Grand Canyon Problem

Johnathan Corbin
Spaceflight Mechanics - Spring 2019

February 28, 2019

1 Formula Derivation

For the Grand Canyon problem, a rock is dropped with zero initial velocity off a cliff at the Grand Canyon. The rock is able to fall, unobstructed, for 1.5 kilometers. The goal is to determine where the rock will land assuming that it has a radius $r_Q = 0.05meters$ and mass $m_Q = 1.0kg$.

We begin by formulating the equations of motion for the rock. However, I would first like to assign various values and terminology to the system. Throughout the derivations, point O will be the center of the Earth, P is the edge of the cliff at the Grand Canyon, and Q is the falling rock. Figure 1 shows a side view of the cliff with the attached coordinate system, while Figure 2 shows the coordinate system with respect to the Earth. In the figure, Z is the distance the rock has fallen and the k direction points away from the center of the earth.

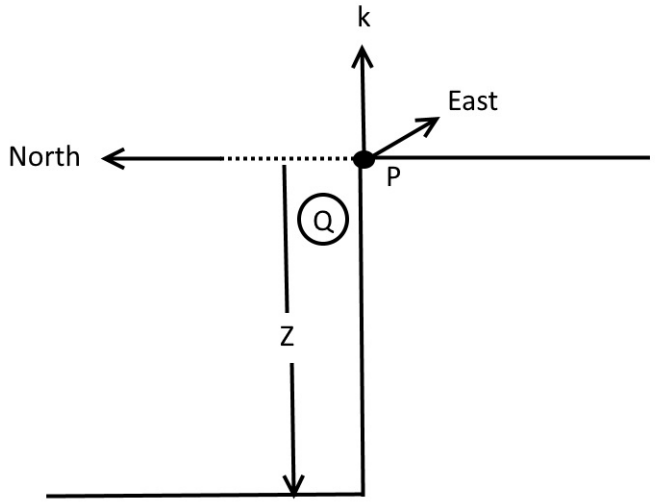

Figure 1: Coordinate View of Cliff

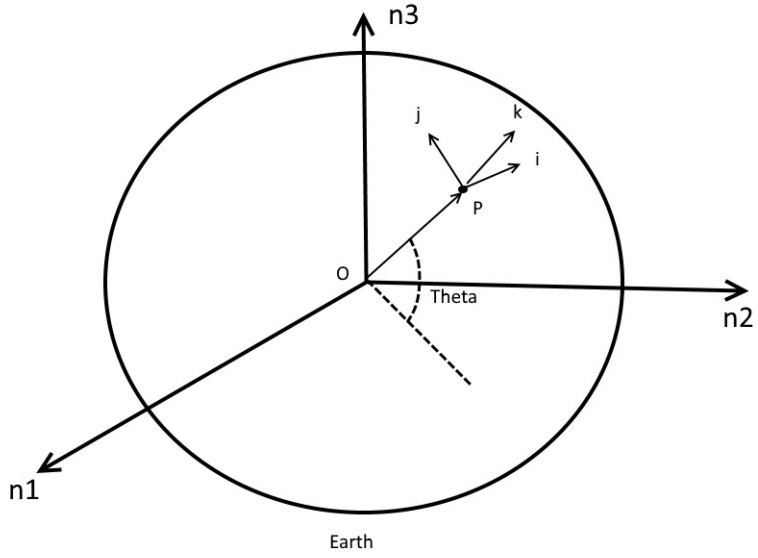

Figure 2: Coordinate View from Earth

Analysis of this problem begins with the free body diagram of the rock, Q, shown in Figure 3. The two forces acting on the body are $\vec{F_g}$ and $\vec{F_D}$ and are due to gravity and drag, respectively. Newton's second law provides a good starting point and is given by:

$$\sum \overrightarrow{F}^Q = m_Q({}^N\overrightarrow{a}^Q) \tag{1}$$

Using the free body diagram, it can be shown that for this system, Newton's second law becomes the following:

$$\sum \overrightarrow{F}^Q = m_Q({}^N\overrightarrow{a}^Q) = \overrightarrow{F_g} + \overrightarrow{F_D} \tag{2}$$

