

Student

student@u.rochester.edu

CSC 290E

ODE Project

9 December 2010

Abstract

Systems in nature often depend on many variables that change with time, so modeling the behavior of these systems can sometimes be a complex affair. In order to model these systems, we use differential equations, which “express how related phenomena (variables) changing over time affect each other” (Brown). The main objective of the ODE project was to use MATLAB to numerically analyze differential equations. Results were gathered using both student-written derivative functions and MATLAB’s built-in ODE solving functions. The two main problems addressed were related to ballistics and environmental engineering.

Introduction

Differential equations can be used to model the behavior of systems in nature. Some applications include physics ($F=ma$, where $a=\frac{dv}{dt}$), electrical circuits, environmental engineering, population dynamics, and many other fields.

For this project, I focused on solving differential equations using MATLAB, not on deriving new laws or solving equations analytically. Deriving new laws can take years of study, and fun analytical methods of solving DEs can be learned in courses such as MTH 165. I specifically solved systems of ordinary differential equations (ODEs), which don’t use partial derivatives.

One way to address a differential equation is by using matrices. Take, for example, the 2nd order DE:

$$y'' + r(x)y' + s(x) = t(x)$$

We can rewrite this DE as a system of DEs:

$$\frac{d}{dx} \begin{bmatrix} y \\ y' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -s(x) & -r(x) \end{bmatrix} \begin{bmatrix} y \\ y' \end{bmatrix} + \begin{bmatrix} 0 \\ t(x) \end{bmatrix}$$

We can then use some sort of program to plug in values and generate vectors of results.

Another approach is numerical integration. Because a derivative, $\frac{dy}{dt}$, tells us how y changes with t , we can linearly approximate y using a small change in t (*we will call Δt*). We also need an initial condition, $y(0)$. Our estimation therefore becomes: $y(\Delta t) = y(0) + y'(\Delta t)$. A familiar name for this process of numerical integration is “Euler’s method.”

MATLAB has a powerful numerical integration tool, where the user needs to only supply a derivative function, a range of times, and an initial condition. MATLAB generates a vector of values that can be plotted to illustrate a solution to a DE.

Ballistics

In the ODE project I worked through two separate projects, each with multiple sub-problems. The first major project was a ballistics problem based around the battle of Crecy. I wrote a derivative function to calculate x' , y' , x'' and y'' . The differential equations were as follows:

$$x' = u$$

$$y' = v$$

$$x'' = - (1/(2m)) (\rho c a (u^2 + v^2) u / \sqrt{u^2 + v^2})$$

$$y'' = -g - (1/(2m)) (\rho c a (u^2 + v^2) v / \sqrt{u^2 + v^2})$$

I used the following base specifications:

muzzle velocity = 150 m/s

mass of cannonball = 5kg,

air density $\rho = 1.23$,

force of gravity $g = 9.81$,

drag coefficient $c = .5$

diameter $d = .1067$ m

area $a = \pi d^2 / 4$

I then used MATLAB's built in ODE solver ode23 to solve the ODEs. Additionally, I wrote a script to return the distance travelled by the projectile using the last positive y value in the answer vector.

Part 1: Some Experimentation

I first experimented by changing the values of constants such as launch angle and drag.

1. Launch Angle:

- We should see height increase as the launch angle approaches $\pi/2$
- Horizontal range should (as predicted by behavior in a vacuum) reach a maximum at some angle of launch close to $\pi/4$

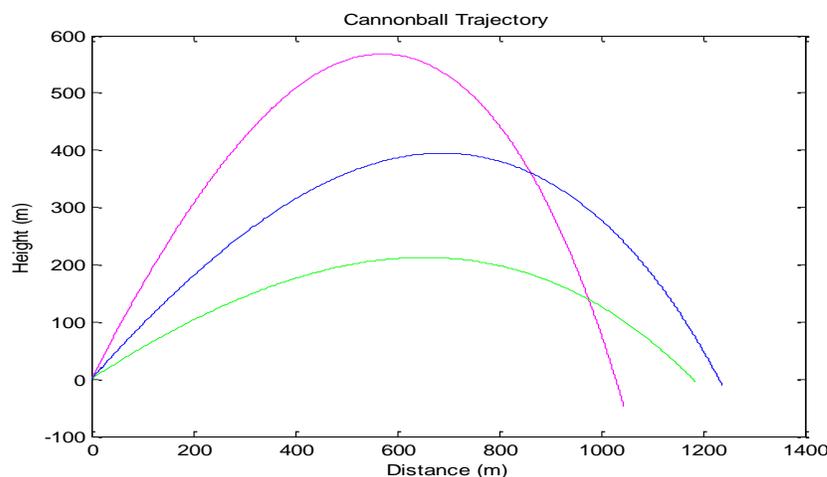


Figure 1

Figure 1 illustrates how launch angle affects trajectory. The green trajectory shows a launch angle of $\pi/6$, the blue trajectory shows a launch angle of $\pi/4$, and the pink trajectory shows a launch angle of $\pi/3$. The green ball travelled 1179.6 m, the blue

