Computing Problem

Realistic Rocket Launch

Johnathan Corbin
661563006

Spaceflight Mechanics
Spring 2019
Rensselaer Polytechnic Institute
April 21, 2019

Contents

1 Nomenclature

Symbol	Description	Units
T	thrust generated by the rocket	N
D	drag on the rocket	N
m_R	mass of the rocket	kg
$\dot{m}$	mass flow rate of the engine	$\frac{kg}{s}$
t_{burn}	burn time of the engine	s
I_{sp}	specific impulse of the propellant	s
V_E	velocity of the ejecta	$\frac{m}{s}$
C_D	drag coefficient	-
d	diameter of the rocket	m
A_R	frontal area of the rocket	m^2
r_E	radius of the Earth	km
g	acceleration due to gravity	$\frac{m}{s^2}$
g_0	standard gravity	$\frac{m}{s^2}$
$\hat{\boldsymbol{t}}$	direction tangential to rocket's motion	-
$\hat{\boldsymbol{n}}$	direction normal to rocket's motion	-
x	downrange distance travelled	m
h	height of the rocket	m
h_0	standard height	m
μ	standard gravity parameter for Earth	$\frac{km^3}{s^2}$
θ	pitchover angle of the rocket	rad
ρ	air density	$\frac{kg}{m^3}$
ρ_0	air density at sea level	$\frac{kg}{m^3}$
P_E	pressure of the ejecta	Pa
P_A	pressure of the atmosphere	Pa
A	area of the nozzle	m^2
m_L	payload mass	kg
m_P	propellant mass	kg
ϵ	structural ratio	-

2 Assumptions

- The system is treated as the flat earth model, rather than cylindrical.
- Two body problem.
- The launch takes place at sea level on the equator, I am going to assume Earth is stationary, rather than spinning, because I'm unsure how to incorporate the spin properly.
- ρ changes with height according to: $\rho = \rho_0 e^{\frac{-h}{h_0}}$
- The pressure effects contributing to thrust are negligble.
- The atmosphere is incompressible.
- Rocket's structural ratio ($\epsilon = 0.01$).
- LOx-Kerosene propelled rocket ($I_{sp} = 353s$).
- Payload mass ($m_L = 1000kg$).
- Initial propellant mass ($m_P = 10000kg$).
- Rocket diameter ($d = 1.65m$).
- Engine burn time ($t_{burn} = 240s$).

3 Formula Derivation

To start, I'm going to use derivations based on the flat earth model that Professor Anderson discussed in lecture. Figure 1 shows a free body diagram of the rocket system, along with the coordinate systems.

