ASEN 2012 Project 2

UID: 4b6686457cc3

This paper applies a computer's ability to quickly produce accurate numerical solutions to systems of differential equations to predict and otherwise analyze the flight of a bottle rocket. The simulation involves the application of kinematic physics, thermodynamics, and aerodynamics. It accounts for a variety of involved factors, including drag, the rocket's mass varying with time, the thrust varying with time, and so on. The program was used to determine the values of four initial properties of the rocket required for a flight of a desired distance. Furthermore, the program was used as a mean of determining which of these four parameters held the greatest influence over the rocket's trajectory.

Nomenclature

t	=	time
θ	=	angle between the rocket's nose and the horizontal axis
g_0	=	acceleration due to gravity
γ	=	ratio of specific heats of air, 1.4
R	=	gas constant of air, equivalent to 287 J/(kg*K)
V	=	velocity of rocket
V_0	=	initial velocity of the rocket
V_e	=	the velocity of the gas exiting the rocket during phase two
x	=	horizontal position of the rocket
z	=	height above ground level
ρ_{air}	=	density of air
ρ_w	=	density of water
v_w^i	=	initial volume of water in rocket
v_B	=	total volume of the bottle rocket
m_B	=	mass of bottle rocket (including nosecone and fins)
m_A	=	mass of air in the bottle rocket
m_R	=	mass of rocket and internal water and air
m_R^i	=	initial mass of rocket and internal water and air
C_D	=	drag coefficient of the rocket
c_D	=	discharge coefficient of the rocket
A_B	=	cross-sectional area of the rocket
A_T	=	area of the rocket's throat
p_a	=	ambient atmospheric pressure
p_0	=	initial absolute pressure of air in rocket
p_{air}^i	=	initial gage pressure of air in rocket
p_{end}	=	the internal air pressure at the end of phase one
T_{end}	=	the internal temperature at the end of phase one
v_{air}^i	=	initial volume of air in rocket
m_{air}^i	=	initial mass of air in rocket
T_{air}^i	=	initial temperature of air in rocket

Table of Contents

I. Introduction

Newton's laws of motion provide a framework for modeling the movement of objects through space. In the forms most useful to this project, these laws manifest themselves as first-order differential equations. In order to model additional phenomena such as drag, laws of aerodynamics must be appealed to. To combine these models and use them to produce a simple model of the bottle rocket's trajectory, we made extensive use of MATLAB's ode45, a powerful tool for generating numerical solutions to systems of differential equations. By providing our simulation with the same parameters as in the verification case, MATLAB produced a trajectory nearly identical to the verification trajectory, thus proving that our methodology was correct. Additionally, by varying the initial parameters of the flight, we were able to predict the properties of a rocket that would impact the ground 80 meters from the launch site. Lastly, we performed an analysis of the system in order to determine which of the parameters held the greatest overall sway over the trajectory.

II. Methodology

A bottle rocket is typically composed of an inverted plastic bottle with a cone mounted to the top and fins affixed to the sides. Thrust is provided through the expulsion of pressurized air and water from the bottle's mouth. To model the rocket's flight trajectory, we look to the equations of kinematics and aerodynamics, many of which have been provided in their differential form. To solve this series of interrelated equations, MATLAB's ode45 numerical solver was used. During the flight, there are three physically distinct phases the rocket passes through: during the first, water is expelled and provides thrust. During the second, pressurized air fills this role, and during the third, the rocket behaves as a ballistic projectile. The properties of the rocket are in part governed by a set of equations unique to that phase. However, the fundamental equations of kinematics apply throughout the entirety of the flight. The rocket's horizontal position is governed by the differential equation

$$\frac{dx}{dt} = V\,cos(\theta) \tag{1}$$

while its vertical position is described by the corresponding equation

$$\frac{dz}{dt} = V\,sin(\theta) \tag{2}$$

where the rocket's velocity V is defined as

$$\frac{dv}{dt} = (F - D - m_R g_0 sin(\theta))/m_R \tag{3}$$

where F and D are respectively the the thrust created by the rocket and the drag acting on it. Drag increases by the square of velocity and is described throughout the entire flight by

$$D = \frac{\rho_{air}}{2} V^2 C_D A_B \quad (4)$$

As the thrust can vary dramatically depending on whether its source is water or air being expelled, F must be uniquely described for each phase the rocket passes through. θ is typically defined by the differential equation

$$\frac{d\theta}{dt} = -g_0 cos(\theta)/V \quad (5)$$

To model for the presence of a launching ramp, the simulation holds $d\theta/dt$ equal to zero while the rocket is travelling at less than one meter per second. For this brief duration, this has the effect of holding the rocket at a constant angle, θ_o, while the rocket accelerates.

We now physically define and classify the forces acting on the rocket during each of the three aforementioned phases.

