# Comparing the accuracy of analytical and numerical methods for predicting a model rocket's apogee using 1-dimensional kinematics

To what extent are numerical methods more accurate than an analytical method at predicting a model rocket's apogee using 1-dimensional kinematics?

**Rishabh Vakil**

Mathematics

3324 words

# Table of Contents

## Introduction

In the past year, I have developed an increasing interest in model rocketry and aerospace engineering alongside my participation in the American Rocketry Challenge (ARC). One of the objectives of the competition is to build a model rocket that will fly as close as possible to a target peak altitude (apogee). While the most trustworthy way to test if a rocket meets this objective is to fly it, repeated testing is costly and eventually wears out the rocket; hence the need for accurate pre-flight testing using computer simulations such as OpenRocket.

Of course, the OpenRocket apogee prediction often deviates from real-world testing results, but it can be quite close. After reading its technical documentation, I found that OpenRocket uses the Runge-Kutta fourth order method (RK4) to approximate the rocket's trajectory, and thus apogee [1]. Numerical methods are generally computation-heavy and carried out using computers, which were not widely available when model rocketry first became popular in the 1950s. Indeed, it is possible to predict a model rocket's apogee using an analytical method, which was published in Estes's "TR-10 Model Rocket Technical Report" in 1967 [2].

These findings prompted me to explore why numerical approximation has largely replaced earlier methods in apogee prediction, beyond its ease of implementation in computer simulation.

## Background and Aim

### *Approaches*

In this essay, I aim to answer the question: To what extent are numerical methods more accurate than an analytical method at predicting a model rocket's apogee using 1-dimensional kinematics? Before answering the question, I will first provide background to both clarify the physical situation I am modeling, and to set up a later discussion of the advantages and limitations inherent to each method. To answer my question, I will derive each method in the context of modeling the physical situation. Doing so will naturally lead to a discussion of the advantages and limitations inherent to each method and how they may affect the accuracy of predictions. Lastly, I will compare the accuracy of the analytical method to two numerical methods of varying order, and thus accuracy and computation time (explicit Euler and Runge-Kutta Fourth Order [RK4]). I will do this by using the same initial conditions and variables to model the rocket's flight and then comparing the results to data collected by on-board electronics during an actual flight. The deviation of a predicted altitude versus time graph from the actual graph will serve as a metric for the accuracy of a particular method, along with the deviation of a predicted apogee from the actual apogee. Note that while the focus is on apogee prediction, having only one data point to judge a method by is ineffective, hence the decision to make use of the entire set of data available when comparing results.

*Background*

This essay seeks to compare the accuracy of two numerical methods (explicit Euler and RK4) versus an analytical method in predicting a model rocket's apogee using one-dimensional kinematics. Unlike popular software such as OpenRocket and RockSim, we disregard the effects of wind because it is a stochastic process whose modelling is beyond the scope of this essay.

We only consider the burn and coast phases of the flight, since we are only interested in predicting the apogee—whatever happens in the landing and recovery phase is not relevant to the aim. Only three forces act on the rocket during the burn/coast phases: thrust ($T$), weight ($mg$), and aerodynamic drag ($D = 1/2\ C_D A \rho v^2$).

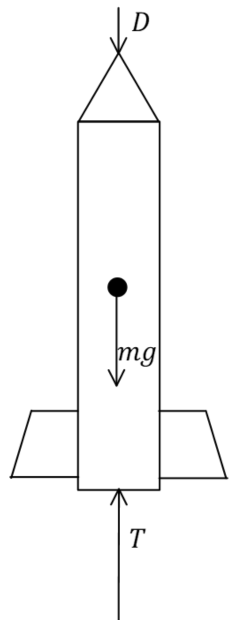


Figure 1: free body diagram of a model rocket in the burn/coast phases (going up).

From this, we can use Newton's Second Law to set up the appropriate differential equation that models the rocket's flight [3]:

$$F_{net} = m\frac{d^2s}{dt^2}$$

$$T - mg - \frac{1}{2}C_D A\rho\left(\frac{ds}{dt}\right)^2 = m\left(\frac{d^2s}{dt^2}\right), s(0) = 0, \frac{ds}{dt}(0) = 0 \qquad (1)$$

by noting that the velocity $v$ and acceleration $a$ are the first and second time-derivatives of the altitude $s$, respectively. The rocket starts on the ground and at rest.

Hence, we are presented with a second order, nonlinear, initial value, ordinary differential equation. Note that multiple variables in this equation are not constants and vary with respect to $t$. Equation (1) was solved analytically in 1967 by holding the variables $T, m, \rho, C_D$ constant throughout the entire burn or coast phase [3]. The resulting *Fehskens-Malewicki equations* [2] provide an exact closed-form solution to predict the rocket's altitude and velocity as a function of time. Equation (1) can also be modeled using numerical methods like Eulerian approximation and RK4, which allow for changing variables.

We are interested in how $s$ varies with respect to $t$. Before attempting to solve this equation, we must first consider how each variable changes.

**Variables**

$T$ changes during the burn phase according to the motor's *thrust curve*, which is a function of the thrust force with respect to the time and nonzero for $0 < t < t_B$ where $t_B$ is the time of motor burnout. During the coast phase, we always have $T = 0$. Thrust curves are published by the National Association of Rocketry (NAR) and can be found online.

