ESTIMATING ROCKET PARAMETERS FROM FLIGHT DATA

Geoffrey Cox¹, Erica Fields², Bo Li³, Sarah Lukens⁴, Xinli Wang⁵, Ke Xu⁶

Problem Presenter:
John Peach
MIT Lincoln Laboratories

Faculty Mentors:
Zhilin Li,
Kartik Sivaramakrishnan
North Carolina State University

Abstract

A rocket trajectory directed by a Lambert Guidance System is determined by several parameters such as initial and final positions and time, mass quantities of the missile and specific impulse. To determine these initial parameters from an observed portion of the rocket’s flight, we developed three approaches in order to obtain good approximations of them. This led to the question of ill-posedness of the inverse problem, in which we proved that an initial trajectory during the thrusting phase does not determine a total unique trajectory based on data before shut down time. We then developed an alternative method to compute the rocket’s trajectory using the original parameters dependent on the angle of the thrust and final time based on solving a system of equations rather than solving an ODE.

1 Introduction

The motion of a rocket is determined by a number of parameters affecting the forces acting on its body. Given a time history of the altitude of the rocket, we would like to estimate these parameters, and ultimately predict characteristics of the remainder of the flight. It is desired to have an algorithm that estimates these parameters over the fewest number of data points to be able to predict the altitude, velocity and acceleration until motor burnout, and to have an estimate of the accuracy with which they may be predicted.

1.1 The Forces acting on a Rocket

The forces on a rocket are thrust, drag, lift, and weight [2]. For this problem, we will assume exo-atmospheric flight only so the lift and drag forces will be zero.

The force due to the weight $F_w$ is given by

$$F_w = mg,$$

where $m$ is the mass of the rocket and $g$ is the force due to gravity.

The thrust is the force which moves the rocket forward. It is the reaction from Newton’s Third Law to the acceleration of the engine. The governing equation for the force due to thrust $F_T$ is given by

$$F_T = I_{SP} \dot{m} g_0,$$

where $\dot{m}$ is the mass flow rate, $g_0$ is the gravitational constant, and $I_{SP}$ is the specific impulse, measured in seconds. The specific impulse characterizes the efficiency of the system, i.e. it is the ratio of the amount of thrust produced to weight flow of the propellants.

¹University of California-Irvine
²University of Georgia
³University of Toledo
⁴Tulane University
⁵New Jersey Institute of Technology
⁶University of North Carolina-Chapel Hill
Figure 1: Free Body Diagram of Forces acting on a Rocket, obtained from the NASA web site [2].

The total force acting on the rocket is the sum of all the forces acting on the rocket, and the acceleration can be found by dividing this total force by the instantaneous mass \( m_0 - \dot{m}t \)

\[
a = \frac{\Sigma F}{m} = \frac{I_{sp} \dot{m} g_0}{m_0 - \dot{m} t} - g,
\]

where \( m_0 \) is the initial mass of the rocket at launch, and \( \dot{m} = dm/dt \) is the rate of change of mass as the rocket flies. The mass changes as the engine expels exhaust gases during the thrusting period, and is a constant quantity until the engine burns out.

For a long range rocket, the rocket is outside of the Earth’s atmosphere for most of its flight, so we can consider the atmospheric effects to be negligible.

1.2 The Lambert Guidance System

Given the initial and final positions of a rocket along with the total time desired to travel this distance, Lambert’s problem is to find the initial velocity so that the rocket will indeed end up at the final position at the final time from its starting position [1].

We consider this problem in the case of the Flat Earth Model; that is, we ignore factors such as the curvature of the Earth, and consider the two-dimensional, point mass missile geometry. In this case, the final positions \((x_f, y_f)\) of the rocket are given by

\[
x_f = x(t) + x_D (t_f - t),
\]

\[
y_f = y(t) + y_D (t_f - t) - \frac{1}{2} g (t_f - t)^2
\]

where \((x_D, y_D)\) are the desired velocities from Lambert’s problem. Solving for the desired velocities from equation 5 gives

\[
x_D = \frac{x_f - x(t)}{t_f - t},
\]

\[
y_D = \frac{y_f - y(t)}{t_f - t} - \frac{1}{2} g (t_f - t)
\]
1.2.1 Booster Steering

In practice, the actual velocity of the rocket does not always match up with the desired velocity found by solving Lambert's problem. Since we do not have impulsive missiles (missiles that can speed up and slow down at will), we need to adjust the steering so that the rocket travels on the desired course; this is known as Lambert Guidance.