A. Phase 1: Water Expulsion

The first phase of the rocket's flight is defined as beginning when the rocket is launched and ending when the volume of water stored within it has been completely expelled. Equivalently, and of particular relevance to our MATLAB simulation, it can be defined as the time during which the volume of pressurized air in the rocket is smaller than the entire volume of the bottle; initially, this volume of air equals the difference between the volume of the rocket and the initial volume of water and increases as phase one progresses. During this phase, thrust is provided by water being expelled through the bottle's throat at a high speed. The magnitude of this thrust is described by

$$F = 2c_D(p - p_a)A_t \quad (6)$$

where p is the pressure of the air in the rocket alongside the water, defined as

$$p = p_0(v^i_{air}/v)^\gamma \quad (7)$$

where v is the volume of pressurized air in the bottle, described for this phase by the differential equation

$$\frac{dv}{dt} = c_D A_t \sqrt{(2/\rho_w)[p_0(v_o/v)^\gamma - p_a]} \quad (8)$$

Additionally, in accordance with equation (3), we must keep track of the rocket's mass m_R, defined here as

$$\frac{dm_R}{dt} = -c_D A_t \sqrt{2\rho_w(p - pa)} \quad (9)$$

We note that we must additionally keep track of the amount of air in the rocket over the course of the flight. As water is the only fuel being expelled from the rocket during phase one, $dm_A/dt = 0$. As previously mentioned, when the water has been completely expelled from the rocket, the volume of pressurized air will have risen to the same volume as the volume of the bottle. In regard to our MATLAB simulation, the equations of phase one are governed by and executed only when the air volume is less than the volume of the bottle. At the moment that this statement becomes untrue, we mark the end of phase one.

B. Phase 2: Pressurized Air Expulsion

When phase two begins, the air contained in the rocket begins at a substantially higher pressure than ambient atmospheric pressure. It ends when the internal pressure falls to the atmospheric pressure. Thus, thrust during this phase is solely provided by the expulsion of pressurized air, ultimately computed by

$$F = -\frac{dm_A}{dt} V_e + (p_e - p_a) A_t \tag{10}$$

However, many of these quantities, including dm_A/dt, V_e, and p_e are as of yet undefined for this phase. The change in the mass of the air with respect to time, equivalent to the change in the rocket's total mass is described by

$$\frac{dm_A}{dt} = \frac{dm_R}{dt} = -c_D \rho_e A_t V_e \tag{11}$$

As a brief aside, we briefly define two values: p_{end} and T_{end}, respectively the pressure and temperature inside the bottle at the end of phase one. They are defined as

$$p_{end} = p_0 (v^i_{air}/v_B)^{\gamma};\ T_{end} = T^i_{air} (v^i_{air}/v_B)^{\gamma-1} \tag{12}$$

During phase two, there are two distinct possibilities: the flow is choked, which occurs when the critical pressure is greater than the ambient atmospheric pressure, or it is not, in which case the ambient pressure is greater than or equal to the critical pressure. The critical pressure, a function of pressure, is defined as

$$p_{cr} = p\left(\frac{2}{\gamma+1}\right)^{\gamma/(\gamma-1)} \tag{13}$$

We additionally note that during choked or unchoked flow, the temperature and air density in the main body of the bottle are defined respectively as

$$T = \frac{p}{\rho R};\ \rho = m_a/v \tag{14}$$

These two quantities will shortly be required in the calculation of the temperature and exhaust speed of the gas in the bottle's throat.

I. Choked Flow: Critical pressure is greater than ambient pressure

If the flow of compressed air is choked, then it is known that p_e, the pressure at the bottle's exit, is equal to the critical pressure defined above. Additionally, the air temperature at the exit is described by

$$T_e = \left(\frac{2}{\gamma+1}\right) T \tag{15}$$

and the velocity of the exhaust is mach one, defined here as

$$V_e = \sqrt{\gamma R T_e} \tag{16}$$

and the density of the air in the bottle's throat is defined as

$$\rho_e = p_{cr}/(RT_e) = p_e/(RT_e) \tag{17}$$

II. Unchoked flow: Ambient pressure is greater than or equal to critical pressure

In the case that the ambient pressure is as great or greater than the critical pressure defined in equation (b), the quantities defined in the case of choked flow are computed differently. Now, we define the pressure at the bottle's exist to be equivalent to the ambient pressure. That is,

$$p_e = p_a \tag{18}$$

Furthermore, the exit temperature is now defined as

$$T_e = T(1 + \frac{\gamma-1}{2}M^2) \tag{19}$$

where the mach number M_e of the exiting exhaust is

$$M_e = \sqrt{\frac{2}{\gamma-1} * [(\frac{p}{p_a})^{(\gamma-1)/\gamma} - 1]} \tag{20}$$

Then, the exhaust velocity during unchoked flow is

$$V_e = M_e * \sqrt{\gamma R T_e} \tag{21}$$

and the exhaust density is

$$\rho_e = p_a/(RT_e) = p_e/(RT_e) \tag{22}$$

From the above, depending on whether the exhaust is choked or unchoked, values may now be obtained for equations (o) and (m). We additionally note that during phase two, $dv/dt = 0$ because the volume of the air has already reached the volume of the entire bottle, which clearly cannot expand further. After the air pressure in the rocket has fallen to or below ambient atmospheric pressure, we mark the end of phase two and switch to phase three.

C. Phase 3: Ballistic Trajectory

During the ballistic phase of the simulation, the physics become greatly simplified. There is no longer any thrust being produced or any change in the rocket's mass. The only forces now acting on the rocket are gravity and drag. Consequently, the rocket's trajectory is now governed solely by the solutions to equations (1), (2), (3), (4), and (5) where $F = 0$ and $m_R = m_B$. The minimal equations governing the physics during phase three are simply executed whenever the conditions for phase one and two are not met. That is, phase three is executed only when the volume of air in the rocket has reached the bottle's volume and the internal air pressure has fallen to ambient pressure.

The solution set provided by ode45 includes the rocket's horizontal and vertical positions during its flight. By plotting these against each other, MATLAB yields a purely spatial plot of the rocket's trajectory.