ball travelled 1230.8 m and the pink ball travelled 1028.7 m. Figure 1 confirms the prediction that increasing launch angle will increase height. Figure 1 also suggests that maximum range is indeed achieved at a launch angle near $\pi/4$, but we do not know that angle yet.

2. Drag

- a. We expect rougher surfaces to decrease in speed faster (greater magnitude of x'' and y''), therefore travelling shorter distances than smooth objects.

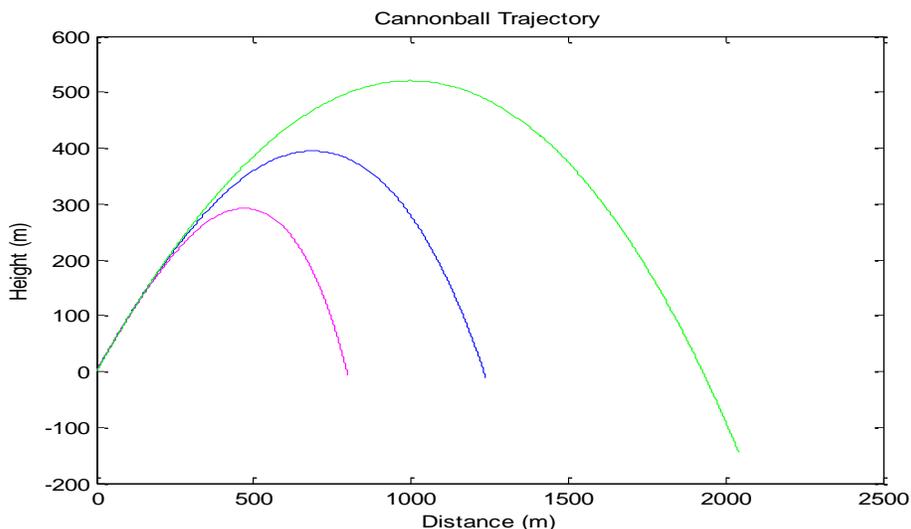


Figure 2

Figure 2 shows how drag affects trajectory. The pink trajectory has a drag coefficient of 1.2, blue has a drag of .5, and green has a drag of .1. Distances are as follows:

Green ($c=.1$): 1926.6 m

Blue ($c=.5$): 1230.8 m

Pink ($c=1.2$): 796.2 m

As predicted, increases in drag lead to decreases in range. This is intuitive, because more drag means a stronger force of air friction acting negatively on the ball.

Part 2: Maximizing Range

In part 1, we estimated that range is maximized at some launch angle near $\pi/4$. In part 2, I calculated the exact angle that maximizes range. Using a for loop, I ran my ODE solving script with launch angles ranging from 0 to $\pi/2$. My script kept track of the maximum range and corresponding angle.

With the base specifications listed above and $c=.2$, the angle that maximizes distance is 0.745731589 radians, or approximately 42.727 degrees.

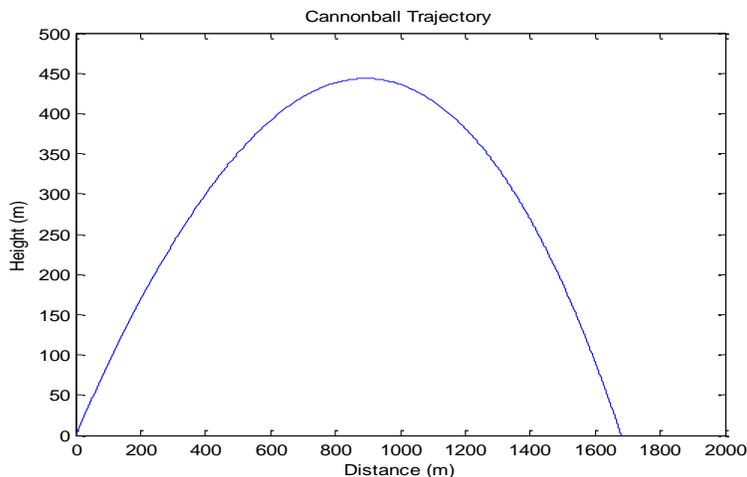


figure 3

Maximum projectile range = 1679.83 m.