We introduce the error $\Delta v$ between the actual velocity, or missile thrust vector, and the desired velocity at a time $t$

$$\Delta v = v - v_D$$

where $v = (\dot{x}, \dot{y})$ is the velocity of the missile, and $v_D = (\dot{x}_D, \dot{y}_D)$ is the desired velocity.

For our rocket to land at the desired place at the desired time, we need to align the actual velocity with the desired velocity to be gained, i.e., we want to make $\Delta v = 0$ or to some sufficiently small value. Then, the engine cuts off and the missile flies ballistically to its intended target.

To align the velocity to the desired velocity, the direction of the rocket’s acceleration should be pointing in the direction of this error $\Delta v$. The magnitude of the acceleration $a_{mag}$ is known from the dividing the total forces by the instantaneous mass, as in 3.

$$a_x = a_{mag} \frac{\Delta v_x}{||\Delta v||}$$

$$a_y = a_{mag} \frac{\Delta v_y}{||\Delta v||}$$

1.3 Computing the Missile’s Trajectory under Lambert Guidance

To compute the trajectory of the missile using Lambert Guidance, we begin with initial data $(x_0, y_0, \dot{x}_0, \dot{y}_0)$, final data $(x_f, y_f, t_f)$, specific impulse $I_{SP}$, the initial mass of the rocket $m_0$ and the rate of change of the mass during the thrusting phase $\dot{m}$. This rate of change is constant, as the rocket expels spent fuel linearly during this portion of the flight.

To compute the position and velocity of the trajectory at each time step over the duration of the flight, an ODE solver is implemented on MATLAB applied to the velocity and acceleration subject to homogeneous initial conditions. After each integration step, the desired velocity of the rocket is obtained in a feedback fashion over time until $\Delta v$ is sufficiently small. The trajectory ends when $y_f = 0$, or equivalently, when the rocket lands.

2 The Problem

Given the parameters $(x_0, y_0, \dot{x}_0, \dot{y}_0, x_f, y_f, t_f, I_{SP}, m_0, \dot{m})$, we can compute the trajectory of the rocket under Lambert Guidance.

From this knowledge, the problem becomes how to compute these parameters given information from just a piece of the rocket’s trajectory during the thrusting phase of its flight, and thus computing the entire trajectory. The data given is the position and time of the piece of the trajectory known during the thrusting phase of the rocket’s flight $(x_i, y_i, t_i)_{i=1}^n$ at $n$ points in the time interval $[t^0, t^*]$. Here, we consider the time interval beginning at $t = 0$ and ending at some final time $t^* < t_{bo}$, where $t_{bo}$ is the burnout time when the engine cuts out, reducing the number of parameters to not include the initial conditions $(x_0, y_0)$.

To compute these parameters, we try different methods and compare results. We use data from different one-stage rockets [3] to test our methods.

2.1 Reduction of Parameters

From observation of the ODE system for updating the rocket trajectory, we found the magnitude of the acceleration can be decoupled from the system, i.e.

$$a_{mag} = \frac{I_{SP} g}{M - t}$$
<table>
<thead>
<tr>
<th>Rocket Type</th>
<th>$x_f$</th>
<th>$y_f$</th>
<th>$t_f$</th>
<th>$I_{SP}$</th>
<th>$m_0$</th>
<th>$m$</th>
<th>$t_{NO}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iranian Shahab-1</td>
<td>$300 \times 10^3$</td>
<td>0</td>
<td>313</td>
<td>230</td>
<td>5900</td>
<td>3765/63</td>
<td>100</td>
</tr>
<tr>
<td>NO-DONG-B</td>
<td>$300 \times 10^3$</td>
<td>0</td>
<td>313</td>
<td>269</td>
<td>19000</td>
<td>100</td>
<td>180</td>
</tr>
<tr>
<td>Shahab-5</td>
<td>$300 \times 10^3$</td>
<td>0</td>
<td>313</td>
<td>230</td>
<td>55848</td>
<td>3765/63</td>
<td>112</td>
</tr>
<tr>
<td>Shahab-5A</td>
<td>$300 \times 10^3$</td>
<td>0</td>
<td>313</td>
<td>230</td>
<td>55856</td>
<td>3765/63</td>
<td>112</td>
</tr>
<tr>
<td>NO-DONG-A</td>
<td>$300 \times 10^3$</td>
<td>0</td>
<td>313</td>
<td>230</td>
<td>15000</td>
<td>3765/63</td>
<td>115</td>
</tr>
</tbody>
</table>