III. Results

In order to test the veracity of our simulation, a test case was provided. The verification indicates that the bottle rocket launched with those parameters travelled slightly less than 54 meters meters and attained a maximum above-ground altitude of around 16.5 meters. Providing our simulation with the same parameters produces the following trajectory.

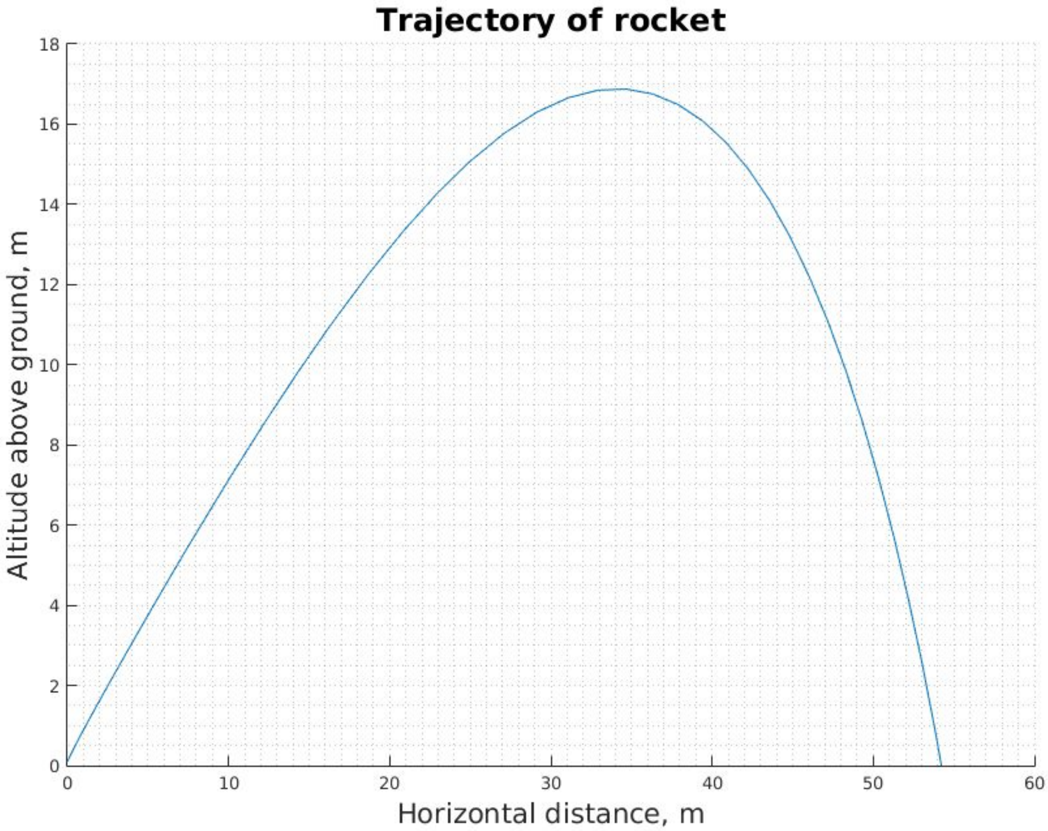

Figure 1. *Trajectory matches the trajectory provided with the verification case.*

IV. Discussion

As seen in figure 1**,** our simulation predicts that the rocket will attain a maximum altitude of just short of 16.6 meters and impacts the ground 53.8 meters from the launch site. As these results appear to perfectly match the trajectory provided with the verification case, we can conclude that our methodology was correct and that our simulation has been programmed successfully.

A task set before us was to explore the parameter space and find a set of initial conditions that would produce a flight with an impact distance of 80 ±1 m from the launch site. Four parameters were made available for variation: the initial air pressure in the rocket, the initial mass of water in the rocket (or equivalently, the water's initial volume), the rocket's coefficient of drag, and the rocket's launch angle. From equation (4), we see that the drag force acting on the rocket is directly proportional to the coefficient of drag. Thus, it is intuitive that by lowering C_D to its minimum possible value, 0.3, we can extend the rocket's flight. To demonstrate this, the below plot depicts the trajectory of the rocket taken from the verification case, but with one difference: its drag coefficient has been lowered from 0.5 to 0.3.