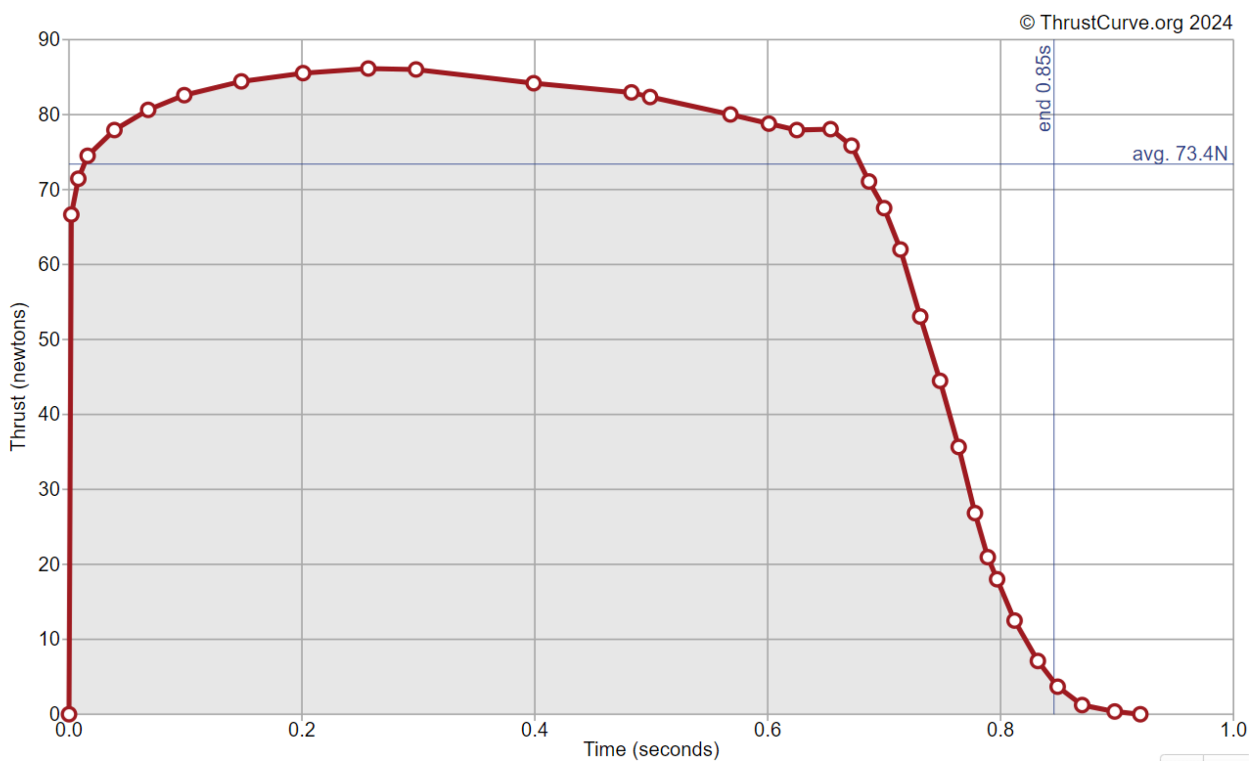


Figure 2: thrust curve of an F67-9W motor, manufactured by Aerotech [4].

The rocket mass $m$ decreases during the burn phase as propellant is burned. Since the propellant mass flow rate is directly proportional to the thrust, we additionally have $\frac{dm}{dt} \propto T$ [5]. $m$ remains constant during the coast phase.

The acceleration due to gravity $g$ remains virtually constant throughout the flight because the rocket remains close to the surface of the earth at all times. Thus, we treat $g = 9.8$ m s$^{-1}$ as a constant.

Likewise, the coefficient of drag $C_D$ remains virtually constant for a subsonic flight such as this one [6]. The reference area $A$ is the cross-sectional area of the rocket's body and remains constant. The air density $\rho$ decreases as a rocket flies higher, but the difference is small (less than 5%) for a height of around 1,000 feet [7]. It can be approximated as constant, but the best model would take this variable into account.

By now it should be apparent that equation (1) is difficult to solve exactly because multiple variables are changing with respect to each other.

Analytical Method[1]

Let us restate equation (1):

$$T - mg - \frac{1}{2}C_D A\rho\left(\frac{ds}{dt}\right)^2 = m\left(\frac{d^2s}{dt^2}\right), s(0) = 0, \frac{ds}{dt}(0) = 0$$

Since we have set the variables $T, m, \rho, C_D$ constant, we can simplify this equation by setting $W = mg$ and $K = \frac{1}{2}C_D A\rho$ as constants. Then we have

$$T - W - Kv^2 = \frac{W}{g}\frac{dv}{dt}, v = \frac{ds}{dt} \tag{2}$$

[1] The following derivation was adapted from [2].

where $T$ is equal to the average thrust during the burn phase and $T = 0$ during the coast phase; $W$ is equal to the average of the "wet" (with propellant mass) and "dry" (without propellant mass) weights during the burn phase (as propellant is consumed during the burn phase) and equal to the dry weight during the coast phase; and $\rho$ is set constant to the air density at the launch site. Note that we have now converted this into a system of two first-order differential equations. This way, we can first solve for velocity and then integrate to find an altitude versus time function. Importantly, this approach makes solving the differential equation easier. Let us first consider the burn phase.