Table 1: Parameters for Various Rockets

<table>
<thead>
<tr>
<th>$I_{SP}$</th>
<th>Initial Guess</th>
<th>True Parameters</th>
<th>Computed Parameters</th>
<th>Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>100</td>
<td>230</td>
<td>229.999950</td>
<td>&lt; 0.1%</td>
</tr>
<tr>
<td>M</td>
<td>10</td>
<td>$\approx 98.735$</td>
<td>98.725065</td>
<td>&lt; 0.1%</td>
</tr>
</tbody>
</table>

Table 2: True and Computed Parameters by Method of Reduction

and so by curve fitting the acceleration, we would be able to estimate just $(I_{SP}, M)$ first without the data of the rocket’s final time and position. Using $a_{mag} = a_x^2 + (a_y + g)^2$, we compute $a_{mag}$ from the trajectory of the rocket, using the finite difference estimates from the known data to express $a_{mag}$, the acceleration magnitude corresponding with data at time $t_f$.

In other words, we want to find $X = (I_{SP}, M)$ that minimizes

$$
F(I_{SP}, M) = a_x^2 + (a_y + g)^2 - \frac{I_{SP} g}{M-t_f}.
$$

(12)

These parameters can thus be computed independent of the final data of the rocket’s trajectory.

We computed $(I_{SP}, M)$ by using the nlinfit command on Matlab. The relative error of the parameters was less than 1%, see Table 2.

3 Approach 1: Predicting Shut-Down Time

Having an estimate for $(I_{SP}, M)$, we search for ways to predict the final data $(x_f, t_f)$. Recall, under Lambert Guidance, the velocity is updated in a feedback fashion to align in direction of desired velocity by accelerating the rocket in the direction of the error $\Delta v = v - v_D$, and when $\Delta v$ is sufficiently small, the engine shuts off. Since this value is obtained at each time step using updated information from the prior time step, it is not intuitive that this value could be modeled and used over the trajectory. However, looking at plots of $\Delta v$ over time for various rocket parameters shows that perhaps it could be modeled quadratically, or by some fitting polynomial.

The idea for this approach is to interpolate the error $\Delta v$, and to use the interpolating function to predict cut off time and location. From this information, we can predict where and when the rocket will land from basic physics.

From the data $(x_i, y_i, t_i)_{i=1}^n$, we can’t approximate $(\Delta v_x, \Delta v_y, t_i)_{i=1}^n$ exactly, because knowing $\Delta v$ as well as knowing $v$ would mean we, as an observer, would exactly know $v_D$, the Lambert Velocity, which we have no way of knowing. Figure 3 shows the computed results for the final parameters using this method.

To get around this, we use other features of the problem that we do know. Using the approximation for acceleration, we can compute its direction, which is also the same direction at the correction velocity $\Delta v$, although we have no way of knowing anything about the magnitude. From experimental trials, we observed that this angle is constant for all time during the thrusting period.

Also, we do know that as the time of the flight goes to burnout time, but since $\Delta v$ approaches zero, the angle between the velocity and $\Delta v$ should also go to zero. The next idea is to compute the velocity and the direction of the acceleration of the data points, and then try to compute an expression for this angle to polyfit and approximate the roots to this expression get burnout time.

However, since shut off time is not exactly when $\Delta v$ equals zero, but at some sufficiently small time, the angle is not sufficiently close to zero.
Figure 2: Velocity error $\Delta v$ over time in $(x, y)$ directions.

<table>
<thead>
<tr>
<th></th>
<th>Computed</th>
<th>Actual</th>
<th>Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{SP}$</td>
<td>59.12</td>
<td>58.51</td>
<td>0.01</td>
</tr>
<tr>
<td>$t_f$</td>
<td>312.70</td>
<td>313</td>
<td>0.001</td>
</tr>
<tr>
<td>$x_f$</td>
<td>29943.81</td>
<td>30000</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Table 3: Computed Parameter Results using Shut-Down Predictor Method for final data