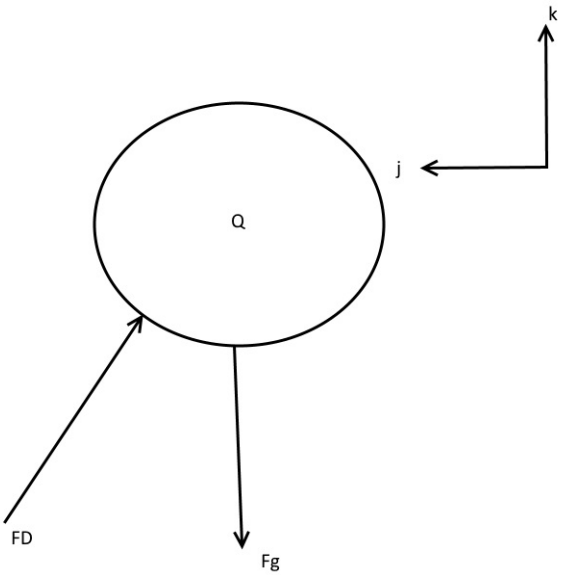

Figure 3: Free Body Diagram of the Rock

Basic dynamics tells us that the five-term acceleration equation for a system is as follows:

$$^{N}\overrightarrow{a}^{Q} = ^{N}\overrightarrow{a}^{P} + ^{E}\overrightarrow{a}^{\frac{Q}{P}} + ^{N}\overrightarrow{\alpha}^{E} \times \overrightarrow{r}^{PQ} + ^{N}\overrightarrow{\omega}^{E} \times (^{N}\overrightarrow{\omega}^{E} \times \overrightarrow{r}^{PQ}) + 2^{N}\overrightarrow{\omega}^{E} \times ^{E}\overrightarrow{V}^{\frac{Q}{P}} \tag{3}$$

where E denotes the reference frame at point P and N denotes the non-rotating inertial frame centered at point O. At this point, it is important to note that $^{N}\overrightarrow{\alpha}^{E} = 0$ for our case. Additionally, $^{N}\overrightarrow{\omega}^{E} = \Omega\hat{n_3}$ where $\Omega = 7.292 \times 10^{-5}$ $[\frac{rad}{s}]$ and is the angular velocity of the Earth.

Next, it is important for us to define values for the two external forces, F_g and F_D. Traditionally, the drag force on a body is given by $F_D = \frac{1}{2}\rho C_D V^2 A$ where ρ is the density of air, V is the magnitude of the velocity of the body, C_D is the coefficient of drag, and A is the projected frontal area. This expression for F_D holds in our case, but we will modify it slightly to the following equation to account for the direction of the of the velocity term so that the drag always opposes velocity:

$$\overrightarrow{F_D} = -^{E}\overrightarrow{V}^{\frac{Q}{P}} V(\frac{1}{2}\rho C_D A) \tag{4}$$

This equation for drag will produce a drag force that has the correct magnitude and direction, even if the velocity of the rock changes direction. However, the velocity must be known. Incidentally, we don't know the velocity as it contains variables we are solving for in the problem. As such, the velocity can be defined by: $^{E}\overrightarrow{V}^{\frac{Q}{P}} = \dot{x}\hat{i} + \dot{y}\hat{j} + \dot{z}\hat{k}$. Lastly, the force due to gravity, F_g is defined as follows:

$$\overrightarrow{F_g} = \frac{-\mu\overrightarrow{r}^{OQ}}{(r^{OQ})^3} \tag{5}$$

where u is the standard gravitational parameter for the celestial body. In our case, $u_{earth} = 398600$ $[\frac{km^3}{s^2}]$. The formula requires the vector $\overrightarrow{r}^{OQ}$ to be known. $\overrightarrow{r}^{OQ}$ can be defined as the sum of the vectors from O to P and from P to Q ($\overrightarrow{r}^{OQ} = \overrightarrow{r}^{OP} + \overrightarrow{r}^{PQ}$). It is known that $\overrightarrow{r}^{OP}$ acts in the $\hat{k}$ direction and is the sum of the radius of Earth and the altitude of the Grand Canyon. As such, assuming the altitude of the Grand Canyon is roughly 2 km, $\overrightarrow{r}^{OP} = r_{earth} + altitude = 6380\hat{k}$ $[km]$. $\overrightarrow{r}^{PQ}$ is unknown and contains the variables x, y, and z that we are to solve for in this system. As such, $\overrightarrow{r}^{PQ} = x\hat{i} + y\hat{j} + z\hat{k}$. Finally, the vector from O to Q can be defined as follows:

$$\overrightarrow{r}^{OQ} = x\hat{i} + y\hat{j} + (z + 6380)\hat{k}\ [km] \tag{6}$$

Before the total acceleration can be determined from the five-term acceleration value, all the components need to be in terms of the same basis. Upon further inspection, all the current values for position and velocity are in terms of the $\hat{i}$, $\hat{j}$, $\hat{k}$ basis except for one, $^{N}\overrightarrow{\omega}^{E}$. $^{N}\overrightarrow{\omega}^{E}$ is written in terms of $\hat{n_3}$, so we need a way to convert the direction $\hat{n_3}$ to the $\hat{i}$, $\hat{j}$, $\hat{k}$ basis. Figure 4 shows how $\hat{n_3}$ can be derived from $\hat{j}$ and $\hat{k}$.