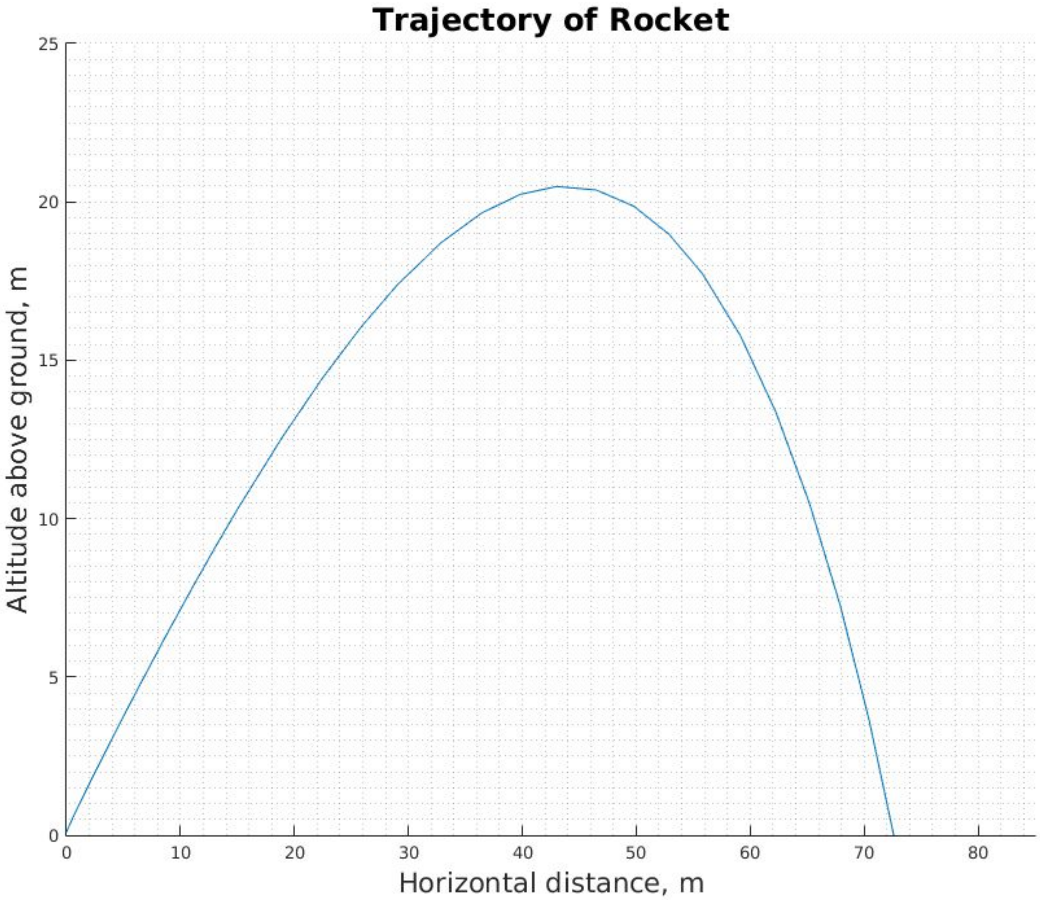

Figure 2.

Though closer to our target of 80 meters, we must change one additional parameter to reach the desired distance. From intuition and from equation (6), we can conclude that pressurizing the air in the rocket to a greater degree will generate more thrust throughout phases one and two. By raising the internal pressure 20,000 pascals from the initial pressure listed in the verification case but otherwise reusing the parameters that let to the trajectory depicted in figure 2, MATLAB produces the following trajectory.

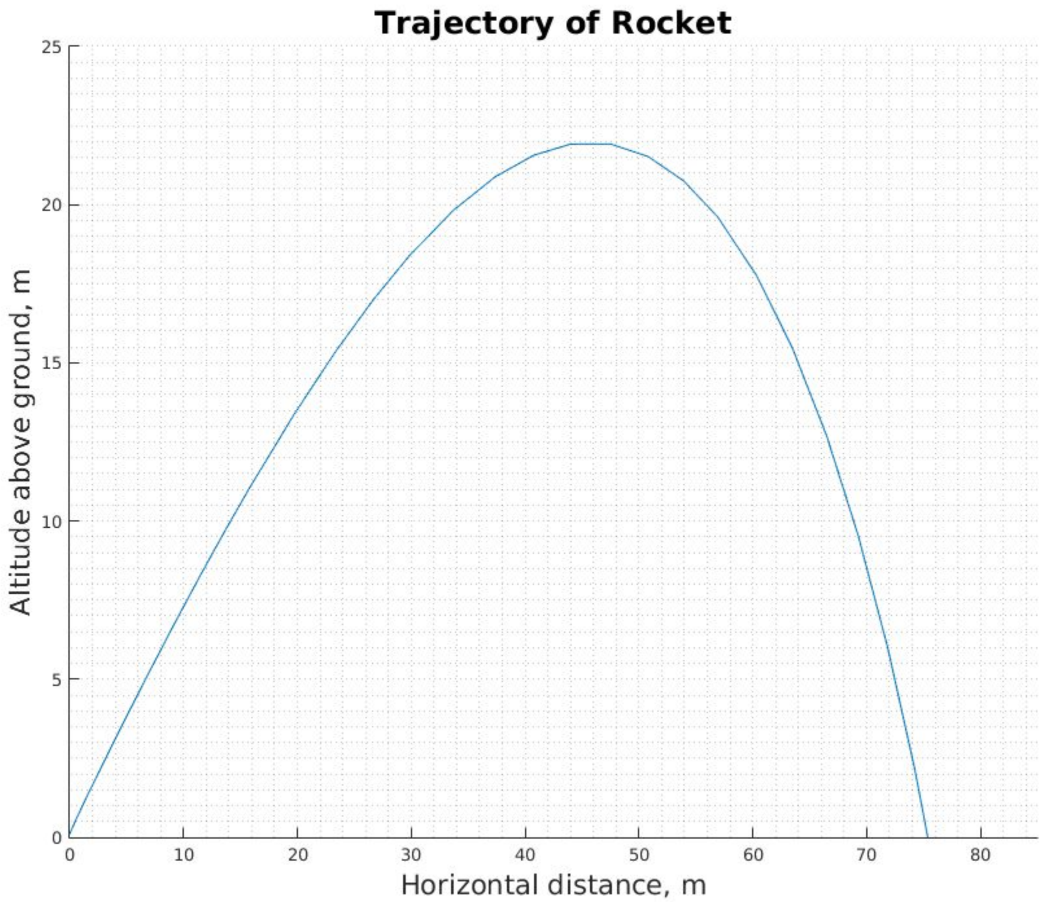

Figure 3.