3.1 Approach 2: The Fitting Method

An alternative approach to finding the rocket’s parameters $(I_{SP}, \dot{m}, m_0, x_f, y_f, t_f)$ and the entire flight path from data of a portion of the trajectory from $t = 0$ to $t = t^*$ is to treat the problem as that of a nonlinear least squares problem, where data $(x_i, y_i)_{i=1}^n$ is fitted with the model $f(I_{SP}, \dot{m}, m_0, x_f, y_f, t_f, t)$ that is nonlinear in $t$. We choose a residual function $r_i(t) = f(X, t) - (x_i, y_i)$, and try to choose $X = (I_{SP}, \dot{m}, m_0, x_f, y_f, t_f)$ so that the fit is as close as possible to minimize the sum of the least squares of the residual. In other words, we want to minimize

$$J(X) = \sum_{i=1}^n (f(X, t_i) - (x_i, y_i))^2 \tag{13}$$

over the parameters $X = (I_{SP}, \dot{m}, m_0, x_f, y_f, t_f)$.

First, we try to reduce the number of variables. Note that since the trajectory ends when the missile hits the ground, we know that $y_f = 0$. Furthermore, we also know that from the magnitude of the acceleration due to the thrusting force,

$$a_{mag}(t) = \frac{I_{SP} g}{M - t}, \text{ where } M = \frac{m_0}{\dot{m}}. \tag{14}$$

Since this is the only location of occurrence of $m_0$ and $\dot{m}$ in the computation of the trajectory, we can consider them coupled and solve for $M$ alone. Hence, we reduce the problem for minimizing over only four variables; $X = (I_{SP}, M, x_f, t_f)$.

To solve this, we use the built-in solver `nlinfit` in Matlab’s Statistics Toolbox, which fits nonlinear least squares data by a Gauss-Newton method. To compute the velocity and acceleration of the rocket during the known portion of the data, we use second order central finite differencing.
Figure 3: Polyfit of Trajectory using the Shut-Down Predictor Method from the NO-DONG-B rocket

<table>
<thead>
<tr>
<th>Initial Guess</th>
<th>True Parameters</th>
<th>Computed Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_f$</td>
<td>$4 \times 10^9$</td>
<td>$3 \times 10^9$</td>
</tr>
<tr>
<td>$t_f$</td>
<td>290</td>
<td>313</td>
</tr>
<tr>
<td>$I_{Sp}$</td>
<td>200</td>
<td>230</td>
</tr>
<tr>
<td>$M$</td>
<td>81</td>
<td>98.7251</td>
</tr>
</tbody>
</table>

Table 4: Results via method 1 from an Initial Guess

For the initial guess $X = [400000, 290, 200, 5900 \times 63/3765]$, in contrast to the actual model parameters $[300000, 313, 230, 5900 \times 63/3765]$, we get the predicted trajectories shown in figure 4.

Although the results demonstrate a relative error of 2 – 5% for the computed values of $(x_f, t_f)$, there are still a couple of problems. One problem is that while this may be an acceptable tolerance for many physical problems, it is not for rockets. The difference of final position, while only about a 2% relative error, can still be thousands of meters, so this would not be an accurate way to estimate where a rocket will land in practicality.

Another problem is that good results depend on good initial guesses, which may not be available in a realistic situation.

### 3.2 The Divide and Conquer Fitting Method

From the above discussion and methods, the natural question to ask is why we didn’t use the estimated parameters for $(I_{Sp}, M)$ computed in equation 12 and then solve the nonlinear least squares problem 13 for the remaining parameters $(x_f, t_f)$.

It is clear from figure 5 that the fitting is good enough near the known data (the relative error is $10^{-14}$ point-wise), but as time increases, the predicted trajectory falls off from the actual one, and the final estimated parameters are not that good. In this example, $x_f \approx 1.659e5$ and $t_f \approx 232.77$, (where the actual values are $3e10$ and $313$ for $x_f$ and $t_f$ respectively).

However, for a better initial guess $[330000, 280]$, 6 shows that the fit is better. Surprisingly, you’ll see with the same initial guess as our first one, the fit was better for the first model.

Studying the trajectories computing using this model, which look completely different, led to the question of uniqueness; is the rocket trajectory unique to its given data? The answer, after some analysis, showed that this is not the case.
Figure 4: Comparison of Predicted Data and Actual Data for Flight Trajectory using Fitting Method. Graphs depict altitude, ground truth, and velocity from left to right.

4 The Ill-Posedness of the Inverse Problem

The inverse problem investigated here of computing flight parameters from flight data \( (x_i, y_i, t_i)_{i=1}^n \) turns out to be ill-posed. Hints of why this may be so have popped up in the computing approaches, such as the fact the direction of \( \Delta v \) is constant found in attempting approach 1.