In a vacuum, maximum distance is achieved when the launch angle is $\pi/4$. Figure 3 shows that we must sacrifice some height to achieve maximum range due to factors such as air friction.

Part 3: Unorthodox Warfare

In this part of the assignment I explored the trajectories of a non-traditional projectile, the meatball. I used a 6'' (15.24 cm) diameter meatball with a drag coefficient of 0.7. I imagine that a 1 lb club pack of ground beef from Wegmans could be rounded into a ball with a diameter of roughly 6''. To be safe, I estimated the weight of the meatball to be 1.5 pounds. Thus, mass is 0.68 kg. Using my optimization method described in part 2, I found the maximum range of my meatball to be 253.5 m with an elevation of .4284 radians (24.55°) and muzzle velocity of 300 m/s.

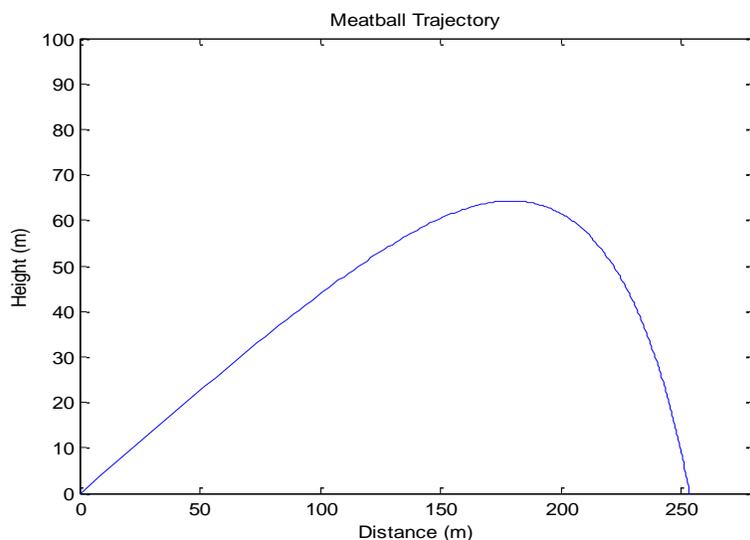


Figure 4

From figure 4 it may seem that a ball of meat could be a viable option if ammunition runs low. However, we have not yet taken into consideration the effect of wind on the projectile (we will address

wind in part 5). We will soon see that light-weight meatballs are affected greatly by wind, and blow off target quite easily.

Part 4: Variations in Air Density

Prior to this point in the project, we have assumed that air density is a constant value, ρ . Our base specifications set ρ to 1.23. However, actual air density varies with height. According to Wikipedia, air density can be expressed as:

$$\rho = \frac{p \cdot M}{R \cdot T}$$

where

$$p = p_0 \cdot \left(1 - \frac{L \cdot h}{T_0}\right)^{\frac{g \cdot M}{R \cdot L}}$$

$$T = T_0 - L \cdot h$$

and

sea level standard atmospheric pressure $p_0 = 101325$ Pa

sea level standard temperature $T_0 = 288.15$ K

Earth-surface gravitational acceleration $g = 9.80665$ m/s².

temperature lapse rate $L = 0.0065$ K/m

universal gas constant $R = 8.31447$ J/(mol·K)

molar mass of dry air $M = 0.0289644$ kg/mol

h = height in meters

I rewrote my derivative equations, using these equations to express air density as a function of height instead of as a constant.

Using the base specifications, a new cannonball trajectory looks like this (elevation of $\pi/4$):

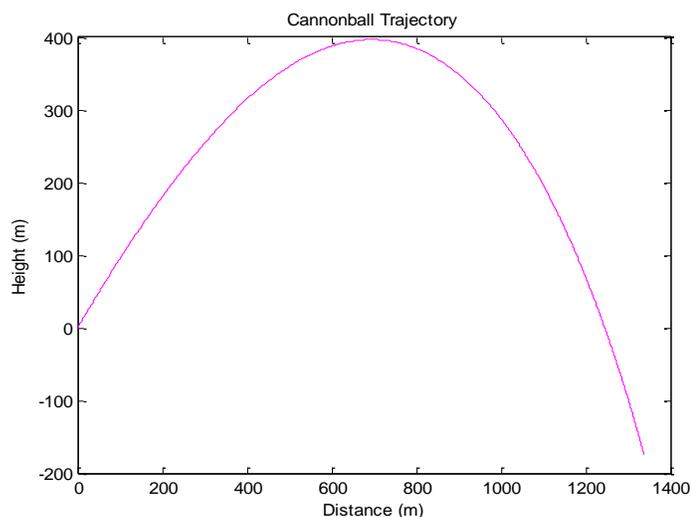


Figure 5

Now compared to constant air density:

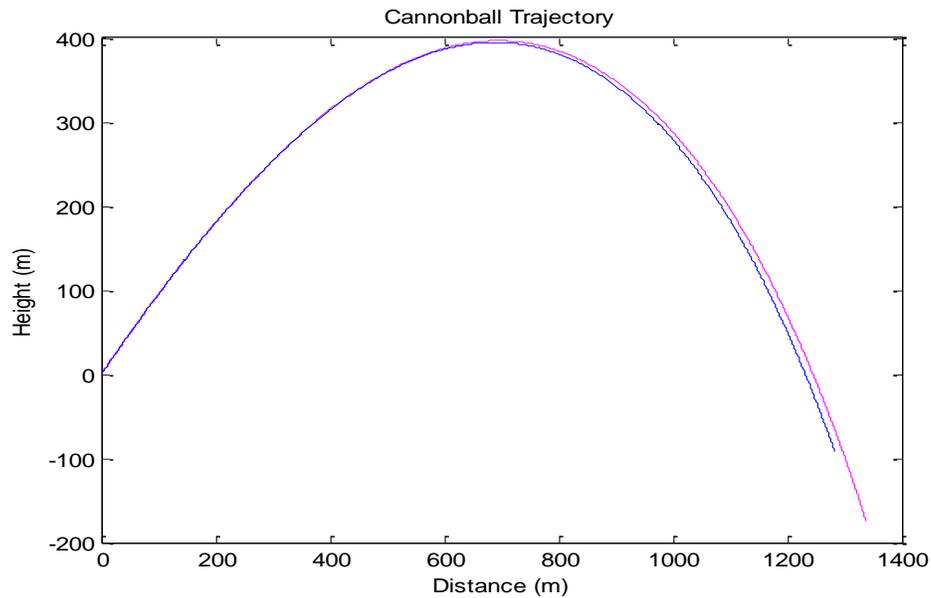


Figure 6

We can see from figure 6 that assuming constant air density slightly decreases range (the blue line is the trajectory assuming $\rho=1.23$). Under base specifications and elevation of $\pi/4$, the difference in range is approximately 13.8 m.

Range ($\rho=1.23$): 1230.81 m

Range (ρ changes with h): 1244.65 m

I also compared maximum range using a muzzle velocity of 300 m/s and drag coefficient of .2:

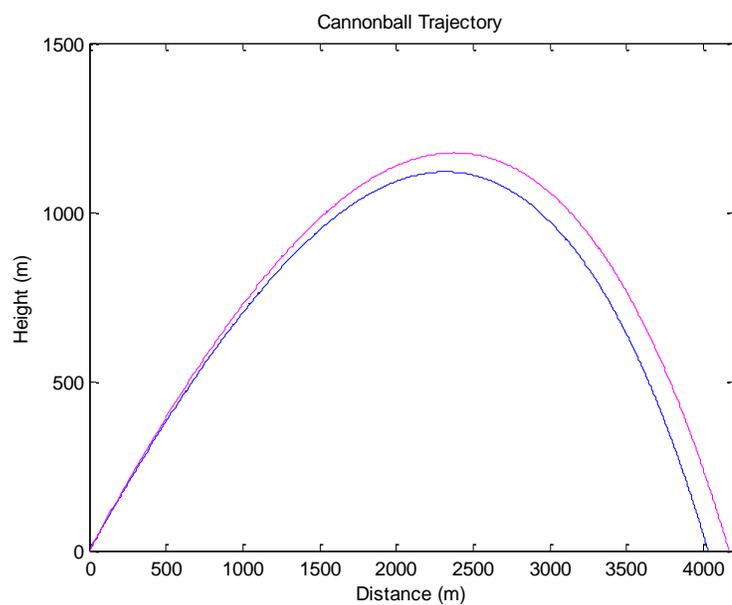


Figure 7

Figure 7 Shows the differences in trajectories between DEs with $\rho=1.23$ (blue) and with ρ varying with height (pink).

Max range (pink): 4165.6 m at 40°

Max range (blue): 4030.16 m at 39.1°

It seems intuitive that the cannonball will travel further when air density varies with height. Air density decreases as height increases, so the magnitudes of x'' and y'' when air density varies will be smaller in comparison to x'' and y'' when density is constant. A smaller magnitude of x'' and y'' means less resistance and more range.