Though marginally closer than in figure 3, we must raise the internal air pressure slightly higher to reach 80 meters. Through trial and error, we determined that a flight of 80.00 meters could be achieved with an initial total air pressure of 485,000 pascals, an initial water volume of 0.001 m^3, a drag coefficient of 0.3, and a launch angle of 45°. The trajectory of the rocket launched with those parameters is plotted below.

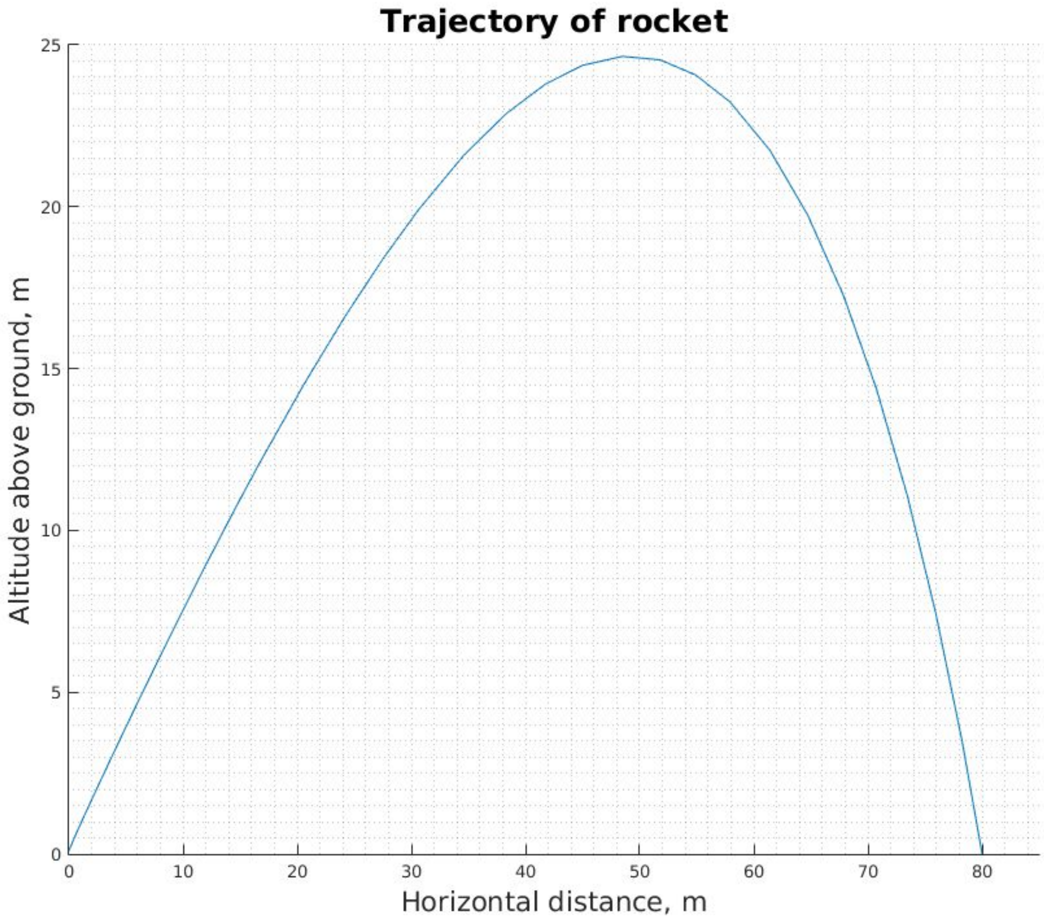

Figure 4. *Parameters capable of propelling the rocket a horizontal distance of 80 meters were determined, leading to this figure.*

From outside research, we discovered that the type of plastic bottle most similar to those that will be used in our rockets have a typical bursting pressure of between 1,000,000 and 1,400,000 pascals. As our chosen initial pressure of 485,000 pascals is substantially lower than these limits, it is quite likely that a rocket pressurized to this degree would be safe. Assuming a worst case scenario, the bottle will explode at the 1,000,000 pascals, the lower bursting pressure. This would provide us with a relatively lofty factor of safety of 2.06. Assuming a more moderate bursting pressure of 1,200,000, we have an even higher factor of safety of 2.47.

We now perform a brief analysis of the influence of the four variable parameters over the bottle rocket's trajectory. The parameters of the verification case will be used as the default set of initial conditions. Each parameter will be varied by 10% of their respective possible range and the distance the rocket travels before impact will be recorded. As an example, the coefficient of drag can be varied from 0.3 to 0.5. 10% of the difference, 0.2, is 0.02. Thus, for the verification case wherein the default coefficient of drag is 0.5, the trajectories based on rockets with drag coefficients of 0.48 and 0.52 will be simulated and recorded. By default, the rocket based on the verification case travels 53.8 meters before impact. In the table below, we record variations on this default impact distance as each of the four parameters is varied by 10% while the others are held to their default value.