To prove why \( \Delta v \) is constant, we look at the angle at the first two time steps, when \( t = 0 \) and when \( t = \Delta t \) to derive an expression for the constant angle, and then use induction.

First, let’s prove that the direction of \( \Delta v \) is a constant.

1. At time \( t = 0 \),

\[
\begin{align*}
\Delta v_x(0) &= v_{xD}(0) - v_x(0) = \frac{x_f}{t_f}, \\
\Delta v_y(0) &= v_{yD}(0) - v_y(0) = \frac{1}{2gt_f}.
\end{align*}
\]

Define \( \theta = \cot^{-1}\left( \frac{x_f/1}{2gt_f} \right) = \cot^{-1}\left( \frac{2x_f}{gt_f^2} \right) \).

Since \( a_{mag} \) depends on \( t \) only, we can write \( a_{mag}(t) = a(t) \). At \( t = 0 \), \( a(0) = Isp \times g/M \). Then

\[
\begin{align*}
a_x(0) &= a(0) \cos \theta \\
a_y(0) &= a(0) \sin \theta - g.
\end{align*}
\]

Therefore,

\[
\begin{align*}
v_y(\Delta t) &= v_y(0) + a_y(0)\Delta t = a(\sin \theta - g)\Delta t \\
x(\Delta t) &= x(0) + v_x(0)\Delta t + 1/2a_x(0)\Delta t^2 = 1/2a \cos \theta \Delta t^2 \\
y(\Delta t) &= 1/2(a \sin \theta - g)\Delta t^2
\end{align*}
\]
2. At time \( t = \Delta t \),

\[
\begin{align*}
    v_x(t) &= \frac{x_f - x(t)}{t_f - t} = \frac{x_f - 1/2a \cos \theta \Delta t^2}{t_f - t} \\
    v_y(t) &= 1/2u(t_f - t) - \frac{a(t_f - t) - (a \sin \theta - g) \Delta t^2}{2(t_f - t)} - \frac{A \cos \theta + B \cos \theta}{A \sin \theta + B \sin \theta} = \cot^{-1} \theta \\

    \frac{\Delta v_x}{\Delta v_y} &= \frac{v_x((k+1)\Delta t) - v_x(k\Delta t)}{v_y((k+1)\Delta t) - v_y(k\Delta t)} = \cot^{-1} \theta.
\end{align*}
\]

where \( A = t_f \sqrt{v_x^2 + v_y^2} \) and \( B = a\Delta t(1/2\Delta t - t_f) \).

3. Assuming the direction of \( \Delta v \) is constant for \( k < n \) steps. At \( t = (n+1)\Delta t \), following the same step, we have

\[
\Delta v_x((k+1)\Delta t) = \Delta v_y((k+1)\Delta t) = \cot^{-1} \theta.
\]

Following all the steps listed above, you’ll see if \( x_f/t_f^2 \) is fixed, as well as \( I_{SP} \) and \( M \), different rockets will have the same trajectories before any of their engines shut down. To test, let \( x_f = 300000/0.64 \) and \( t_f = 313/0.8 \) and fix \( (I_{SP}, M) \), you’ll see the two rockets have the same trajectories before shut down time at \( t = 53 \) seconds.

Therefore, the solution to the inverse problem is not unique. However, the solver proves to be very good. The parameters obtained by nlinfit are \( [x_f, t_f, I_{SP}, M] = [316178.816, 321.329, 230.00, 98.725] \), from which \( x_f/t_f^2 = 3.06219314272882 \), while \( x_f/t_f^2 \) computed from the actual model is \( 300000/313^2 = 3.06219314272882 \).
5 An Alternative Approach

From our numerical experiments, we find out that the direction of thrust does not change. In this case, it is easy to set up the mathematical model without using Lambert Guidance model, which means we can find the trajectory of missile from the designer side without needing to solve the ODE system. Here gravity and thrust are the only forces. According to Newton’s Law, the accelerations are

\[ a_x = \frac{I_s p g}{M - t} \cos \alpha \]
\[ a_y = \frac{I_s p g}{M - t} \sin \alpha - g, \]

where \( \alpha \) is the angle between the thrust and ground. Integrating the above equations with the initial conditions that the velocities are zero at the beginning gives

\[ v_x(0) = 0, \]
\[ v_y(0) = 0. \]
We obtain the equations of velocity,

\[ v_x(t) = \cos \alpha \ast Isp \ast g \ln \frac{M}{M - t}, \]
\[ v_y(t) = \sin \alpha \ast Isp \ast g \ln \frac{M}{M - t} - gt. \]