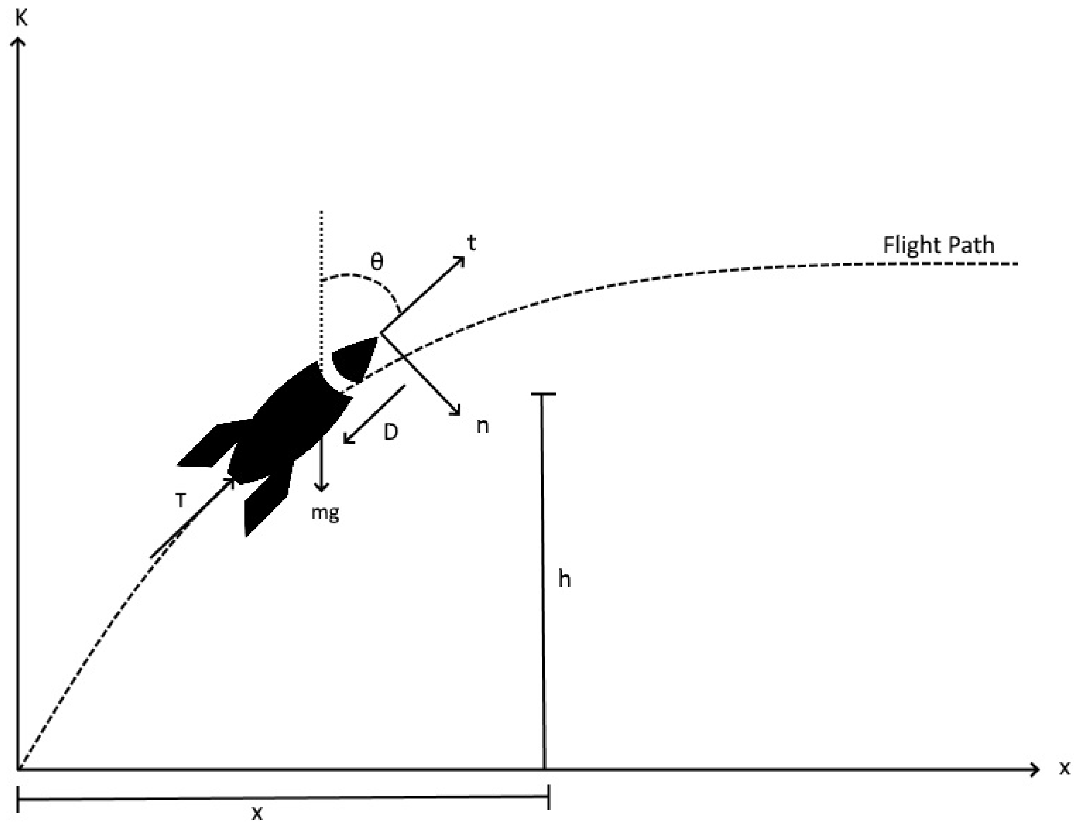

Figure 1: Free Body Diagram of Rocket System

While seemingly complicated at first, our system shows that only three main forces act on the rocket: thrust (T), drag (D), and gravity (mg). An appropriate place to start deriving equations begins with Newton's Second Law, given as the following:

$$\sum \overrightarrow{F}^R = m_R({}^N\overrightarrow{a}^R) \tag{1}$$

The sum of the forces acting on the system is the summation of the thrust, drag, and gravity force terms. Additionally, the acceleration of the rocket can be expressed in the $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{n}}$ directions by the following definition:

$$^N\overrightarrow{a}^R = \dot{v}\hat{\boldsymbol{t}} + v\dot{\theta}\hat{\boldsymbol{n}} \tag{2}$$

By substituting equation (2) and the separate forces into equation (1), the following equation is derived:

$$\overrightarrow{T} + \overrightarrow{D} + \overrightarrow{mg} = m_R(\dot{v}\hat{\boldsymbol{t}} + v\dot{\theta}\hat{\boldsymbol{n}}) \tag{3}$$

To gain further insight into equation (3), it is necessary to express the thrust, drag, and gravity forces in the $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{n}}$ directions, along with making substitutions for the values of the forces. It is important to note that the thrust and drag forces always act in the $\hat{\boldsymbol{t}}$ direction, while gravity has a component in both the $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{n}}$ directions. To start, the thrust, drag, and gravity terms are defined as follows:

$$\overrightarrow{T} = ((P_E - P_A)A + \dot{m}V_E)\hat{\boldsymbol{t}} \tag{4}$$

$$\overrightarrow{D} = (-\frac{1}{2}\rho C_D A_R v^2)\hat{\boldsymbol{t}} \tag{5}$$

$$\overrightarrow{mg} = (\frac{m_R \mu}{(r_E + h)^2})(-cos(\theta)\hat{\boldsymbol{t}} + sin(\theta)\hat{\boldsymbol{n}}) \tag{6}$$

As shown in equation (6), the gravity term has a component in both the $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{n}}$ directions. Figure 2 shows how the gravity term is broken into the two separate components.

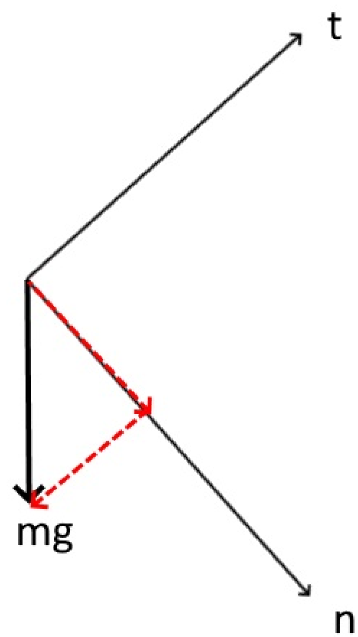

Figure 2: Gravity Force and the Two Components

Combining equations (3), (4), (5), and (6) yields the following equation:

$$((P_E - P_A)A + \dot{m}V_E - \frac{1}{2}\rho C_D A_R v^2)\hat{\boldsymbol{t}} + (\frac{m_R \mu}{(r_E + h)^2})(-cos(\theta)\hat{\boldsymbol{t}} + sin(\theta)\hat{\boldsymbol{n}}) = m_R(\dot{v}\hat{\boldsymbol{t}} + v\dot{\theta}\hat{\boldsymbol{n}}) \tag{7}$$

Equation (7) is a vector equation that can be broken into two separate scalar equations. There will be one equation for each $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{n}}$ directions.

- $\hat{\boldsymbol{t}}$: $\dot{v} = ((P_E - P_A)A + \dot{m}V_E - \frac{1}{2}\rho C_D A_R v^2 - \frac{m_R \mu}{(r_E+h)^2} cos(\theta))\frac{1}{m_R}$
- $\hat{\boldsymbol{n}}$: $\dot{\theta} = \frac{\mu}{v(r_E+h)^2} sin(\theta)$

These two equations are the full set of nonlinear differential equations for the rocket system. Additionally, once the system of differential equations is solved, the rate of change of height ($\dot{h}$) and rate of change of downwards distance ($\dot{x}$) can be found as follows:

$$\dot{h} = vcos(\theta) \tag{8}$$

$$\dot{x} = vsin(\theta) \tag{9}$$

Lastly, several other equations are necessary in order to account for various effects on the rocket, such as density decreasing with height and the velocity of the eject. The last few miscellaneous equations are as follows:

$$V_E = (I_{sp})g_0 \tag{10}$$

$$\rho = \rho_0 e^{\frac{-h}{h0}} \tag{11}$$

$$A_R = \pi(\frac{d}{2})^2 \tag{12}$$

Substituting these values into the full set of nonlinear differential equations will allow us to accurately model the trajectory of the rocket.

4 Results

Using the previously derived equations of motion, I was able to numerically solve the differential equations using Matlab. A copy of the code I used is found in section 5: Matlab Code. To solve the differential equations, I make use of the ode45 function.

4.1 No Atmosphere, No Pitch

My first result explores what happens to the rocket when there is no atmosphere and the rocket doesn't go through a pitchover. Since the Earth is being considered as stationary, one would expect the rocket to rise straight up until the thruster shuts off, then fall back to earth. Figure 3 shows a plot of my results for the case with no atmosphere and no pitch.