To solve this equation using separation of variables, we must isolate $t$ and $v$ on opposite sides. Multiplying both sides by $\frac{dt}{T-W-Kv^2}$ we have

$$dt = \frac{W}{g}\frac{dv}{T - W - Kv^2}.$$

Integrating the LHS involving the differential $dt$ from the initial time $t_0$ to some time $t$, and integrating the RHS involving the differential $dv$ from the initial velocity $v_0$ to velocity $v$ at time $t$ yields

$$\int_{t_0}^{t} dt = \frac{W}{g}\int_{v_0}^{v}\frac{dv}{T - W - Kv^2}.$$

We can integrate the more complicated RHS by noting that

$$\int \frac{dU}{a^2 - b^2U^2} = \frac{1}{ab}\tanh^{-1}\left(\frac{bU}{a}\right).$$

We then substitute the following:

$$\begin{aligned} a^2 &= T - W \\ b^2 &= K \\ U &= v \end{aligned}$$

to yield

$$t - t_0 = \frac{W}{g}\frac{1}{\sqrt{(T-W)K}}\tanh^{-1}\left(\sqrt{\frac{K}{T-W}}v\right) - \frac{W}{g}\frac{1}{\sqrt{(T-W)K}}\tanh^{-1}\left(\sqrt{\frac{K}{T-W}}v_0\right).$$

If we let $t_0 = 0$, we have $v_0 = v(t_0) = 0$, and the equation simplifies to

$$t = \frac{W}{g}\frac{1}{\sqrt{(T-W)K}}\tanh^{-1}\left(\sqrt{\frac{K}{T-W}}v\right) \tag{3}$$

since $\tanh^{-1} 0 = 0$.

This equation is not very useful in its present form but isolating $v$ will allow us to determine the rocket's velocity. To make this process simpler, we can define constants $a_0$ and $\beta_0$:

$$a_0 = \frac{T}{W} - 1$$
$$\beta_0 = \frac{W}{K}$$

We can substitute these constants into equation (3) after a bit of algebraic manipulation:

$$\begin{aligned}\frac{W}{g}\frac{1}{\sqrt{(T-W)K}} &= \frac{1}{g}\sqrt{\frac{W^2}{(T-W)K}} \\ &= \frac{1}{g}\sqrt{\frac{W}{K}\frac{1}{\left(\frac{T-W}{W}\right)}} \\ &= \frac{1}{g}\sqrt{\frac{W}{K}\frac{1}{\left(\frac{T}{W}-1\right)}} \\ &= \frac{1}{g}\sqrt{\frac{\beta_0}{a_0}},\end{aligned}$$

also

$$\sqrt{\frac{K}{T-W}} = \sqrt{\frac{K/W}{(T-W)/W}}$$
$$= \sqrt{\frac{1/\beta_0}{\frac{T}{W}-1}}$$
$$= \sqrt{\frac{1}{\beta_0 a_0}}$$
$$= \frac{1}{\sqrt{\beta_0 a_0}}.$$

Thus, equation (3) becomes

$$t = \frac{1}{g}\sqrt{\frac{\beta_0}{a_0}}\tanh^{-1}\left(\frac{v}{\sqrt{\beta_0 a_0}}\right).$$

Noting that the identity form $M = \tanh^{-1}(N)$ means that $N = \tanh(M)$ yields

$$v = \sqrt{\beta_0 a_0}\tanh\left(g\sqrt{\frac{a_0}{\beta_0}}t\right) \tag{4}$$

which gives the velocity during the burn phase $0 \leq t \leq t_B$.

Unfortunately, equation (4) becomes useless for the coast phase, when there is no thrust ($T = 0$). This means that $a_0 = -1$, so that equation (4) no longer returns a real number. Instead, we can go back to equation (2), set $T = 0$, and solve again. Then we have

$$-W - Kv^2 = \frac{W}{g}\frac{dv}{dt}.$$

Using separation of variables as we did before, we multiply both sides by $\frac{dt}{-W-Kv^2}$ to yield

$$dt = -\frac{W}{g}\frac{dv}{W+Kv^2}.$$

Integrating the LHS involving the differential $dt$ from the time of burnout $t_B$ to some time $t$, and integrating the RHS involving the differential $dv$ from the burnout velocity $v_B$ to velocity $v$ at time $t$ yields

$$\int_{t_B}^{t} dt = -\frac{W}{g}\int_{v_B}^{v} \frac{dv}{W + Kv^2}.$$

We can integrate the more complicated RHS by noting that

$$\int \frac{dU}{a^2 + b^2U^2} = \frac{1}{ab}\tan^{-1}\left(\frac{bU}{a}\right)$$

and substituting

$$\begin{aligned} a^2 &= W \\ b^2 &= K \\ U &= v \end{aligned}$$

to yield

$$t - t_B = -\frac{W}{g}\frac{1}{\sqrt{WK}}\tan^{-1}\left(\sqrt{\frac{K}{W}}v\right) + \frac{W}{g}\frac{1}{\sqrt{WK}}\tan^{-1}\left(\sqrt{\frac{K}{W}}v_B\right). \tag{5}$$

Again, we can make some substitutions using $a_0$ and $\beta_0$ to help us isolate $v$:

$$\begin{aligned} \frac{W}{g}\frac{1}{\sqrt{WK}} &= \frac{1}{g}\sqrt{\frac{W^2}{WK}} \\ &= \frac{1}{g}\sqrt{\frac{W}{K}} \\ &= \frac{1}{g}\sqrt{\beta_0}, \end{aligned}$$

also

$$\sqrt{\frac{K}{W}} = \sqrt{\frac{1}{\beta_0}}$$
$$= \frac{1}{\sqrt{\beta_0}}.$$