Integrating the above equations again with the initial positions of missile \((0, 0)\), we obtain the positions of missile

\[ x(t) = \cos \alpha \ast Isp \ast g((t - M) \ln \frac{M}{M - t} + t), \]
\[ y(t) = \sin \alpha \ast Isp \ast g((t - M) \ln \frac{M}{M - t} + t) - \frac{1}{2}gt^2. \]  

(16)

All above equations describing the trajectory of missile is up to the shut down time \(t_{SD}\). After the shut down time, the motion of missile is quite simple. It is like throwing a ball with specific velocities at specific positions where the only force is gravity. If \(x_f\) is the final place we want to go and \(t_f\) is the time we want to spend, then

\[ x_f = x(t_{SD}) + v_x(t_{SD})(t_f - t_{SD}), \]
\[ y_f = y(t_{SD}) + v_y(t_{SD})(t_f - t_{SD}) - \frac{1}{2}g(t_f - t_{SD})^2, \]

(17)

where all values \(x(t_{SD}), y(t_{SD}), v_x(t_{SD})\) and \(v_y(t_{SD})\) can be evaluated from the equations (16) and (17), and \(y_f = 0\). Then the above equations become

\[ x_f = \cos \alpha \ast Isp \ast g((t_f - M) \ln \frac{M}{M - t_{SD}} + t_{SD}), \]
\[ 0 = \sin \alpha \ast Isp \ast g((t_f - M) \ln \frac{M}{M - t_{SD}} + t_{SD}) - \frac{1}{2}gt_f^2. \]

(18)

In the above system of equations, there are four parameters \(\alpha, t_{SD}, x_f\) and \(t_f\). If two parameters are known, then the other two will be solved quickly. For example, we assume \(x_f\) and \(t_f\) are known. We transform the equations (19) in such a way that \(t_{SD}\) and \(\alpha\) depend on \(x_f\) and \(t_f\) implicitly.

\[ x_f^2 + \left(\frac{1}{2}gt_f\right)^2 - Isp^2g^2((t_f - M) \ln \frac{M}{M - t_{SD}} + t_{SD}^2) = 0, \]
\[ x_f \tan \alpha - \frac{1}{2}gt_f^2 = 0. \]

(19)

Solving this system equations gives an alternate way of computing the rocket’s trajectory without solving an ODE. A comparison of the methods can be seen in figure 8. The error in the computed trajectory is due to the errors in the ODE solver, while any error in solving the system of equations is in the solver, which is much less and not dependent on a build up of error from previous time steps.

6 Future Work

Our model is not only useful for regular rockets, but also for multi-stage rockets or solid fuel rockets by assuming 90% of the motor mass is fuel. We can also use this to set an upper bound on the burnout time \(t_{BO}\), by setting \(t = 0.9M\). We can also set a lower bound for the computed \(x_f\) by assuming the motor shuts down the instant after the last observation in the known trajectory, at time \(t^*\). Setting upper and lower bounds may help to find the trajectory out of the set of solution trajectories that we want.
Figure 8: Computed Trajectory using Improved Method with Trajectory found by solving ODE

Figure 9: Blow up of difference in final positions between Improved method and ODE solved method
A major assumption that was made that is unrealistic is in not accounting for the curvature of the earth. Calculations for Lambert Guidance taking the curvature of the earth are readily available, see [1], so taking the next step should not be that difficult, and should be a natural progression. We also assumed gravity to be constant, while in actuality it changes with respect to distance from the Earth’s center and varies over the Earth’s surface due to local features such as the equatorial bulge. The expression for gravity with respect to altitude $h$ and the radius of the earth $Re$ is given by

$$g(h) = g_0 \left( \frac{Re}{Re+h} \right)^2. \quad (21)$$

Another consideration that may be taken into account in assuming the flight direction to be vertical are the forces lift and drag, which depend on local air density.

However, based on these results, we have a much better understanding of the problem, necessary parameters, and the importance of an initial guess to keep in mind if wanting to develop efficient ways to practically compute information on a rocket based on observing a portion of its flight.

References