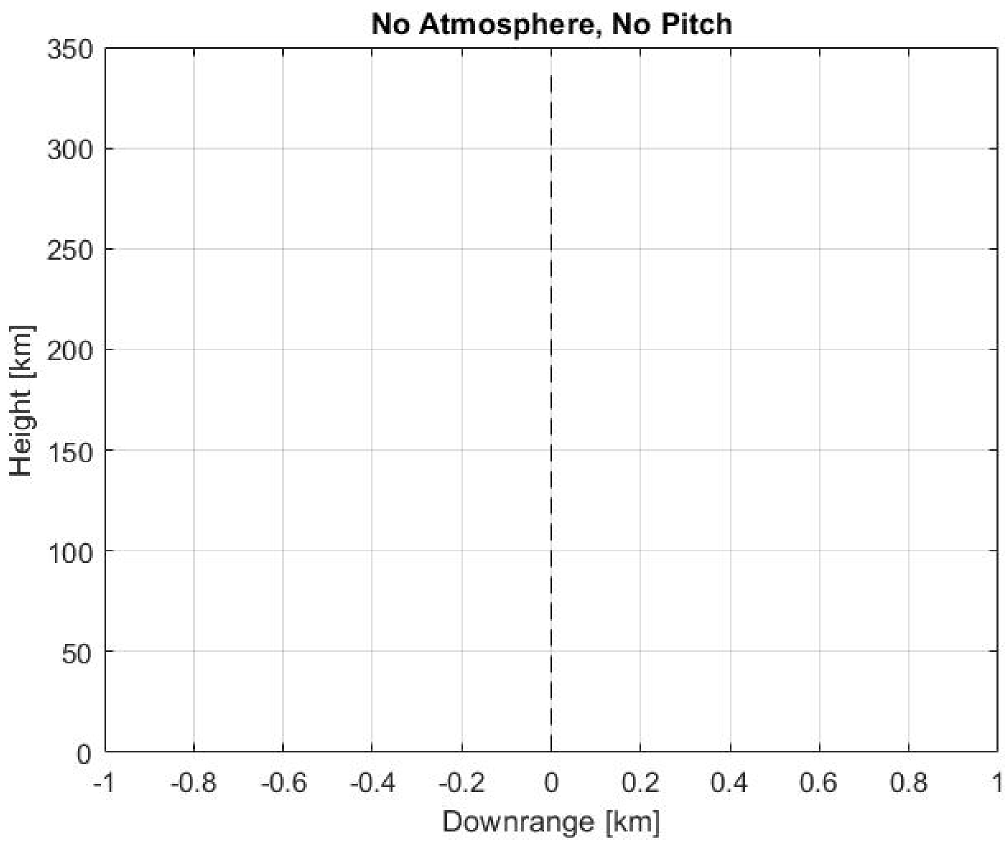

Figure 3: Results for No Atmosphere and No Pitch

The numerical result follows what we expect to happen to the rocket. The rocket increases its altitude while the thruster is firing, but once the propellant runs out, nu more thrust is generated. At this point the rocket no longer increases its altitude, but falls straight back to earth. The rocket reaches a maximum altitude of about 340 kilometers.

4.2 Introducing Pitch

Moving forward, a rocket that moves only straight up and then crashes straight back down is hardly useful. By introducing a pitch angle to the rocket, part of the thrust is directed towards pushing the rocket downrange. Figure 4 shows my results for when I introduce a $0.1°$ pitch to the rocket when it is 500 meters above the launch pad. I accomplish this by calling ode45 twice for each trajectory. First, I call ode45 with a value of 0 for all initial conditions and integrate the differential equations until I reach the height of 500 meters, upon which ode45 stops integrating. Next, I change the initial conditions for velocity, height, downrange distance, and time to be the end values of whatever was calculated during the first ode45 call. I also change the initial condition for the pitchover angle at this point to be the desired "kick" in pitch angle. I then call ode45 again with these new conditions until I reach the end burn time.