Thus, equation (5) becomes

$$t - t_B = -\frac{1}{g}\sqrt{\beta_0}\tan^{-1}\left(\frac{v}{\sqrt{\beta_0}}\right) + \frac{1}{g}\sqrt{\beta_0}\tan^{-1}\left(\frac{v_B}{\sqrt{\beta_0}}\right)$$

and we are now ready to isolate $v$:

$$\frac{g(t - t_B)}{\sqrt{\beta_0}} = -\tan^{-1}\left(\frac{v}{\sqrt{\beta_0}}\right) + \tan^{-1}\left(\frac{v_B}{\sqrt{\beta_0}}\right)$$
$$\tan^{-1}\left(\frac{v}{\sqrt{\beta_0}}\right) = \tan^{-1}\left(\frac{v_B}{\sqrt{\beta_0}}\right) - \frac{g(t - t_B)}{\sqrt{\beta_0}}$$
$$v = \sqrt{\beta_0}\tan\left(\tan^{-1}\left(\frac{v_B}{\sqrt{\beta_0}}\right) - \frac{g(t - t_B)}{\sqrt{\beta_0}}\right). \tag{6}$$

Equation (6) is valid for $t_B \leq t \leq t_A$ where $t_A$ is the time of apogee (which can be obtained by setting $v = 0$). One can verify that the velocity curve is continuous at $t_B$ by setting $t = t_B$ and finding that equation (6) yields $v = v_B$.

Also, let us find $t_A$, which will be useful when finding the altitude functions:

$$\sqrt{\beta_0}\tan\left(\tan^{-1}\left(\frac{v_B}{\sqrt{\beta_0}}\right) - \frac{g(t_A - t_B)}{\sqrt{\beta_0}}\right) = 0$$
$$\tan\left(\tan^{-1}\left(\frac{v_B}{\sqrt{\beta_0}}\right) - \frac{g(t_A - t_B)}{\sqrt{\beta_0}}\right) = 0$$
$$\tan^{-1}\left(\frac{v_B}{\sqrt{\beta_0}}\right) - \frac{g(t_A - t_B)}{\sqrt{\beta_0}} = 0$$
$$\tan^{-1}\left(\frac{v_B}{\sqrt{\beta_0}}\right) = \frac{g(t_A - t_B)}{\sqrt{\beta_0}}$$
$$t_A = \frac{\sqrt{\beta_0}}{g}\tan^{-1}\left(\frac{v_B}{\sqrt{\beta_0}}\right) + t_B \tag{7}$$

In summary, we have

$$v = \begin{cases} \sqrt{\beta_0 a_0} \tanh\left(g\sqrt{\frac{a_0}{\beta_0}}t\right) & 0 \le t \le t_B \\ \sqrt{\beta_0} \tan\left(\tan^{-1}\left(\frac{v_B}{\sqrt{\beta_0}}\right) - \frac{g(t - t_B)}{\sqrt{\beta_0}}\right) & t_B \le t \le t_A. \end{cases}$$

Now that we have a piecewise function for the velocity during the burn and coast phases, we can integrate to find an altitude versus time function that can predict the rocket's apogee.

For the burn phase, we have $ds = \sqrt{\beta_0 a_0} \tanh\left(g\sqrt{\frac{a_0}{\beta_0}}t\right) dt$ since $v = ds/dt$ . This is a first order differential equation that we can solve by integrating both sides (note that as before, the $s, t$ in the bounds are conceptually distinct from those in the integrand):

$$\int_{s_0}^{s} ds = \sqrt{\beta_0 a_0} \int_{t_0}^{t} \tanh\left(g\sqrt{\frac{a_0}{\beta_0}}t\right) dt$$

We can integrate the more complicated RHS by noting that

$$\int \tanh x \, dx = \ln \cosh x.$$

Substituting $x = g\sqrt{\frac{a_0}{\beta_0}}\, t$ and $dx = g\sqrt{\frac{a_0}{\beta_0}}\, dt$ yields

$$\int_{s_0}^{s} ds = \frac{\sqrt{\beta_0 a_0}}{g\sqrt{\frac{a_0}{\beta_0}}} \int_{t_0}^{t} \tanh\left(g\sqrt{\frac{a_0}{\beta_0}}t\right) g\sqrt{\frac{a_0}{\beta_0}}\, dt$$

$$s - s_0 = \frac{\beta_0}{g} \ln \cosh\left(g\sqrt{\frac{a_0}{\beta_0}}t\right) - \frac{\beta_0}{g} \ln \cosh\left(g\sqrt{\frac{a_0}{\beta_0}}t_0\right).$$

If we let $t_0 = 0$, we have $s_0 = s(t_0) = 0$, and the equation simplifies to

$$s = \frac{\beta_0}{g} \ln \cosh \left( g \sqrt{\frac{a_0}{\beta_0}} t \right) \tag{8}$$

since $\ln \cosh 0 = 0$.