By analyzing Figure 4, one can derive the following definition of $\hat{n_3}$ in terms of the angle ϕ, which is the latitude of the location P ($\hat{n_3} = cos(\phi)\hat{j} + sin(\phi)\hat{k}$). In our case, $\phi = 34°$. $^{N}\overrightarrow{\omega}^{E}$ becomes $^{N}\overrightarrow{\omega}^{E} = \Omega cos(\phi)\hat{j} + \Omega sin(\phi)\hat{k}$. Lastly, it is important to note the following two quantities to substitute into the five-term acceleration value: $^{N}\overrightarrow{a}^{P} = ^{N}\overrightarrow{\omega}^{E} \times (^{N}\overrightarrow{\omega}^{E} \times \overrightarrow{r}^{OP})$ and $^{E}\overrightarrow{a}^{\frac{Q}{P}} = \ddot{x}\hat{i} + \ddot{y}\hat{j} + \ddot{z}\hat{k}$. With these quantities, we can now make substitutions into our five-term acceleration value. After the proper substitutions have been made, the result is a vector equation for $^{N}\overrightarrow{a}^{Q}$ that can be split into three scalar equations, one for each direction of the basis. Those equations are as follows:

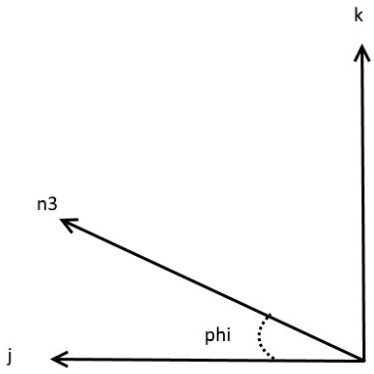

Figure 4: Basis Transformation

- $\hat{i}$: $a_x = \ddot{x} + 2\Omega\dot{z}cos(\phi) - 2\Omega\dot{y}sin(\phi) - \Omega^2 xcos^2(\phi) - \Omega^2 xsin^2(\phi)$
- $\hat{j}$: $a_y = \ddot{y} + 2\Omega\dot{x}sin(\phi) + \Omega sin(\phi)(\Omega zcos(\phi) - \Omega ysin(\phi)) + 6380\Omega^2 cos(\phi)sin(\phi)$
- $\hat{k}$: $a_z = \ddot{z} - 6380\Omega^2 cos^2(\phi) - 2\Omega\dot{x}cos(\phi) - \Omega cos(\phi)(\Omega zcos(\phi) - \Omega ysin(\phi))$

Additionally, substituting in our known values into our equations for $\overrightarrow{F_D}$ and $\overrightarrow{F_G}$ yields the following values for those forces:

$$\overrightarrow{F_D} = -\frac{1}{2}\rho C_D A\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}(\dot{x}\hat{i} + \dot{y}\hat{j} + \dot{z}\hat{k}) \tag{7}$$

$$\overrightarrow{F_g} = \frac{-\mu m_Q}{(6380 + z)^2}\hat{k} \tag{8}$$

Now, we have all the terms necessary to make the substitutions into Newton's second law. As before, the result is one vector equation, or three scalar equations. The three scalar equations for the $\hat{i}$, $\hat{j}$, and $\hat{k}$ directions are as follows:

- $\hat{i}$: $-\frac{1}{2}\rho C_D A\dot{x}\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} = m_Q(\ddot{x} + 2\Omega\dot{z}cos(\phi) - 2\Omega\dot{y}sin(\phi) - \Omega^2 xcos^2(\phi) - \Omega^2 xsin^2(\phi))$
- $\hat{j}$: $-\frac{1}{2}\rho C_D A\dot{y}\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} = m_Q(\ddot{y} + 2\Omega\dot{x}sin(\phi) + \Omega sin(\phi)(\Omega zcos(\phi) - \Omega ysin(\phi)) + 6380\Omega^2 cos(\phi)sin(\phi))$
- $\hat{k}$: $\frac{-\mu m_Q}{(6380+z)^2} - \frac{1}{2}\rho C_D A\dot{z}\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} = m_Q(\ddot{z} - 6380\Omega^2 cos^2(\phi) - 2\Omega\dot{x}cos(\phi) - \Omega cos(\phi)(\Omega zcos(\phi) - \Omega ysin(\phi)))$