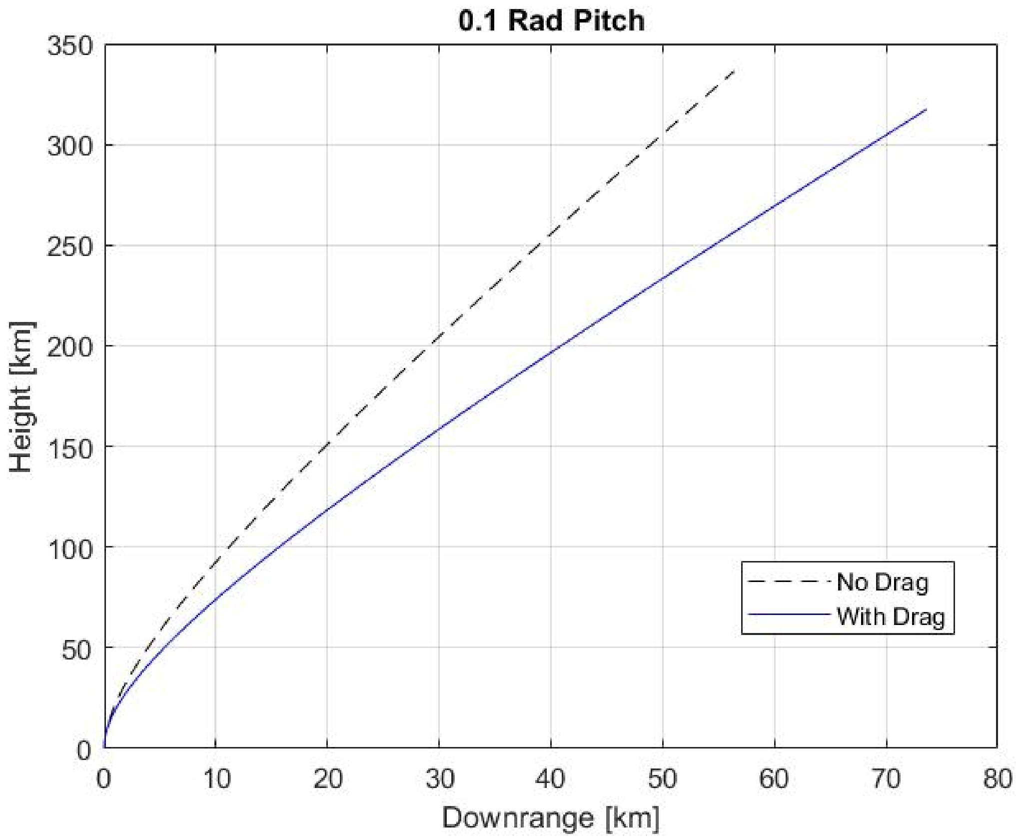

Figure 4: Results for 0.1 Degree Pitch at 500 Meters

4.3 Circular Orbit

Lastly, we seek to have our rocket model put itself into a circular orbit with a height of 150 kilometers. For circular orbits, the necessary velocity for given radius of the orbit is given by the following equation:

$$V_{circ} = \sqrt{\frac{\mu}{r_{orbit}}} \tag{13}$$

In this case, since we desire an orbit altitude of 150 kilometers, the rocket needs a velocity of about 7.81 $\frac{km}{s}$. As such, a height of 150 kilometers and a velocity of 7.81 $\frac{km}{s}$ would mean that the rocket is roughly in a circular orbit. By changing the initial pitchover angle and the mass of the propellant, figure 5 was the results I was able to achieve.

In the end, I was able to achieve a final burn height of 155.185 kilometers, and a final burn velocity of 7.8145 $\frac{km}{s}$. To accomplish this, I slowly started by increasing only the mass of the propellant until I was roughly at the necessary velocity, but with an incorrect height. Then, I started changing only the pitching angle to reduce my overall height. As I increased the pitching angle, my maximum height decreased, but often my velocity increased. As such, I occasionally needed to reduce the mass of the propellant to compensate for the increased velocity from the larger pitching angle. In the end, I found that a propellant mass of 15,350