	Total initial pressure (pa)	Initial water volume (m^3)	Coefficient of drag	Launch angle (°)
+10%	60.5	45.2	52.5	50.3
-10%	42.9	N/A	55.0	51.7

Table 1. *A trajectory could not be calculated when decreasing the initial water volume by 10% due to programming issues. Nonetheless, we see clearly that the the initial pressure is slightly more influential in the rocket's trajectory than is water volume.*

For the sake of clarity, we note that the maximum possible pressure is considered to be the minimum bursting pressure, 1,000,000 pascals, and the minimum pressure is the ambient atmospheric pressure. The maximum water volume is the entire volume of the bottle and the minimum is none of it. The maximum launch angle is 90° and the minimum is 0°. Clearly, the greatest deviation from the default impact distance of 53.8 meters is achieved when varying the total initial pressure by 10%, and thus we conclude that the initial internal air pressure holds the greatest influence over the rocket's trajectory.

V. Conclusion

As a result of this project, we have developed a MATLAB program for modeling the flights of bottle rockets to some reasonable degree of accuracy. Additionally, we have learned a great deal about the MATLAB solver ode45's powerful ability to numerically provide solutions to large systems of differential equations in ways that would be impractically difficult to carry out manually. Additionally, in a relatively rational and logically sound way, we went about determining which of the parameters that went into the trajectory's calculation influenced it most strongly, concluding that the rocket's internal air pressure held greatest sway. Though it has not been designed to account for some of reality's more messy aspects, such as wind, this simulation will likely be very helpful when the time arrives to physically construct bottle rockets.

VI. References

The MathWorks, Inc., "ode45: Solve nonstiff differential equations; medium order method," *MATLAB Documentation* [online database], URL: http://www.mathworks.com/help/matlab/ref/ode45.html?refresh=true [cited 6 December 2015]

Aerospace Department, Univesity of Colorado Boulder, "Project 2 Bottle Rocket Design," *ASEN 2012 Project 2* [online database], URL: https://learn.colorado.edu/d2l/le/content/111675/viewContent/2188968/View [cited 6 December 2015]

VII. Appendix

Code

main.m

```
% Purpose: Establish constants and initial conditions of the system,
% execute ode45, extract horizont and vertical coordinates from solution
% set, and plot.

% Inputs: None.

% Ouputs: The X and Y coordinates of the bottle rocket as a function of
% time, a plot of the rocket's trajectory

% Assumptions: Flow during phase 1 is incompressible, flow during phase 2
% is isentropic and compressible, no wind acts on the rocket, force due to
```

```
% gravity is constant

% ID Number: 4b6686457cc3

% Date created: November 25th, 2015

% Date modified: December 6th, 2015

global g0 gamma R pa rhoair rhow vB Cd cd Ab At p0 vAi mAi TAi mB;
   % g0: Acceleration due to gravity
   % gamma: Ratio of specific heats of air
   % R: The gas constant for air
   % pa: Ambient pressure
   % rhoair: Density of outside air
   % rhow: Density of water
   % vB: The volume of the bottle
   % Cd: Drag coefficient
   % cd: Discharge coefficient
   % Ab: Cross-sectional area of bottle
   % At: Area of throat
   % p0: Initial air pressure in bottle
   % vAi: Initial volume of air in bottle
   % mAi: The initial mass of air in the bottle
   % TAi: The initial temperature of air in the bottle
   % mB: The mass of the bottle, cone, and vanes (without air and water)

%%{
% Parameters for test case
p0=427682; % +/- 91705 pascals [Initial total pressure]
vWi=0.001; % +/- 0.0002 m^3 [Initial volume of water]
Cd=0.5; % +/- 0.02 [Coefficient of drag]
theta0=45*(pi/180); % +/- 9 degrees [Launch angle]
%%}

%{ % Uncomment this block and comment out the above parameters to test the 80 meter flight
% Parameters that will lead to an 80 meter flight
p0=485000; % The initial gage pressure inside the rocket, pa
vWi=0.001; % The initial volume of water in the rocket, m^3
Cd=0.3; % The rocket's coefficient of drag
theta0=45*(pi/180); % The rocket's launch-angle
%}

pa=82943.93; % Ambient pressure, pa
g0=9.81; % Acceleration due to gravity
R=287; % Gas constant for air
cd=0.8; % Discharge coefficient of rocket
rhoair=0.961; % Density of outside air, kg/m^3
vB=0.002; % Volume of bottle, m^3
```

```
gamma=1.4; % Ratio of specific heats for air
rhow=1000; % density of water, kg/m^3
Dt=0.021; % Diameter of throat, m
Db=0.105; % Diameter of bottle, m
mB=0.07; % Mass of empty bottle, kg
Ab=pi*(Db/2)^2; % Cross-sectional area of bottle, m^2
At=pi*(Dt/2)^2; % Throat area of bottle, m^2
TAi=300; % Initial temperature of air in bottle, K
vAi=vB-vWi; % Initial volume of air in bottle, m^3
mAi=(p0*vAi)/(R*TAi); % The initial mass of air in the rocket, kg
mri=mB+rhow*(vB-vAi)+vAi*(p0/(R*TAi)); % The total initial mass of the rocket, kg
V0=0; % Rocket's initial velocity, m/s
x0=0; % The rocket's initial horizontal position, m
z0=0.1; % The rocket's initial vertical position, m

%% Evaluate the system
[t sol] = ode45(@diffeqs,[0 10], [x0 z0 V0 theta0 mri mAi vAi]); % Run ode45
x=sol(:,1); % Extract horizontal coordinates from the solutions
z=sol(:,2); % Extract vertical coordinates from the solutions
hold on;
plot(x,z); % Plot the trajectory
title('Trajectory of rocket','FontSize',18);
xlabel('Horizontal distance, m','FontSize',16);
ylabel('Altitude above ground, m','FontSize',16);
axis([0 85 0 25]);
grid minor;
```