It is obvious that these equations are pretty horrific. Matlab can easily compute the necessary cross products and numerically solve the differential equations for us. However, if we want an analytical solution, several simplifying assumptions need to be made. First, let us assume $C_D = 0$ to remove the drag term from each equation. Additionally, I would like to assume that the force due to gravity, F_g doesn't depend on the height of the rock, z. These assumptions would yield the following system of differential equations that we should be able to analytically solve:

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix}0 & 2\Omega sin(\phi) & -2\Omega cos(\phi)\\ -2\Omega sin(\phi) & 0 & 0\\ 2\Omega cos(\phi) & 0 & 0\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} + \begin{bmatrix}\Omega^2 & 0 & 0\\ 0 & \Omega^2 sin^2(\phi) & \Omega^2 sin(\phi)cos(\phi)\\ 0 & \Omega^2 cos(\phi)sin(\phi) & \Omega^2 cos^2(\phi)\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix} + \begin{bmatrix}0\\ -6380\Omega^2 cos(\phi)sin(\phi)\\ 6380\Omega^2 cos^2(\phi) - \frac{u}{6380}\end{bmatrix}$$

Sadly, after a bit of effort I was unable to analytically solve the system of equations. I believe that I need to make several more simplifying assumptions to reduce the equations further, but I am unsure what assumptions to perform. I believe I could make an assumption with reaching terminal velocity, where the acceleration in the $\hat{i}$ and $\hat{j}$ direction would be 0.

Even though I was unable to analytically solve the system of equations, it is still possible to numerically solve the system of differential equations using Matlab. Figure 5 shows the solution to the equations of motion when the drag value is 0. Figure 6 shows the solution to the equations of motion when the drag is a non-zero value. For both plots, the Matlab code used can be found in the appendix of this paper. In both plots, the graph ends once the rock reaches a height of -1500 m, when it hits the ground.

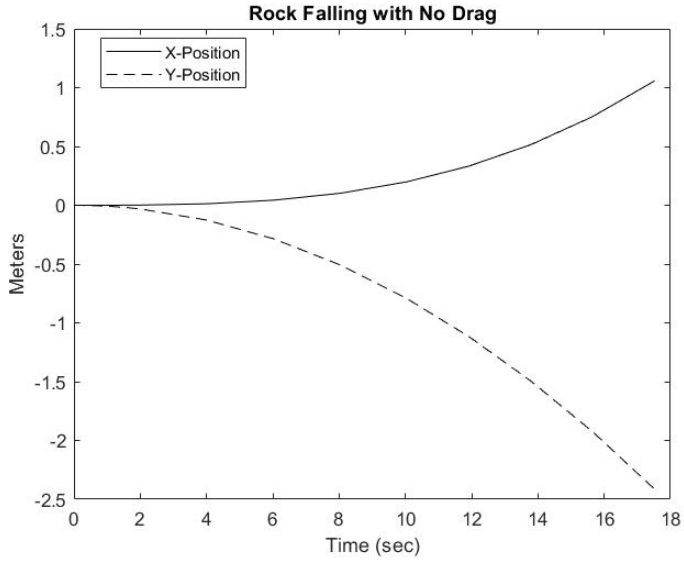

Figure 5: Position of the Rock as a Function of Time - No Drag

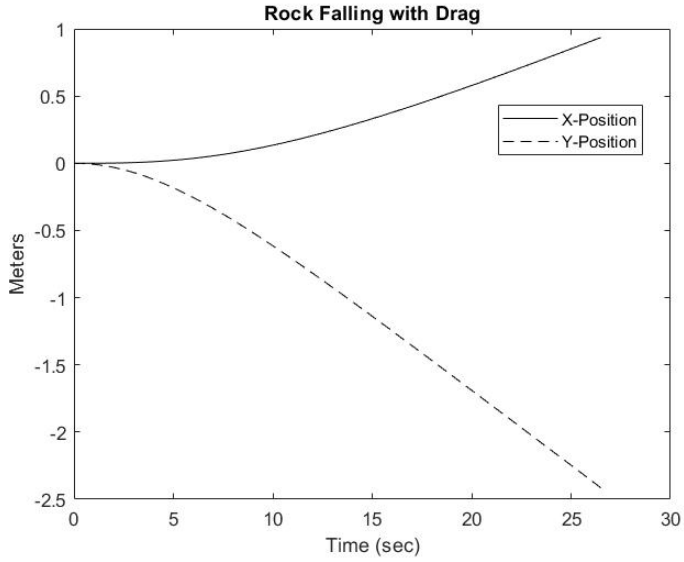

Figure 6: Position of the Rock as a Function of Time - Drag

2 Results

From this exercise, I learned a lot. I learned how to use the odeset function in Matlab to stop the numerical integration of a problem at a set value, which is something I hadn't previously known existed. In addition, I felt like this problem really challenged my skills in both coding numerical solutions and analytical. I discovered that my skills in solving differential equations analytically isn't quite where I would like them to be.