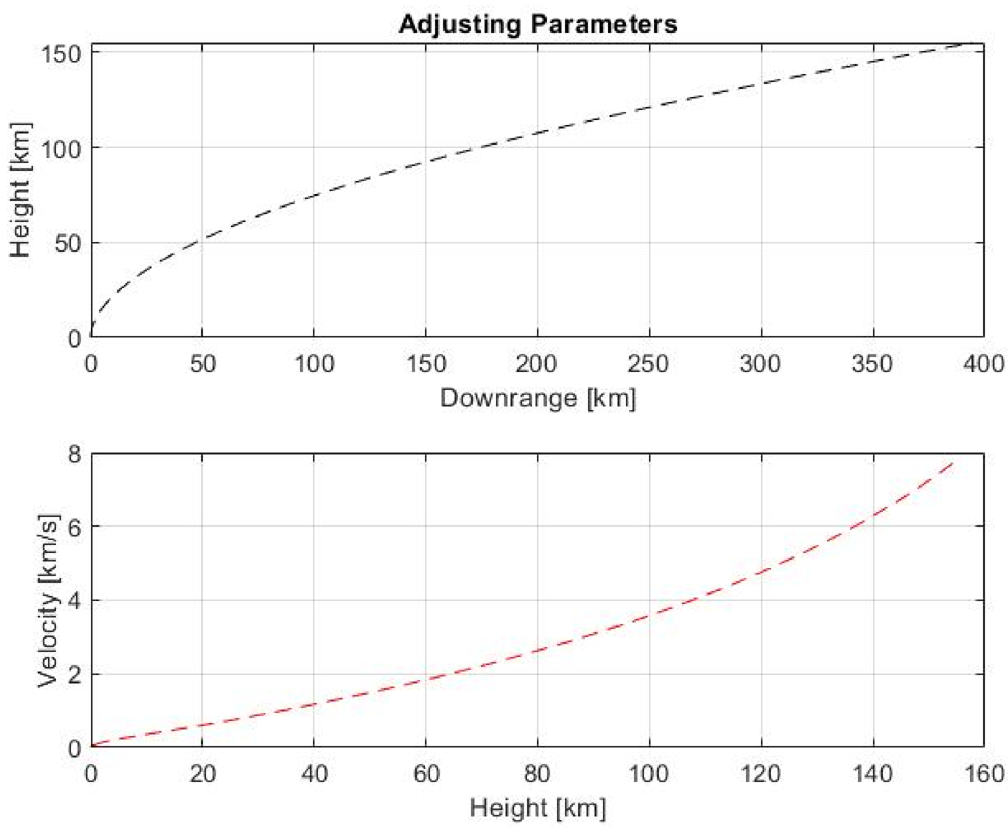

Figure 5: Results from Adjusting Parameters

kilograms and an initial pitching angle of 1.1° achieves the results specified above. In an LMS announcement, Paul described that the downrange distance for this case should be well over 1000 kilometers. My downrange distance is only about 400 kilometers. I opted to not change the parameters further to try and increase my downrange distance, since the rocket is already at the required height and velocity. However, the final pitch angle at the end of the burn is only 77.6°. This poses a problem, since it means that the velocity isn't perfectly tangential to the surface of the Earth. I attempted to keep adjusting the parameters, but was unable to come to a good balance of final height, velocity, and an angle of 90°. Figure 5 was the closest I could get.

There are several things that my code doesn't account for. First, my model assumed a stationary Earth. In reality, the spin of the Earth would have given the rocket a little extra initial velocity. Second, a final circularizing burn would be required once the rocket reaches its final height. I haven't attempted to find what that delta-v would be, but for my results in figure 5, it would likely be a large burn since so much of my velocity isn't tangential to the surface of the Earth. A more accurate model would take both of these considerations into account and likely generate better parameters for the rocket than those that I've come to.

5 Matlab Code

Contents

Rocket Problem

Spaceflight Mechanics - Spring 2019 Johnathan Corbin

```
function RocketProblem
```

Clearing console and variables.

```
clear, clc
```

Declaring constant values given from problem statement.

```
Isp = 353; %LOx - Kerosene propellant, [s]
epsilon = 0.01; %Rocket's structural ratio
mass_L = 1000; %Payload mass, [kg]
mass_P = 10000; %Propellant mass, [kg]
diameter = 1.65; %Rocket diameter, [m]
t_burn = 240; %Total burn time, [s]
r_earth = 6378 * 1000; %Radius of the Earth, [m]
t0 = 0; %Initial time, [s]
pitch = 500; %Height at which pitchover begins, [m]
u = 398600 * 1000^3; %Standard gravitational parameter for Earth, [m^3/s^2]
s_g = 9.81; %Standard gravity, [m/s^2]
rho0 = 1.225; %Sea level air density, [kg/m^3]
hscale = 7500; %Density scale height, [m]
```

Intermediate calculations of various values.

```
A = pi * (diameter / 2)^2; %Frontal area of the rocket, [m^2]
tspan = [t0, t_burn]; %Time range for the integrator, [s]
mass_structure = -epsilon * mass_P / (epsilon - 1); %Structural mass of the rocket, [kg]
m_dot = mass_P / t_burn; %Mass rate of ejecta, [kg/s]
thrust = m_dot * Isp * s_g; %Thrust generated by ejecta, [N]
```