diffeqs.m

```
% Purpose: Solve the system of differential equations and ultimately
% provide the

% Inputs: a time span over which to simulate, a set of 7 initial conditions.
% Initial conditions are to be given in a vector of the following order: X
% position, Z position, velocity, angle with respect to horizontal, total
% mass of rocket, mass of air internal to rocket, volume of air internal to
% rocket

% Ouputs: The X and Y coordinates of the bottle rocket as a function of
% time, a plot of the rocket's trajectory

% Assumptions: Flow during phase 1 is incompressible, flow during phase 2
% is isentropic and compressible, no wind acts on the rocket, force due to
% gravity is constant

% ID Number: 4b6686457cc3

% Date created: November 25th, 2015
```

```
% Date modified: December 6th, 2015

function [sol] = diffeqs(t,iv)
%% Initial conditions and global variables
  x=iv(1); % Extract horizontal position
  z=iv(2); % Extract vertical position
  V=iv(3); % Extract velocity of rocket
  theta=iv(4); % Extract angle between rocket's nose and horizontal
  mr=iv(5); % Extract total mass of rocket
  mA=iv(6); % Extract total mass of air
  v=iv(7); % Extract total volume of air

  global g0 gamma R pa rhoair rhow vB Cd cd Ab At p0 vAi mAi TAi mB;
  %% Additional factors (drag, etc)
  q=0.5*rhoair*V^2; % Dynamic pressure, always applies
  D=q*Cd*Ab; % Drag force on bottle, always applies
  if ~v<vB % If phases two or three... calculate pressure now for comparison in ifelse
        p_end=p0*(vAi/vB)^gamma;
        p=p_end*(mA/mAi)^gamma; % Calculate pressure
  end

  %% Phase 1: Water is providing thrust
  if v<vB % If air volume is still less than bottle volume...
        p=p0*(vAi/v)^gamma; % The total gas pressure inside the bottle
        F=2*cd*(p-pa)*At; % Calculate thrust
        dmdt=-cd*At*sqrt(2*rhow*(p-pa)); % Calculate change in rocket mass; water is exiting rocket
    dvdt=cd*At*sqrt((2/rhow)*(p0*((vAi/v)^gamma)-pa)); % Change in volume of air
        dAdt=0; % Change in mass of air; water is the only thing currently leaving rocket

  %% Phase 2: Pressurized air is providing thrust
  elseif p>pa % If internal air pressure is still higher than atmospheric pressure...
        T_end=TAi*(vAi/vB)^(gamma-1); % Temperature at end of phase one
        pcr=p*(2/(gamma+1))^(gamma/(gamma-1)); % Critical pressure
        rho=mA/vB; % Calculate the air density in the main bottle
        T=p/(rho*R); % Calculate the temperature of the air in the main bottle
        if pcr>pa % Choked flow
        pe=pcr; % Set exit pressure as critical pressure
        Te=(2/(gamma+1))*T; % Calculate exit temperature
        Ve=sqrt(gamma*R*Te); % Calculate exhaust velocity
        rhoe=pe/(R*Te); % Calculate air density at exit
        else
        pe=pa; % Set exit pressure as ambient pressure
       Me=sqrt((2/(gamma-1))*(((p/pa)^((gamma-1)/gamma))-1)); % Calculate mach number of exhaust
        Te=T*(1+((gamma-1)/2)*Me^2); % Calculate exit temperature
        Ve=Me*sqrt(gamma*R*Te); % Calculate exit velocity
        rhoe=pe/(R*Te); % Calculate air density at exit
        end
        dAdt=-cd*rhoe*At*Ve; % Change of air mass in bottle
```

```
        dmdt=dAdt; % Air is the only thing leaving rocket now, so dmdt=dAdt
        dvdt=0; % As the air's volume now equals the bottle's volume, dvdt=0
        F=-dAdt*Ve+(pe-pa)*At; % Calculate the thrust generated by the outflowing air

    %% Phase 3: Rocket follows ballistic trajectory
    else
        F=0; % No thrust generated
        dmdt=0; % No change in mass
        dvdt=0; % No change in air volume
        dAdt=0; % No change in air mass
    end

    %% Conditions that apply through all three phases
    if V<1 % Hold theta constant until V>1 m/s to account for presence of ramp
        dthetadt=0;
    else
        dthetadt=-(g0*cos(theta))/V; % Else: this at all times
    end
    dzdt=V*sin(theta); % Calculate the change in vertical position
    dxdt=V*cos(theta); % Calculate the change in horizontal position
    dVdt=(F-D-(mr*g0*sin(theta)))/mr; % Calculate the change in velocity

    %% Store solutions and return to main.m
    sol=zeros(7,1); % Allocate empty matrix for solutions
    sol(1)=dxdt; % Solve horizontal position
    sol(2)=dzdt; % Solve vertical position
    sol(3)=dVdt; % Solve velocity
    sol(4)=dthetadt; % Solve angle relative to horizontal
    sol(5)=dmdt; % Solve mass of rocket
    sol(6)=dAdt; % Solve mass of air in rocket
    sol(7)=dvdt; % Solve volume of air in rocket
end
```

Algorithm

Purpose: Use MATLAB to calculate and plot the trajectory of a bottle rocket.

Givens:

- The equations governing the mechanics and aerodynamics of a rocket's flight
- The initial flight parameters for a verification case

Required: The horizontal and vertical position of the bottle rocket as a function of time.