In regards to the solutions to the problem, it is obvious that the rotation of the earth has a small effect on the position the rock lands at. I doubt this effect would be noticeable if the distance fallen was any shorter. An interesting observation to note that made me double check my code is that the position the rock lands at doesn't appear to change when drag is introduced into the problem. At first, this confused me, but upon closer inspection it makes sense. The drag force could ever only oppose velocity in our case, since we assume there is no wind. As a result, the drag could ever only work against the motion of the rock. This is apparent when the timescale for both rocks is viewed. Both rocks end up in the same position, but the rock experiencing drag takes almost twice as long to fall. This makes sense, since we know that objects in a vacuum would fall at equal speeds and that air resistance is what causes objects to fall at different speeds on earth. This has seemingly been proven by our analysis of the rock falling.

3 Matlab Code

Contents


```
%Johnathan Corbin
%Spring 2019
```

Determining accleration

```
clear, clc

syms omega x y z xv yv zv xa ya za phi rho cD A
w = [0; omega*cos(phi); omega*sin(phi)];
rOP = [0; 0; 6380];
rPQ = [x; y; z];
aQP = [xa; ya; za];
vQP = [xv; yv; zv];
aP = cross(w, cross(w, rOP));

aQ = aP + aQP + cross(w, cross(w, rPQ)) + 2*cross(w, vQP);
```

Numerical No Drag

```
%global A CD rOP m rho Omega u d
global d A CD rOP m rho Omega u

A = pi * .05^2; %Frontal area of the sphere
CD = 0;
rOP = (6378 + 2) * 1000; %Position vector from center of earth to cliff
m = 1; %Mass of the rock.
rho = 1.225; %Density of air at sea level
Omega = [0; (7.292*10^-5)*cos(34 * pi / 180); (7.292*10^-5)*sin(34 * pi / 180)];
u = 398600 * (1000^3); %Gravitation coefficient
d = 1500;

fname = 'Canyon'; % Setting name of the .m file for the state variable time derivatives

initial_conditions = [0; 0; 0; 0; 0; 0];

Opt = odeset('Events', @GC_Event);
[time, y, te, ye, ie] = ode45(@Canyon, [0, 1e6], initial_conditions, Opt);
figure(1)
plot(time, y(:,1), 'k')
hold on
plot(time, y(:,2), '--k')
title('Rock Falling with No Drag')
legend('X-Position','Y-Position', 'Location', 'best')
xlabel('Time (sec)')
ylabel('Meters')
```

Numerical With Drag

```
CD = 0.42;
Opt = odeset('Events', @GC_Event);
[time, y, te, ye, ie] = ode45(@Canyon, [0, 1e6], initial_conditions, Opt);
figure(2)
```

```
plot(time, y(:,1), 'k')
hold on
plot(time, y(:,2), '--k')
title('Rock Falling with Drag')
legend('X-Position','Y-Position', 'Location', 'best')
xlabel('Time (sec)')
ylabel('Meters')
```

Functions

```
function dydt = Canyon(t, y)

global A CD rOP m rho Omega u

dydt = zeros(size(y));

F_g = -(u*m/(rOP + y(3))^2)*[0; 0; 1];

F_D = -(1/2) * rho * CD * A * sqrt(y(4)^2 + y(5)^2 + y(6)^2) * [y(4); y(5); y(6)];

rOQ = (rOP) * [0; 0; 1] + [y(1); y(2); y(3)];

acceleration = (F_g + F_D - m * (cross(Omega, cross(Omega, rOQ)) + 2*cross(Omega,[y(4);y(5);y(6)])))/m;

dydt(1) = y(4);
dydt(2) = y(5);
dydt(3) = y(6);
dydt(4) = acceleration(1);
dydt(5) = acceleration(2);
dydt(6) = acceleration(3);
end

function [value,isterminal,direction] = GC_Event(time,y)
% The second arguement of this function is the system state vector being
% integrated by ODE45
global d
%Event funtion to stop integration when rock hit -1500[m]
%if y(3) > -1500 %If rock is higher than -1500[m]
if y(3) > -d %If rock is higher than -1500[m]
 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
```