Part 5: Wind

Now we must account for a west-east wind blowing. In order to model the affect of the wind on the cannonball, I edited my derivative functions and added z' and z'' . z'' looks similar to x'' , but instead of using u and v we use the velocity of the wind relative to the cannonball. My new equations looked like this:

$$z' = k$$

$$z'' = -(1/(2*m))*(\rho*c*a*(wvel-k)^2), \text{ where } wvel \text{ is wind velocity.}$$

I considered a wind of 100 mph, or 44.704 m/s. Under base specifications and at the launch angle that maximizes range, the trajectory as seen from above appears as follows:

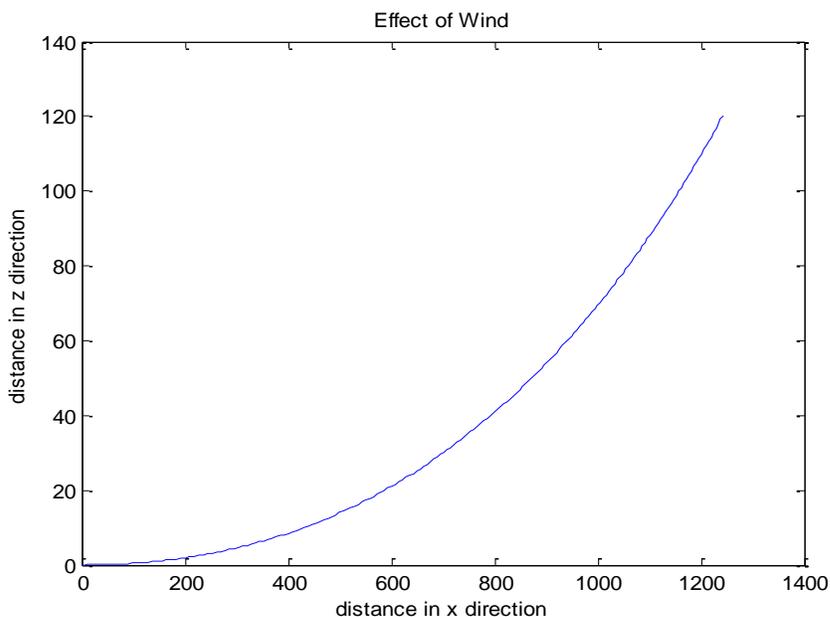


Figure 8

Figure 8 shows the wind's effect on the trajectory of the cannonball, blowing it approximately 120 m off target. The cannonball's path looks parabolic because it is being accelerated.

Do you remember our meatball projectile? When we account for wind, trajectory as seen from above looks like this:

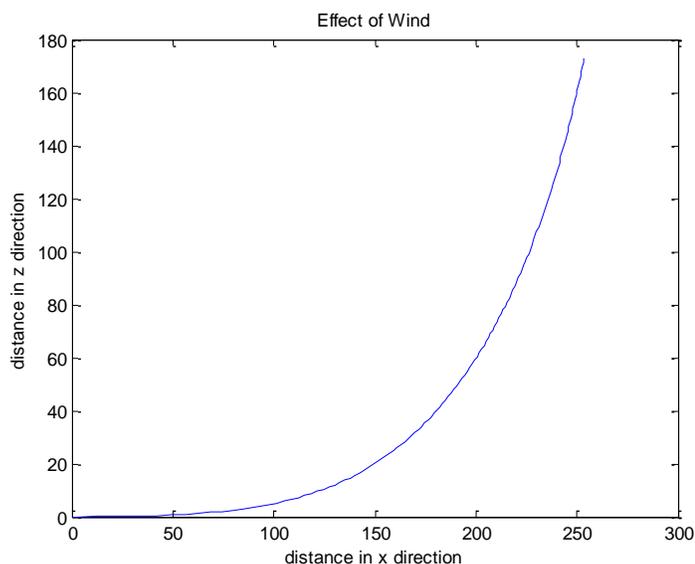


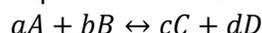
Figure 9

Figure 9 shows that a meatball will be blown almost 180 m off-target during the course of its 253 m flight. It seems, therefore, that it would not serve well as a long range projectile, but perhaps could still splatter some goop on enemy helmets.

Environmental Engineering

The second project in my ODE assignment dealt with environmental engineering. I obtained the individual problems from the reading, Environmental Engineering Science, by William W Nazaroff and Lisa Alvarez-Cohen. Predominantly, this section involved modeling reaction rates using differential equations and MATLAB DE solvers.

