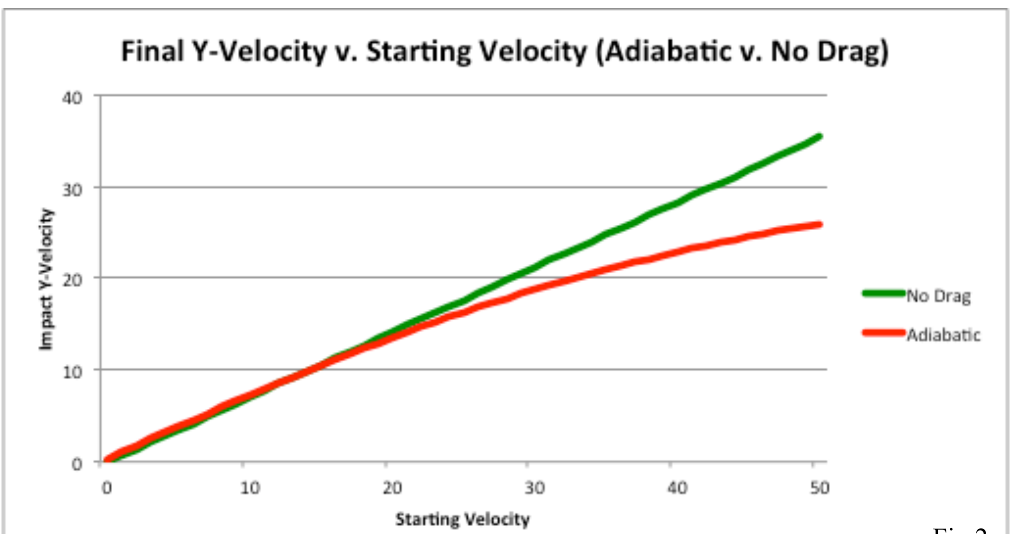

Fig 2

We also observe the obvious lack of divergence between the adiabatic and constant-drag models in Fig 3 (Note: We use a 3-D graph here so that we can see both the models; using a 2D graph, one model covers the other):

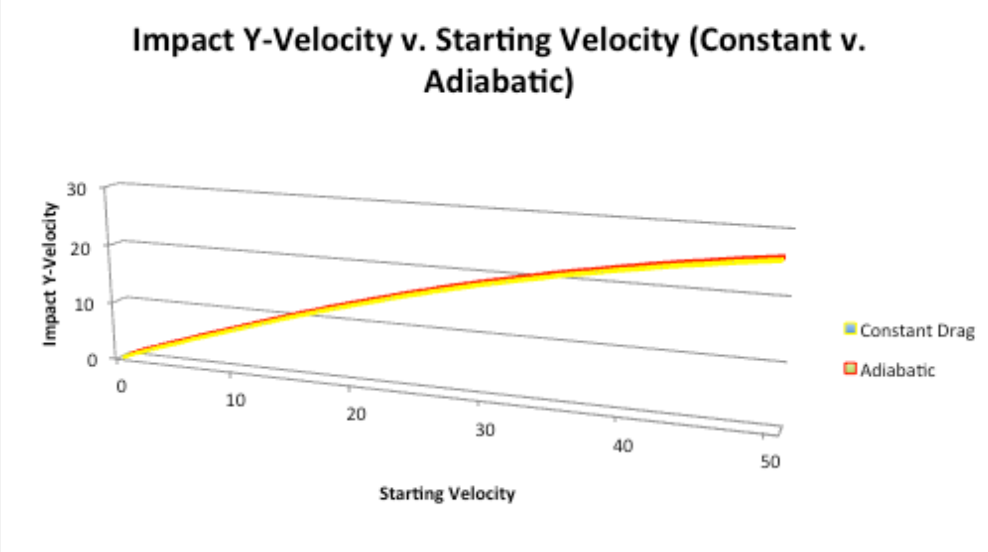

Fig 3

Isothermal Air Density

Assumptions: In an isothermal density model ($\rho = \rho_o e^{-\frac{y}{y_o}}$), we assume that all heights achieved by the cannonball are at the same temperature (288 K). We also assume an initial air density ρ of 1.225 kg/m^3 at MSL. In our code, for the value of y_o, we used the previous height and calculated the density at the new height relative to the old height.

Results and Analysis: First, we compared the isothermal model to the no-drag model. We observed that the two models started to diverge when the starting velocity magnitude was 29 m/s (65 mph, approximately the speed of a car on a freeway); this divergence is clearly visible in Fig 4. This is probably a result of the changing nature of y/y_o, as the projectile gains a greater height (and that Δy increases) with each increase in velocity.