For the coast phase, we have that

$$ds = \sqrt{\beta_0} \tan \left( \tan^{-1} \left( \frac{v_B}{\sqrt{\beta_0}} \right) - \frac{g(t - t_B)}{\sqrt{\beta_0}} \right) dt. \tag{9}$$

Note from equation (7) that

$$t_A - t_B = \frac{\sqrt{\beta_0}}{g} \tan^{-1} \left( \frac{v_B}{\sqrt{\beta_0}} \right)$$

so that after multiplying both sides by $\frac{g}{\sqrt{\beta_0}}$ we have

$$\tan^{-1} \left( \frac{v_B}{\sqrt{\beta_0}} \right) = \frac{g(t_A - t_B)}{\sqrt{\beta_0}}.$$

Substituting into equation (9), we have

$$ds = \sqrt{\beta_0} \tan \left( \frac{g(t_A - t_B)}{\sqrt{\beta_0}} - \frac{g(t - t_B)}{\sqrt{\beta_0}} \right) dt$$

$$ds = \sqrt{\beta_0} \tan \left( \frac{g}{\sqrt{\beta_0}} (t_A - t) \right) dt.$$

Now that the differential equation has been significantly simplified, we can integrate both sides:

$$\int_{s_B}^{s} ds = \sqrt{\beta_0} \int_{t_B}^{t} \tan \left( \frac{g}{\sqrt{\beta_0}} (t_A - t) \right) dt$$

We can integrate the more complicated RHS by noting that

$$\int \tan x \, dx = -\ln \cos x.$$

Substituting $x = \frac{g}{\sqrt{\beta_0}}(t_A - t)$ and $dx = -\frac{g}{\sqrt{\beta_0}}dt$ yields

$$\int_{s_B}^{s} ds = \sqrt{\beta_0} \cdot \frac{-\sqrt{\beta_0}}{g} \int_{t_B}^{t} \tan\left(\frac{g}{\sqrt{\beta_0}}(t_A - t)\right)\left(-\frac{g}{\sqrt{\beta_0}}dt\right)$$

$$s - s_B = -\frac{\beta_0}{g} \cdot -\ln\cos\left(\frac{g}{\sqrt{\beta_0}}(t_A - t)\right) - \left(-\frac{\beta_0}{g} \cdot -\ln\cos\left(\frac{g}{\sqrt{\beta_0}}(t_A - t_B)\right)\right)$$

$$s = \frac{\beta_0}{g}\ln\cos\left(\frac{g}{\sqrt{\beta_0}}(t_A - t)\right) - \frac{\beta_0}{g}\ln\cos\left(\frac{g}{\sqrt{\beta_0}}(t_A - t_B)\right) + s_B. \tag{10}$$

Now we have all the necessary tools to construct an equation for the apogee. Setting $t = t_A$ in equation (10), substituting the RHS of equation (8) evaluated at $t = t_B$ for $s_B$ in equation (10), we have the apogee:

$$s_A = \frac{\beta_0}{g}\ln\cosh\left(g\sqrt{\frac{a_0}{\beta_0}}t_B\right) - \frac{\beta_0}{g}\ln\cos\left(\frac{g}{\sqrt{\beta_0}}(t_A - t_B)\right)$$

since $\ln\cos 0 = 0$.

Substituting equation (7) for $t_A$ and simplifying, we have

$$s_A = \frac{\beta_0}{g}\ln\cosh\left(g\sqrt{\frac{a_0}{\beta_0}}t_B\right) - \frac{\beta_0}{g}\ln\cos\left(\tan^{-1}\left(\frac{v_B}{\sqrt{\beta_0}}\right)\right). \tag{11}$$

The dual trigonometric functions (the $\cos(\tan^{-1} x)$ term) suggest that some simplifying can be done. Consider the following right triangle:

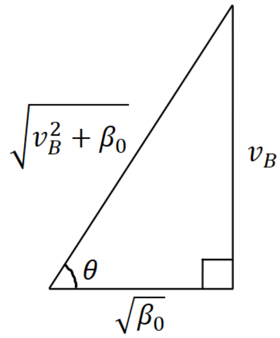


Figure 3

Notice that

$$\theta = \tan^{-1}\left(\frac{v_B}{\sqrt{\beta_0}}\right)$$

so that

$$\cos\theta = \cos\left(\tan^{-1}\left(\frac{v_B}{\sqrt{\beta_0}}\right)\right) = \frac{\sqrt{\beta_0}}{\sqrt{v_B^2+\beta_0}}.$$

Substituting into the second term on the RHS of equation (11) we have

$$\begin{aligned}
&-\frac{\beta_0}{g}\ln\cos\left(\tan^{-1}\left(\frac{v_B}{\sqrt{\beta_0}}\right)\right)\\
&= -\frac{\beta_0}{g}\ln\left(\frac{\sqrt{\beta_0}}{\sqrt{v_B^2+\beta_0}}\right)\\
&= -\frac{\beta_0}{g}\ln\left(\frac{1}{\sqrt{\frac{v_B^2}{\beta_0}+1}}\right)\\
&= -\frac{\beta_0}{g}\ln\left(\frac{v_B^2}{\beta_0}+1\right)^{-\frac{1}{2}}\\
&= \frac{\beta_0}{2g}\ln\left(\frac{v_B^2}{\beta_0}+1\right)
\end{aligned}$$

so that equation (11) simplifies to

$$s_A = \frac{\beta_0}{g}\ln\cosh\left(g\sqrt{\frac{a_0}{\beta_0}}t_B\right) - \frac{\beta_0}{2g}\ln\left(\frac{v_B^2}{\beta_0}+1\right). \tag{12}$$

Equation (12) is the simplest representation of a model rocket's apogee, derived using the analytical method.

Numerical Methods

While the analytical method certainly provides a relatively quick way to obtain a pen-and-paper estimation for the apogee, it requires that certain variables be set constant, unlike what is physically true.