No atmosphere, no pitchover angle.

```
theta0 = 0; %Pitchover angle, [rad]
Cd = 0; %Drag coefficient
v0 = 0; %Initial velocity, [m/s]
x0 = 0; %Initial downrange distance, [m]
h0 = 0; %Initial height, [m]
```

```
f0 = [v0, theta0, x0, h0]; % Initial conditions vector
Opt1 = odeset('Events', @Begin_Pitch);
[time, f] = ode45(@rocketMan, tspan, f0, Opt1);
v = f(:, 1);
theta = f(:, 2);
x = f(:, 3);
h = f(:, 4);

figure(1)
plot(x / 1000, h / 1000, '--k')
hold on

v0 = v(end); %Initial velocity, [m/s]
x0 = x(end); %Initial downrange distance, [m]
h0 = h(end); %Initial height, [m]

tspan = [time(end), t_burn];
f0 = [v0, theta0, x0, h0];
Opt2 = odeset('Events', @Crashed);
[~, f] = ode45(@rocketMan2, tspan, f0, Opt2);
v = f(:, 1);
theta = f(:, 2);
x = f(:, 3);
h = f(:, 4);

plot(x / 1000, h / 1000, '--k')
grid on
title('No Atmosphere, No Pitch')
xlabel('Downrange [km]')
ylabel('Height [km]')
```

No atmosphere, 0.1 rad pitchover angle.

```
theta0 = 0.1 * pi / 180; %Pitchover angle, [rad]
Cd = 0; %Drag coefficient
v0 = 0; %Initial velocity, [m/s]
x0 = 0; %Initial downrange distance, [m]
h0 = 0; %Initial height, [m]
tspan = [t0, t_burn];

f0 = [v0, 0, x0, h0]; % Initial conditions vector
Opt1 = odeset('Events', @Begin_Pitch);
[time, f] = ode45(@rocketMan, tspan, f0, Opt1);
v = f(:, 1);
theta = f(:, 2);
x = f(:, 3);
h = f(:, 4);

figure(2)
plot(x / 1000, h / 1000, '--k')
hold on

v0 = v(end); %Initial velocity, [m/s]
```

```
x0 = x(end); %Initial downrange distance, [m]
h0 = h(end); %Initial height, [m]

tspan = [time(end), t_burn];
f0 = [v0, theta0, x0, h0];
Opt2 = odeset('Events', @Crashed);
[~, f] = ode45(@rocketMan2, tspan, f0, Opt2);
v = f(:, 1);
theta = f(:, 2);
x = f(:, 3);
h = f(:, 4);

p(1) = plot(x / 1000, h / 1000, '--k');
grid on
title('0.1 Rad Pitch')
xlabel('Downrange [km]')
ylabel('Height [km]')
```

Atmosphere, 0.1 rad pitchover angle.

```
theta0 = 0.1 * pi / 180; %Pitchover angle, [rad]
Cd = 0.3; %Drag coefficient
v0 = 0; %Initial velocity, [m/s]
x0 = 0; %Initial downrange distance, [m]
h0 = 0; %Initial height, [m]
tspan = [t0, t_burn];

f0 = [v0, 0, x0, h0]; % Initial conditions vector
Opt1 = odeset('Events', @Begin_Pitch);
[time, f] = ode45(@rocketMan, tspan, f0, Opt1);
v = f(:, 1);
theta = f(:, 2);
x = f(:, 3);
h = f(:, 4);

plot(x / 1000, h / 1000, 'b')
hold on

v0 = v(end); %Initial velocity, [m/s]
x0 = x(end); %Initial downrange distance, [m]
h0 = h(end); %Initial height, [m]

tspan = [time(end), t_burn];
f0 = [v0, theta0, x0, h0];
Opt2 = odeset('Events', @Crashed);
[~, f] = ode45(@rocketMan2, tspan, f0, Opt2);
v = f(:, 1);
theta = f(:, 2);
x = f(:, 3);
h = f(:, 4);

p(2) = plot(x / 1000, h / 1000, 'b');
legend(p([1 2]), 'No Drag','With Drag', 'location', 'best')
```