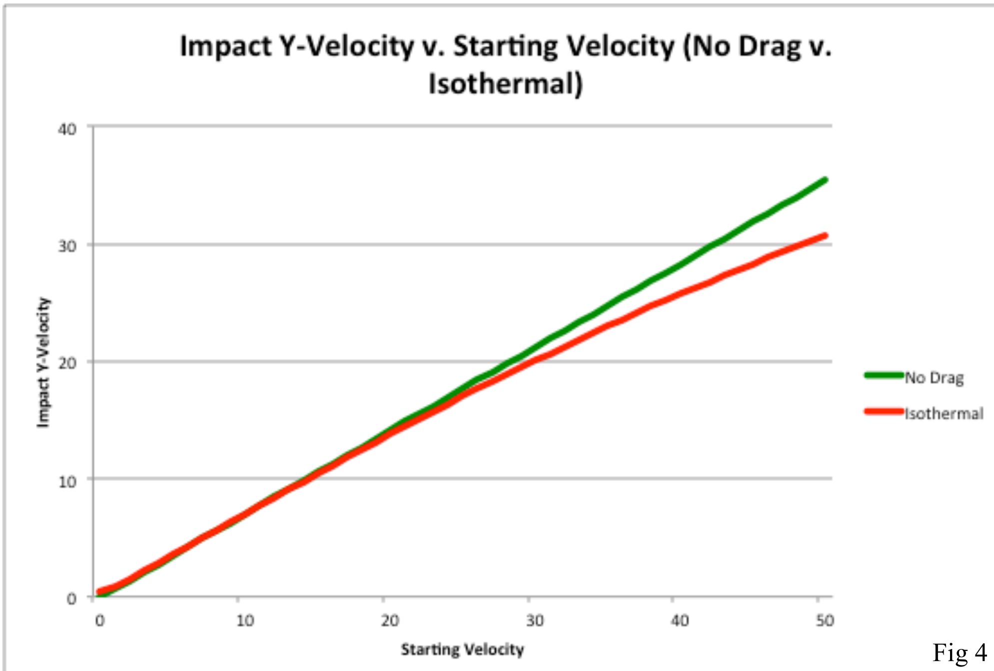

Fig 4

We also compare the isothermal model to the adiabatic model (Note: We do not compare the isothermal model to the constant-drag model, as the constant-drag model and the adiabatic one are extremely similar for the heights being considered in the problem). The divergence occurs at a starting velocity magnitude of 27 m/s (60.4 mph) and can be directly attributed to differences between the models in calculating density (Fig 5).

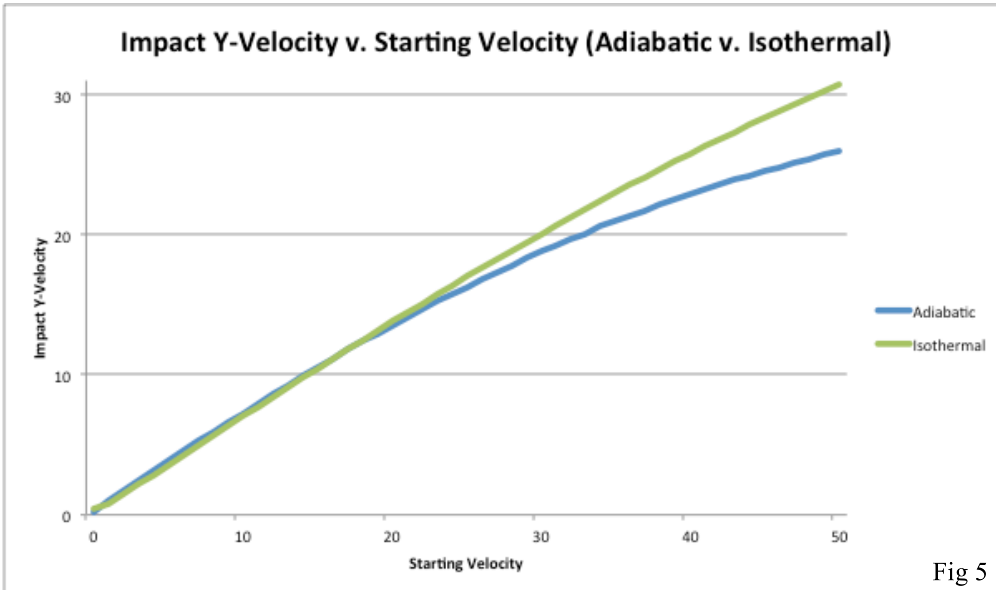

Fig 5

Comparison: We take into account changes in temperature in the adiabatic model, but cannot do so in the isothermal model (by definition). Both models take into account altitude changes and their impact on the air density.

Conclusion

We have observed that three of the models provide significantly different results for the same input parameters. Our observations from numerous simulations of a projectile launched from ground level follow:

- No-drag models provide a velocity that is higher than those provided by any of the other three.
- Constant drag (assuming a constant density at all parts of the cannonball's flight) is extremely similar to adiabatic drag, with the impact y-velocities at a starting velocity of 50 m/s differing by 0.006%.
- Isothermal drag differs greatly from all of the other models.
- Divergence occurs for almost all pairs of models between 20 m/s and 30 m/s.

Further investigations would entail performing actual physical experiments in order to verify our computational results. Another possible project would involve computationally determining the importance of relativistic effects in the cannonball's flight; however, an issue would be that these effects are so small that they are below the level of precision that can be handled by a normal household computer.