Let us again consider the equation we want to solve and consider another approach. Restating equation (2) but taking out the constants and isolating the derivatives, we obtain the following system:

$$\begin{cases} \dfrac{ds}{dt} = v & (13) \\ \dfrac{dv}{dt} = \dfrac{1}{m}\left(T - mg - \dfrac{1}{2}C_D A \rho v^2\right) & (14) \end{cases}$$

Previously, we solved for $s$ as a function of $t$ by first solving equation (14) by integrating both sides, and then substituting the resulting function into the RHS of equation (13) and integrating both sides. The crux of the problem was integrating because we had to make several simplifications (setting variables constant) to be able to integrate.

While researching numerical methods for solving this particular system, I found a study that implemented a general technique known as *finite difference methods* [3]. This works by *discretizing the solution domain* (in our case this is the time $t$). Very simply, this means that the solution curve is redefined at only particular points in the domain separated by a small timestep $\Delta t$, creating a *discrete temporal grid* of points. Hence, given an initial condition at time $t_n$, the exact derivative the next point on the solution curve can be approximated using a technique known as *finite difference approximation (FDA)* which is then used to approximate the point on the solution curve corresponding to time $t_{n+1} = t_n + \Delta t$. This result replaces the original initial condition, and we then proceed to estimate the point at $t_{n+2}$, and so on in an iterative process. In our case, we would use known initial conditions starting with $n = 0$ and incrementing thereafter to iteratively estimate $v(t_{n+1})$. We would use our estimated velocity values to iteratively estimate $s(t_{n+1})$, hence eventually constructing an altitude vs. time graph for the model rocket.

However, this all sounds quite abstract. How does FDA work to approximate derivatives, when we only start with a complete set of information for one point (our initial conditions)?

The study I read used infinite Taylor series expansion as the FDA [3]. Consider the function $f(t)$, known at initial condition $f(t_0)$, and with derivatives $f'$, $f''$, etc. Then the Taylor series is

$$f(t) = f(t_0) + \frac{1}{1!}f'(t_0)(\Delta t) + \frac{1}{2!}f''(t_0)(\Delta t)^2 + \cdots.$$

Choosing $t_0 = n$ (the base point location in the discrete temporal grid) and $t = n + 1$ gives

$$f_{n+1} = f_n + f^t|_n \Delta t + \frac{1}{2} f^{tt}|_n \Delta t^2 + \cdots + \frac{1}{m!} f^{(m)}|_n \Delta t^m + \cdots. \tag{15}$$

Note our switch to sequence notation corresponding to a discrete set of points, where $f^{(m)}|_n$ signifies the $m$th derivative of $f$ with respect to $t$ at time $n$.

Solving equation (15) for $f^t|_n$ yields

$$f^t|_n = \frac{f_{n+1} - f_n}{\Delta t} - \frac{1}{2} f^{tt}|_n \Delta t^2 - \cdots.$$

Since evaluating an infinite series is not practical, we can truncate it, thereby introducing a *truncation error* of order $m$ when we truncate starting at the $(m + 1)$th term:

$$f^t|_n = \frac{f_{n+1} - f_n}{\Delta t} + O(\Delta t) \tag{16}$$

This is a *first-order accurate* FDA. The order of the truncation error $O(\Delta t)$ represents the rate at which the truncation error approaches zero as $\Delta t$ goes to zero. For example, if $\Delta t$ was halved, the truncation error in equation (16) would halve, while the truncation error of a fourth-order accurate FDA would decrease by a factor of $2^4 = 16$.

*Explicit Euler Method*

The explicit Euler method (also known by other similar names) is based off equation (16) and only requires the state at $t = t_n$ to estimate the state at $t = t_{n+1}$ (hence the term "explicit"). Suppose we have some derivative function for $v(t)$, $v' = f(t, y)$ with initial conditions $v(t_0) = v_0$. Then using the FDA in equation (16) at base point $n$ we have:

$$\frac{v_{n+1} - v_n}{\Delta t} + O(\Delta t) = f(t_n, y_n) = f_n$$
$$v_{n+1} - v_n \approx f_n \Delta t \;\; O(\Delta t)$$
$$v_{n+1} \approx v_n + f_n \Delta t \;\; O(\Delta t) \tag{17}$$

Equation (17) represents the explicit Euler method, which can be used iteratively to estimate the state at the next timestep. $f_n = v'$ can be updated for each iteration using equation (14). In this manner, an estimated velocity curve $v(t)$ can be constructed by iterating until $v(t_A) = 0$, which occurs after $n = t_A/\Delta t$ iterations.

We can again use the explicit Euler method to construct an estimated altitude curve $s(t)$:

$$s_{n+1} \approx s_n + v_n \Delta t \;\; O(\Delta t) \tag{18}$$

Using this method, it is possible to estimate the apogee without the need to keep variables constant, because their values can be updated at each timestep. Note that this is true even for variables that depend on $s$ (in our case, $\rho$, air density) since our method is explicit: the result for $s_n$ can be used to find $\rho_n$ to substitute into equation (14). The result ($f_n$) can be used to approximate $v_{n+1}$ using equation (17) which can then be used to approximate $s_{n+1}$ using equation (18).