We can denote a kinetic chemical equation as follows:



Where a , b , c , and d represent stoichiometric coefficients and A , B , C and D represent chemical species. The Reaction rate, R_1 , tells us the number of occurrences of the above reaction per unit time. Often, R_1 is expressed in units of concentration per time. R_1 therefore describes the rates at which concentration of chemical species increase and decrease. For the above reaction,

$$R_1 = -\frac{1}{a} \left(\frac{d[A]}{dt} \right) = -\frac{1}{b} \left(\frac{d[B]}{dt} \right) = \frac{1}{c} \left(\frac{d[C]}{dt} \right) = \frac{1}{d} \left(\frac{d[D]}{dt} \right)$$

In order to express the overall rate law of the reaction, we need to only consider the concentration of the reactants in the reaction (in most cases). An example of a rate law is:

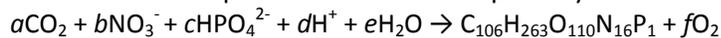
$$R_y = k_y [A]^\alpha [B]^\beta$$

where k_y is the rate constant, and α and β are empirical coefficients (Nazaroff, Alvarez-Cohen). The sum of $\alpha + \beta$ tells us the order of the reaction.

Part 1: Stoichiometry

Part one of this project is a Gaussian elimination exercise. Stoichiometry is “the application of the principle of material balance to a chemical transformation,” (Nazaroff, Alvarez-Cohen). By obeying

the law of conservation of matter, we can predict the value of some chemical coefficients based on others. The example I solved involved a photosynthesis reaction:



This problem yields six algebraic equations, five for unknowns and one for the conservation of charge:

$$\text{C: } a = 106$$

$$\text{O: } 2a + 3b + 4c + e = 110 + 2f$$

$$\text{N: } b = 16$$

$$\text{H: } c + d + 2e = 263$$

$$\text{P: } c = 1$$

$$\text{+/-: } -b - 2c + d = 0$$

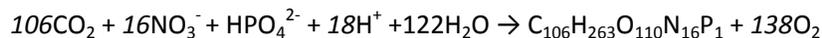
To solve these equations, I rewrote them in matrix form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 3 & 4 & 0 & 1 & -2 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & -2 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} 106 \\ 110 \\ 16 \\ 263 \\ 1 \\ 0 \end{bmatrix}$$

I then used my Gaussian elimination program in MATLAB to solve the system, and I got the result vector:

$$\begin{bmatrix} 106 \\ 16 \\ 1 \\ 18 \\ 122 \\ 138 \end{bmatrix}$$

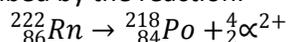
Plugging these values into the original chemical equation, we see:



Mass is conserved, so our equation is balanced.

Part 2: Radon

Part 2 of this project involved numerically modeling the radioactive decay of radon-222. The radioactive decay of radon can be described by the reaction:



The rate law for this reaction is $k = 2.1 \times 10^{-6} \text{ s}^{-1}$

Furthermore, the reaction is elementary, so the rate law is first order: $R = kC_{\text{Rn}}$

We can express the time rate of change in Rn concentration and the initial condition:

$$\frac{dC_{\text{Rn}}}{dt} = -kC_{\text{Rn}}$$

$$C_{\text{Rn}}(0) = C_0$$

From this point, instead of solving analytically, I wrote a MATLAB radon derivative function and used MATLAB to solve the differential equations numerically. I obtained the graph:

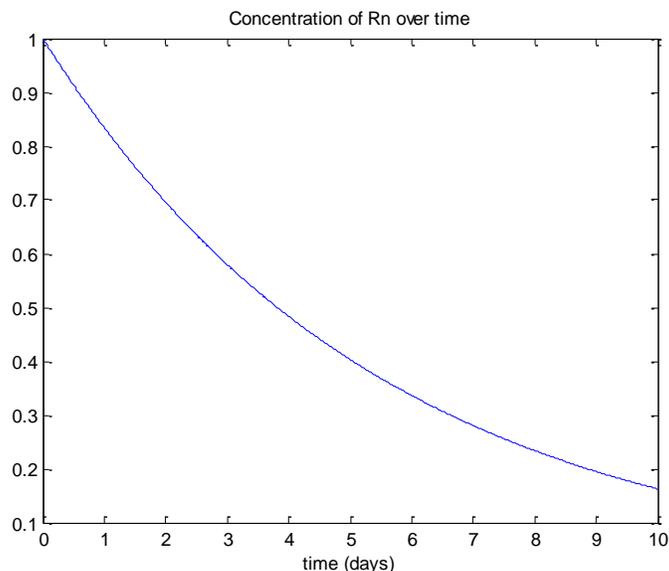


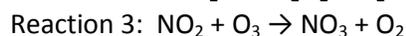
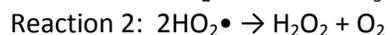
Figure 10

As expected, figure 10 shows the concentration of Rn decreasing exponentially toward zero. We can also see, for example, that after 4 days approximately half of the Rn has decomposed.

