**CAP Tutorial: Mathematical Modeling**

**June 19, 2013**

*Modeling a Zombie Apocalypse using ODEINT*

This example demonstrates how to solve a system of first order ODEs using SciPy. Note that a Nth order equation can also be solved using SciPy by transforming it into a system of first order equations. In this lighthearted example, a system of ODEs can be used to model a "zombie invasion", using the equations specified in Munz et al. 2009.

The system is given as:

dS/dt = P - B\*S\*Z - d\*S

dZ/dt = B\*S\*Z + G\*R - A\*S\*Z

dR/dt = d\*S + A\*S\*Z - G\*R

with the following notation:

S: the number of susceptible victims

Z: the number of zombies

R: the number of people "killed"

P: the population birth rate

d: the chance of a natural death

B: the chance the "zombie disease" is transmitted (an alive person becomes a zombie)

G: the chance a dead person is resurrected into a zombie

A: the chance a zombie is totally destroyed



This involves solving a system of first order ODEs given by: dy/dt = f(y, t), where y = [S, Z, R].

**Part 1**

Let’s first get a copy of the python code we’ll be using for today’s exercise. In a terminal, please enter the following:

cd $HOME

mkdir modeling

cp /afs/physics.unc.edu/users/l/lam9/public/zombie.py $HOME/modeling/zombie.py

cd modeling

To run the code, you can then do the following:

ipython

run zombie.py

This script should both display a figure showing the number of zombies and people over time and save the figure automatically as a .png file called Zombie-R0=0-P=0.png. Note that I named the file to have some information on the parameters I used (zombie code, initial removed people set to zero, and birth rate set to zero). The following line of code to the text saves the figures:

savefig('Zombie-R0=0-P=0.png', dpi=100)

When you change the code, please also change the name of the picture files so that you don’t write over the old ones. Take note of the time interval over which there is a quick transition from living to zombies.

Now change the initial conditions so that some of the initial population is dead at the beginning. To do this, change the code as follows:

R0 = 0.01\*S0 # 1% of initial pop is dead

How did increasing this initial condition affect the time at which there are more zombies than people?

Now change the initial conditions so that there are 10 new births daily.

P = 10 # 10 new births daily

How did the shape of the graph as t-> 10 change and why? Assuming that zombies cannot have children, is this model realistic when the number of susceptible people is small? How could the model be changed? Feel free to change other parameters to see how it affects the results.

**Part 2**

Save the zombie.py file as zombie.py. Set the parameters back to where we started (P=0, R0=0). Let’s change the dynamics so that when zombies are killed, they cannot be resurrected again. We will need to create a new compartment for this group, and let D be the number of completely removed (cannot be resurrected). The system of equations can be rewritten as follows:

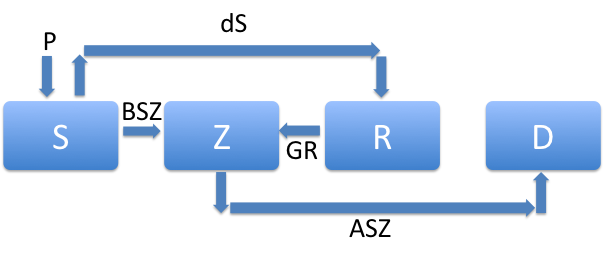
dS/dt = P - B\*S\*Z - d\*S

dZ/dt = B\*S\*Z + G\*R - A\*S\*Z

dR/dt = d\*S - G\*R

dD/dt = A\*S\*Z

And the corresponding diagram of movement between the groups can be drawn as follows:



We need to change add dD/dt and its initial condition in several places in the code. Here is an overview of the changes you need to make:

1) In def f(y,t), add Di=y[3] to hold the values of the totally removed people.

2) In def f(y,t), add f3 = A\*Si\*Zi, and change the equation for f2.

3) The vector that gets returned should now be [f0, f1, f2, f3]

4) In the initial conditions, add D0=0

5) In the vector that holds the initial conditions, y0, add D0.

6) Set D = soln[:, 3].

7) Plot the removed and totally removed by adding the lines

plt.plot(t, R, label='Removed')

plt.plot(t, D, label='Totally Removed')

Why does the number of totally removed not increase after some time? You could explore whether or not it is actually increasing by changing the final time, Tf, to something larger. How could this be changed in the model so that the number of totally removed slowly increases until all of the zombies are gone?

**Part 3**

It’s now your turn to create your own zombie model. You can create new groups (such as infected people who are not yet zombies), you can allow the zombies to reproduce, you can add more than one type of zombie. Use your imagination! To organize your thoughts with a partner, you might do the following:

1) Draw a diagram of the groups and how each type moves between compartments.

2) Write down the system of equations that describes this situation.

3) Save a copy of the zombie.py code as myzombie.py.

4) Alter the code to solve your model system of equations.

5) Be ready to share your results with the class!