*Runge-Kutta Fourth Order Method (RK4)*

Like the explicit Euler method, RK4 is also explicit. However, it differs in that RK4 is a *multistep* method. Intermediate points between $n$ and $n+1$ are introduced to obtain higher order accuracy. RK4 operates as follows [3]:

$$y_{n+1} = y_n + \frac{1}{6}(\Delta y_1 + 2\Delta y_2 + 2\Delta y_3 + \Delta y_4)\ \ O(\Delta t^4) \tag{19}$$

where

$$\begin{aligned} \Delta y_1 &= \Delta t f(t_n, y_n) & \Delta y_3 &= \Delta t f\left(t_n + \frac{\Delta t}{2}, y_n + \frac{\Delta y_2}{2}\right) \\ \Delta y_2 &= \Delta t f\left(t_n + \frac{\Delta t}{2}, y_n + \frac{\Delta y_1}{2}\right) & \Delta y_4 &= \Delta t f(t_n + \Delta t, y_n + \Delta y_3). \end{aligned}$$

I have chosen to not include the derivation of RK4 in this paper because it is lengthy and computation-heavy, and not especially relevant. Rather, we shall make use of this method to first estimate $v_{n+1}$ from $v_n$ and then to estimate $s_{n+1}$ from $s_n$ and $v_n$.

This method does not require that we keep variables constant, because they can be updated at each step. This holds true even for $\rho$ (which is used to calculate $\frac{dv}{dt}$ in equation (14)) which varies as a function of $s$ instead of $t$, since we can account for a changing $\rho$ using the procedure outlined at the end of the previous subsection. Again, this only works because RK4 is explicit.

Comparison of Methods

To answer my question, I must have a control data set that I can compare all methods against. I had initially decided to use flight data I collected over the past year, which includes data like the apogee, and details about the weather and the specifications of my model rocket that provides necessary information to use the three methods outlined above. However, I have instead elected to use data for one flight collected by one of my friends in preparation for the same rocketry competition because the rocket had onboard electronics which collected altitude, velocity, and acceleration data throughout the flight instead of just the apogee, allowing for better comparisons for multiple data points. Also, I made sure to carefully select a flight that had almost no wind, because the effects of wind were not considered in this paper. Having wind would significantly reduce the measured apogee of the rocket as compared to my predictions because the rocket would fly inclined to the vertical.

To test each method, I substituted the necessary information from the flight data into the relevant equations to obtain altitude predictions for each method. For each method, I used a script I wrote in Python to generate data with a timestep of 1 ms. To evaluate the accuracy of each method, I compared the results of each method to the actual values measured by the rocket's electronics (see Table 1 and Figure 4).[2]

| **Method** | **Apogee (m)** |
|---|---|
| Control (real) | 254.022 |
| Analytical | 257.365 |
| Explicit Euler | 260.542 |
| RK4 | 257.099 |

Table 1: comparison of apogee predictions with control.

[2] The programs used to implement each method, and the resulting raw data can be found online at https://github.com/rivak7/extended-essay.