Part 3: Smog

The final part of this project involved using MATLAB to numerically solve rate laws of reactions associated with smog production. The task was to write rate laws to show “the net effect of these three reactions on the mole fractions of OH⁻, HNO₃, HO₂⁻, and NO₂,” (Nazaroff, Alvarez-Cohen).

The equations used were:



These reactions are elementary, so the rate laws can be expressed as:

$$R_1 = k_1 Y_{\text{NO}_2} Y_{\text{OH}\cdot}$$

$$R_2 = k_2 (Y_{\text{HO}_2\cdot})^2$$

$$R_3 = k_3 Y_{\text{NO}_2} Y_{\text{O}_3}$$

We can then express the change of concentrations in time:

$$\frac{dY_{\text{OH}\cdot}}{dt} = -R_1 = -k_1 Y_{\text{NO}_2} Y_{\text{OH}\cdot}$$

$$\frac{dY_{\text{HNO}_3}}{dt} = +R_1 = +k_1 Y_{\text{NO}_2} Y_{\text{OH}\cdot}$$

$$\frac{dY_{\text{HO}_2\cdot}}{dt} = -2R_2 = -2k_2 (Y_{\text{HO}_2\cdot})^2$$

$$\frac{dY_{\text{NO}_2}}{dt} = -R_1 - R_3 = -k_1 Y_{\text{NO}_2} Y_{\text{OH}\cdot} - k_3 Y_{\text{NO}_2} Y_{\text{O}_3}$$

Additionally, I think that $\frac{dY_{\text{O}_3}}{dt} = -R_3 = -k_3 Y_{\text{NO}_2} Y_{\text{O}_3}$

I wrote a MATLAB function to express these derivatives, but I was faced with the issue of finding k values. I addressed the issue in two different ways.

Method a:

I looked online, and found three potential values for k_1 , k_2 , and k_3 . They are:

$$k_1 = 1.336 \cdot 10^{11}$$

$$k_2 = 8.3 \cdot 10^5$$

$$k_3 = 4.284572 \cdot 10^7$$

When using MATLAB's built-in DE solver, I chose a uniform initial condition of $\frac{1}{7}$ mmol for each gas. I obtained the following results:

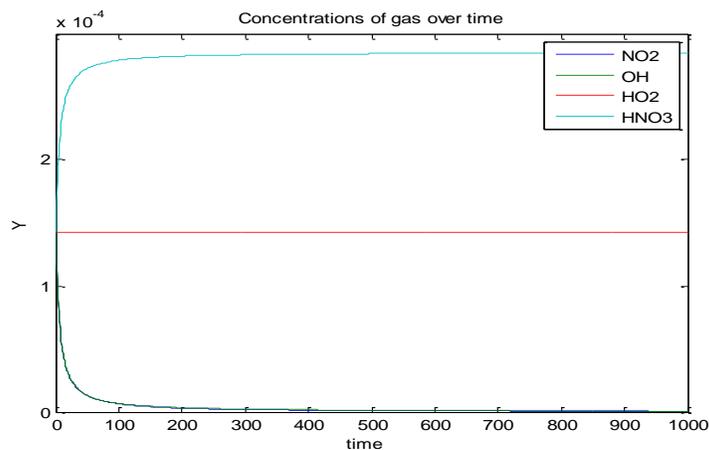


Figure 11

Figure 11 shows that OH and NO_2 almost instantly reacts away, while HO_2 reacts very slowly. As a result of the reaction between OH and NO_2 , the concentration of HNO_3 increases. The relative concentrations of OH and HNO_3 appear to be opposite in sign but equal in magnitude. This relationship should not surprise us, because $\frac{dY_{\text{HNO}_3}}{dt} = -\frac{dY_{\text{OH}}}{dt}$.

I found my results using my current values of k to be peculiar, because the reaction happened surprisingly quickly.

Method b:

I decided to choose arbitrary values of k . First I used $k_1=k_2=k_3=1$.