Assumptions:

- Water being expelled during phase one is incompressible.
- Gas being expelled during phase two is compressible and isentropic.
- There is no wind acting on the rocket.
- The force of gravity is constant.

Diagram:

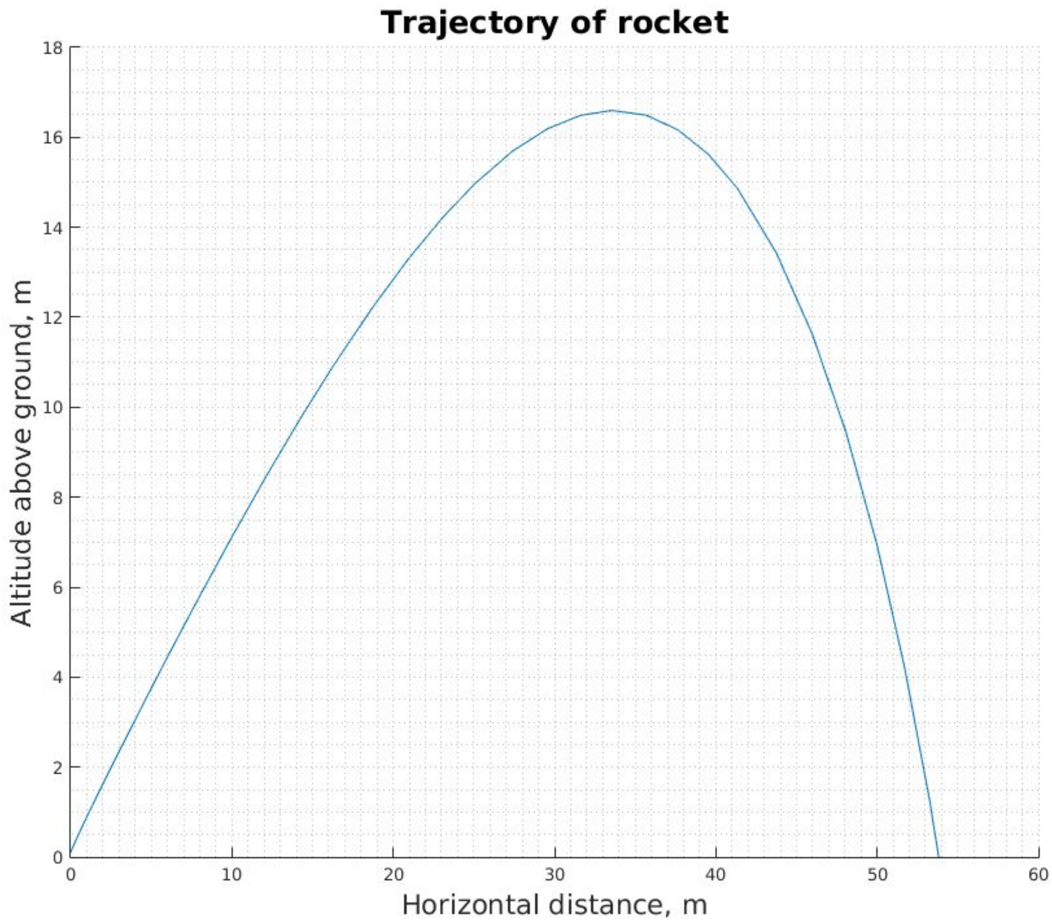

Figure 5. *Demonstration of the simulation's functionality based on parameters of verification case.*

Fundamental principles:

- Newton's laws of motion
- Aerodynamic principles of incompressible flow
- Aerodynamic principles of isentropic flow

Alternate approaches: Newton's laws could be constructed in non-differential forms, thereby allowing us to find solutions without the aid of MATLAB. However, keeping track of the rocket's mass, the air's volume, and so on, would still be necessary and would likely be very difficult to do properly. Thus, modeling the system of differential equations numerically with MATLAB would appear to be our best course of action.

Step-by-step solution:4

- Set constants and initial conditions
- Call ode45
    - Phase one: if the volume of internal air is smaller than the volume of the bottle
        - Apply universal kinematics equations (1) through (5)
        - Apply equations specific to phase one: (6) through (9)
    - Phase two: if the air pressure in the bottle is less than the atmospheric pressure

        - Apply universal kinematics equations (1) through (5)
        - Determine whether or not flow is choked
        - If flow is choked:
            - Apply equations (15) through (17)
        - If flow is unchoked:
            - Apply equations (18) through (22)
        - Apply equations specific to phase two: (10-14)
    - Phase three: any time the conditions for phases one and two are not met
        - Apply universal kinematics equations (1) through (5)
        - Set thrust and changes in volume and mass equal to zero
- Extract solutions and plot horizontal trajectory against vertical trajectory

Simplifying the simulation: Giving the simulation an empty bottle with a positive initial velocity and an internal pressure equal to the ambient atmospheric pressure produces a reasonable trajectory. Thus, the kinematics portion of our simulation is confirmed to function correctly.

Reality check: As the trajectory built from the initial conditions of the verification case match the verification trajectory, we can confirm that our results make sense. Additionally, we note that as the rocket in our simulation horizontally progresses away from the launch site, it slows done. This matches what is should realistically happen: without drag, the rocket would move at a constant horizontal speed. However, the inclusion of drag in our simulation dampens the rocket's horizontal velocity as the flight progresses.