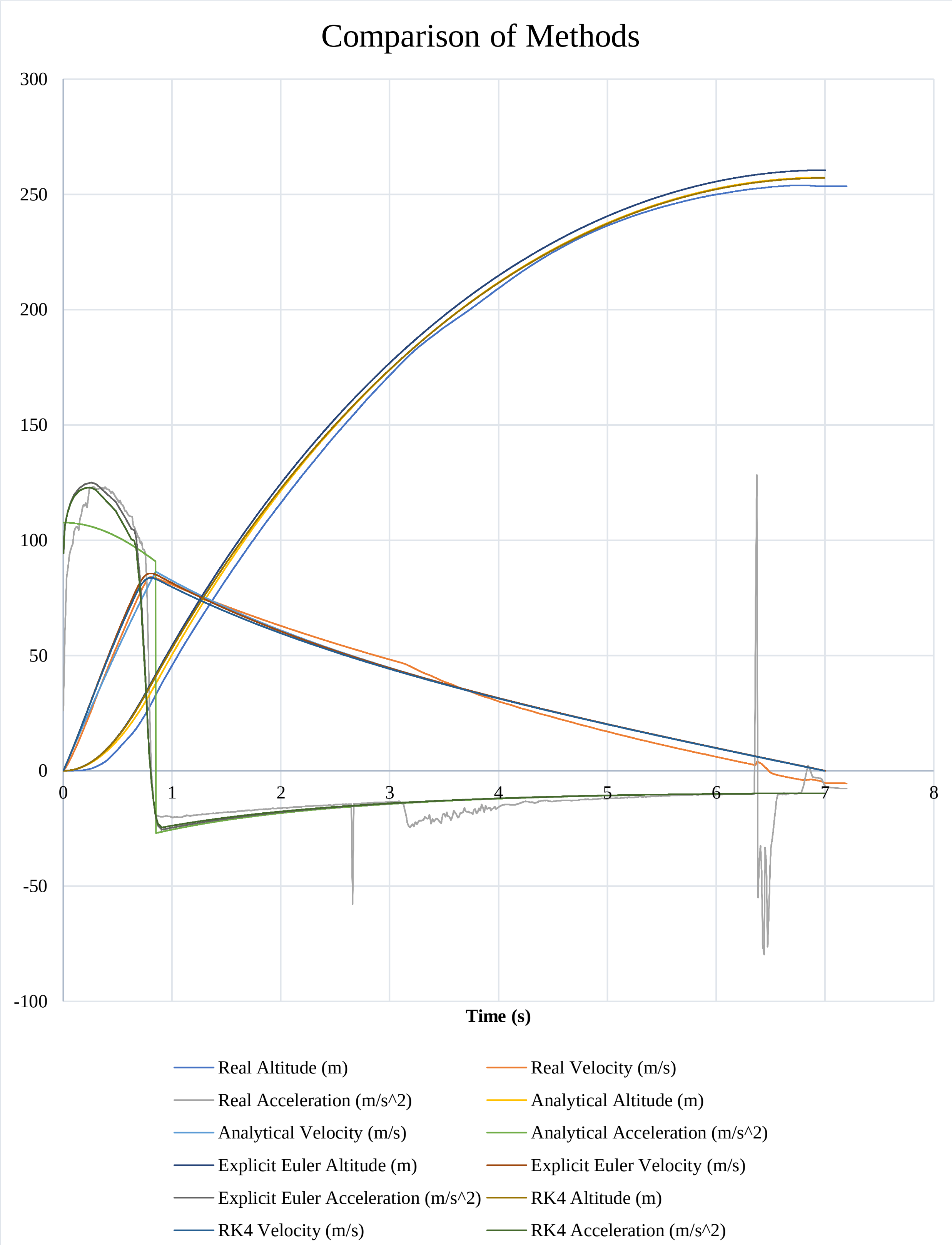


Figure 4: Graphs for each method and the control (real).

Evaluation and Conclusion

That the predictions are so close to the control data set is certainly an amazing result. All the predicted apogees are only a couple meters off from each other and the real value, meaning that all methods discussed here are quite accurate.

To answer my research question, it is surprising to conclude that numerical methods and the analytical method have very similar accuracies at predicting a model rocket's apogee. Given the major approximations involved in the analytical method, I was not expecting this. Presumably, these approximations lead to only small amounts of overall error because most of the error at each step ends up cancelling out with subsequent approximations. For example, while the "Analytical Acceleration" curve deviated the most from the real acceleration curve, integrating it led to much smaller errors in the velocity and altitude curves. Also note that the analytical method's apogee prediction was closer to the real apogee than that predicted by the explicit Euler method, but less close than RK4. This demonstrates the impact of the order of the numerical used on its accuracy. RK4 is widely used among engineers because of its relative simplicity, high order, and the flexibility that being explicit affords.

Of course, one major weakness of all three methods is that they neglect the influence of wind. Note that all three methods consistently overestimated not only the apogee but also the altitude at every other time. This demonstrates how neglecting the effects of even a little wind can introduce significant error. Hence, to make my methods for prediction more generally applicable, I would have to account for the effects of wind, using concepts in probability and statistics. I would also have to make my model multidimensional rather than just relying on one-dimensional kinematics, since wind has a component in all three axes.

It is my understanding that numerical methods have become more widespread in modeling physical situations such as this one because it is relatively easy to add increasing complexity to the model while keeping calculations simple. For example, I was able to adjust the mass of the rocket and air density after each timestep without making the computations significantly more complicated.

That said, both analytical and numerical methods are quite effective at predicting a model rocket's apogee. Their strengths and weaknesses lie in their ease of implementation. While numerical methods may seem unfamiliar, with difficult-to-grasp logic at first, they can easily be applied to more complex situations. For example, numerical methods could be easily implemented in a simulation of a rocket with active altitude control that varies its drag coefficient using canards or airbrakes. On the other hand, the analytical method is more familiar and easy-to-follow through simple steps in algebra and calculus that allow for a relatively quick and impressively accurate back-of-the-envelope apogee estimate. The downside lies in its limitations that make it more cumbersome to apply to more complex situations such as a rocket with active altitude control. In this paper we broke each analytical function into two piecewise functions for the burn phase and the coast phase to deal with two different values for thrust. To model active altitude control using canards or airbrakes, where the drag coefficient could continuously change, we would need to have a different piecewise function for each value of the drag coefficient!

Works Cited

[1] S. Niskanen, "Development of an Open Source model rocket simulation software," M. S. thesis, Helsinki University of Technology, Espoo, Finland, 2009, p.66. [Online]. Available: https://openrocket.sourceforge.net/thesis.pdf. [Accessed: Sept. 7, 2024].

[2] Estes Industries, "Altitude Prediction Charts," Estes Industries, Penrose, CO, 1971, pp.37-39. [Online]. Available: https://www.oldrocketplans.com/pubs/Estes/estTR-10/TR-10.pdf. [Accessed: June 2, 2024].

[3] K. J. Karbon, "Numerical Methods for Model Rocket Altitude Simulation - A Comparative Study of Accuracy and Efficiency," in *NARAM 40 Research and Development Competition*, Aug. 1998, pp.1-15. [Online]. Available: https://www.apogeerockets.com/downloads/PDFs/numeric_methods.pdf. [Accessed: June 2, 2024].

[4] J. Coker, "AeroTech F67W," *thrustcurve.org*, Mar. 10, 2018. [Online]. Available: https://www.thrustcurve.org/motors/AeroTech/F67W/. [Accessed: Nov. 1, 2024].

[5] T. A. Ward, *Aerospace Propulsion Systems*, illustrated ed. Hoboken, NJ: John Wiley & Sons, 2010, p.68. [Online]. Available: https://books.google.com/books?id=KEPgEgX2BEEC&pg=PA68. [Accessed: Sept. 13, 2024].

[6] Glenn Research Center, "Drag Coefficient," *NASA,* July 19, 2024. [Online]. Available: https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/drag-coefficient/. [Accessed: Dec. 29, 2024].

[7] Engineering ToolBox, "Air - Density and Specific Volume vs. Altitude," 2001. [Online]. Available: https://www.engineeringtoolbox.com/air-altitude-density-volume-d_195.html. [Accessed: Sept. 8, 2024].