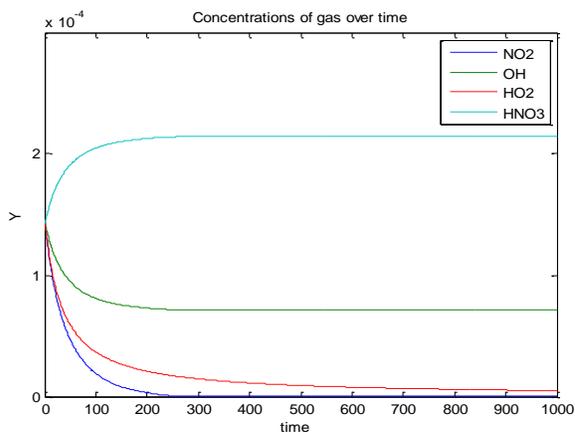


figure 12

I noted a few interesting qualities of figure 12:

First, as previously mentioned, HNO_3 increases as fast as OH decreases, as illustrated by both the original equations and their rate laws. Also, NO_2 was used up much faster than OH , because it has a role in two different reactions.

Next, I changed k_2 to 2.

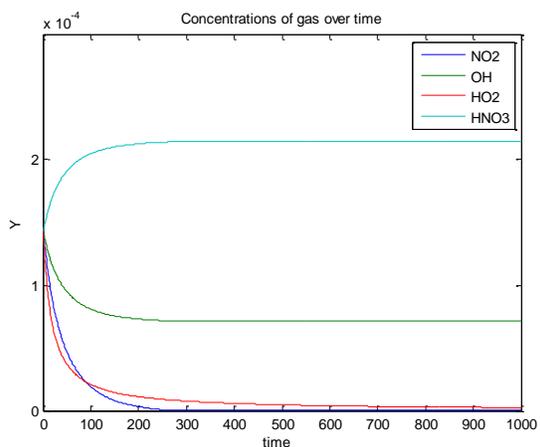


Figure 13

Figure 13 shows concentration over time when $k_1=k_2=1$ and $k_2=2$. It appears that only the concentration of HO_2 was affected. This makes sense after we look at the differential equations. Only $\frac{dY_{\text{HO}_2}}{dt}$ incorporates k_2 .

Next, I changed k_1 to 4.

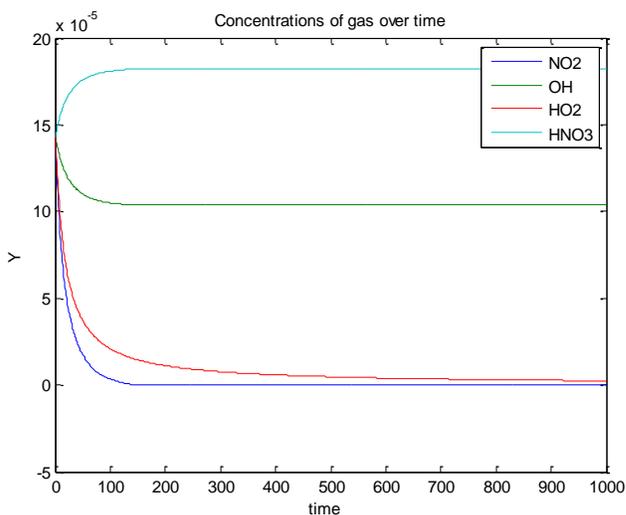


Figure 14

Figure 14 shows concentrations over time when $k_1=4$, $k_2=2$, and $k_3=1$. Figure 14 does not look much different than figure 12. However, in figure 14, the concentration of NO_2 approaches zero much more rapidly than in figure 12. Furthermore, both OH and HNO_3 also seem to react and “level out” faster in

figure 14 than in figure 12. Thus, we can see that increasing k values really does increase the rate of a reaction.

References

Environmental Engineering Science, by William W Nazaroff and Lisa Alvarez-Cohen

http://en.wikipedia.org/wiki/Density_of_air

“Informal Introduction to Differential Equations” by Chris Brown

<http://www.atmos-chem-phys.org/10/10521/2010/acp-10-10521-2010-supplement.pdf>

http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6TFN-3VJ90CF-5&_user=483663&_coverDate=01%2F11%2F1999&_rdoc=1&_fmt=high&_orig=search&_origin=search&_sort=d&_docanchor=&view=c&_searchStrId=1568194916&_rerunOrigin=google&_acct=C000022660&_version=1&_urlVersion=0&_userid=483663&md5=d7496f03fb60af3b815213798025b948&searchtype=a

<http://uk.answers.yahoo.com/question/index?qid=20100729030905AA8PYj0>