Atmosphere, 0.1 rad pitchover angle, changing parameters

```
theta0 = 1.1 * pi /180; %Pitchover angle, [rad]
Cd = 0.3; %Drag coefficient
v0 = 0; %Initial velocity, [m/s]
x0 = 0; %Initial downrange distance, [m]
h0 = 0; %Initial height, [m]
tspan = [t0, t_burn];

mass_P = 15350;
m_dot = mass_P / t_burn; %Mass rate of ejecta, [kg/s]
thrust = m_dot * Isp * s_g; %Thrust generated by ejecta, [N]

f0 = [v0, 0, x0, h0]; % Initial conditions vector
Opt1 = odeset('Events', @Begin_Pitch);
[time, f] = ode45(@rocketMan, tspan, f0, Opt1);
v = f(:, 1);
theta = f(:, 2);
x = f(:, 3);
h = f(:, 4);

figure(3)
subplot(2, 2, [1 2])
plot(x / 1000, h / 1000, '--k')
hold on

subplot(2, 2, [3 4])
plot(h / 1000, v / 1000, '--k')
hold on

v0 = v(end); %Initial velocity, [m/s]
x0 = x(end); %Initial downrange distance, [m]
h0 = h(end); %Initial height, [m]

tspan = [time(end), t_burn];
f0 = [v0, theta0, x0, h0];
Opt2 = odeset('Events', @Crashed);
[~, f] = ode45(@rocketMan2, tspan, f0, Opt2);
v = f(:, 1);
theta = f(:, 2);
x = f(:, 3);
h = f(:, 4);

subplot(2, 2, [1 2])
plot(x / 1000, h / 1000, '--k')
grid on
title('Adjusting Parameters')
xlabel('Downrange [km]')
ylabel('Height [km]')

subplot(2, 2, [3 4])
plot(h / 1000, v / 1000, '--r')
grid on
xlabel('Height [km]')
```

```
ylabel('Velocity [km/s]')

fprintf('\n Final burn height =                                          %g', h(end) / 1000)
fprintf('\n Final burn velocity =                                        %g', v(end) / 1000)
fprintf('\n Final burn angle =                                           %g', theta(end) * 180 / pi)
fprintf('\n')
```

```
 Final burn height =                                155.185
 Final burn velocity =                              7.8145
 Final burn angle =                                 77.6481
```

Rates function called by ODE45.

```
function dydt = rocketMan(t, y)
    dydt = zeros(size(y));

    v = y(1);
    theta = y(2);
    x = y(3);
    h = y(4);

    if t < t_burn
       mass = mass_structure + mass_L + mass_P - m_dot * t;
       T = thrust;
    else
        T = 0;
        mass = mass_structure + mass_L;
    end

    g = -(mass * u) / (r_earth + h)^2;
    density = rho0 * exp(-h / hscale);
    drag = -0.5 * density * Cd * A * v^2;

    theta_dot = 0;
    v_dot = (T + drag + g) / mass;
    x_dot = 0;
    h_dot = v;

    dydt(1) = v_dot;
    dydt(2) = theta_dot;
    dydt(3) = x_dot;
    dydt(4) = h_dot;
end

function dydt = rocketMan2(t, y)
    dydt = zeros(size(y));

    v = y(1);
    theta = y(2);
    x = y(3);
    h = y(4);

    if t < t_burn
```

```
        mass = mass_structure + mass_L + mass_P - m_dot * t;
        T = thrust;
    else
         T = 0;
         mass = mass_structure + mass_L;
    end

    g = (mass * u) / (r_earth + h)^2;
    density = rho0 * exp(-h / hscale);
    drag = 0.5 * density * Cd * A * v^2;

    v_dot = (T - drag - g * cos(theta)) / mass;
    theta_dot = (u * sin(theta)) / (v * (r_earth + h)^2);
    x_dot = v*sin(theta);
    h_dot = v*cos(theta);

    dydt(1) = v_dot;
    dydt(2) = theta_dot;
    dydt(3) = x_dot;
    dydt(4) = h_dot;
end

function [value,isterminal,direction] = Begin_Pitch(~,y)
%Event funtion to stop integration when rocket reaches 500[m]
if y(4) < pitch
 value = 1; %Keep going
else %If not
 value = 0; %Then stop
end
isterminal = 1; %Terminate integration when condtion met
direction = 0; %Direction doesn't matter
end

function [value,isterminal,direction] = Crashed(~,y)
%Event funtion to stop integration when rocket reaches 500[m]
if y(4) > 0
 value = 1; %Keep going
else %If not
 value = 0; %Then stop
end
isterminal = 1; %Terminate integration when condtion met
direction = 0; %Direction doesn't matter
end

end
```