CBE 5790 Modeling and Simulation (Autumn 2018)

Midterm Project

Deadline for uploading files to Carmen: Fri, Oct 26 at 11:59 PM

Use the information given in Worksheet 3 (*Big Trouble on Little Elba*) as the basis for this problem. Your mission is to modify this model to take into account the following two additional pieces of information:

- The company that produces the flu vaccine can provide n_{V0} doses of vaccine at time zero and then an additional n_{Vadd} doses on day t_{Add}.
- A certain number of healthy residents of Elba refuse to get vaccinated, either because of safety concerns, trypanophobia, aichmophobia, belonephobia, enetophobia, personal beliefs, or simple laziness. These "free riders" benefit from the fact that others are being vaccinated – they have less chance of encountering a sick person if lots of people are getting vaccinated – without doing their part to provide this same benefit to everyone else. Such people benefit from the actions of others without assuming their share of the cost or risk for an activity whose effectiveness depends upon a high degree of public participation.

The European Centre for Disease Prevention and Control (ECDC) asks you to develop simulations to explore the following question: Assuming 31,999 healthy people and one person sick with the H3N2 flu at time zero, how does the expected death toll vary depending on the number of free riders, the total number of vaccine doses initially available, and the number of additional doses available on day t_{Add}?

You will generate and compare results from two different models: 1) the continuous-variable deterministic model, obtained by solving the appropriate system of ODEs; 2) the discrete-variable stochastic model, obtained by applying the Gillespie algorithm. Note that for the continuous-variable approach, the addition of vaccine doses on day t_{Add} introduces a discontinuity; this is easily handled by breaking the simulation into two parts: a time interval from 0 to t_{Add}, and then a second time interval from t_{Add} to the final time.

Your function must create a single figure containing two subplots: one showing results from the continuous-variable deterministic model and the other showing results from the discrete-variable stochastic model.

- The figure should contain one row and two columns of subplots. Remember to properly label your subplots (titles, axes, legends, etc.). As described below, these plots will either be time plots of the various human populations or death toll histograms.
- The subplot for the continuous-variable deterministic model results must have four curves showing how the numbers of healthy (including free riders), sick, immune, and dead people vary with time (do not plot the number of vaccine doses).

- If *nRun* = 1 (see function definition below), then the subplot for the discrete-variable stochastic model results must have four curves showing how the numbers of healthy (including free riders), sick, immune, and dead people vary with time (do not plot the number of vaccine doses).

- If *nRun* > 1, the subplot for the discrete-variable stochastic model results must show a histogram of the final death toll results obtained by running the stochastic model *nRun* times.

Your program must be entirely contained in a single file named **lastname_firstname_elba.py**. The main function must be named **elba** and must be designed as follows:

```
def elba(n0, nVadd, tAdd, timeSpan, nMax, nRun):
```

	name	data type	default value	description
parameters	*n0*	tuple of ints	(31999, 0, 1, 0, 0, 0)	Initial numbers of (healthy people who are willing to be vaccinated, healthy people who refuse to be vaccinated (free riders), sick people, immune people, dead people, doses of vaccine)
	nVadd	int	0	Number of additional doses of vaccine available at time *tAdd*. Must be ≥ 0
	tAdd	float	None	Time (day) at which additional doses of vaccine become available. If *nVadd* > 0, *tAdd* must be > 0 and < *timeSpan*
	timeSpan	float	120 days	Time length for simulation
	nMax	int	2,000,000	Maximum number of time steps
	nRun	int	1	Number of times to run the discrete-variable stochastic simulation; must be ≥ 1
out	None (the only output from your function are the plots created, as described above)			

Your program will contain other functions that you design, but a user will interact with your program only through the **elba** function; your other functions will be invisible to the user and therefore you only need to provide a detailed docstring and error handling for the main function (**elba**).

Be sure to review the examples discussed in class. As described in the lecture notes for the Gillespie algorithm, the parameter *nMax* is used in some cases to determine when to end a simulation of the stochastic model.

Structure of your program

The design of this program is largely up to you - there is no one best way to do it, so we expect everyone to submit a unique and original solution.

It is not a good idea to try to pack everything into the main function **elba**. Instead, think about how you can break the tasks into separate functions that can be called from the main function. For example, one possible structure for your **lastname_firstname_elba.py** program file could be:

```
import statements

def elba(…):
```

This is the only function the user calls. It collects input, handles errors if necessary, and then calls the modeling functions to calculate what's needed to create the plots. The parameter nRun determines how many times the Gillespie model function will be called. You could then construct plots here, or define another function for plotting.

```
def ode_model(…):
```

Solve the system of ODEs for the continuous-variable deterministic model.

```
    def dydx(…):

    def Jacob(…):

    return t, n
```

```
def Gillespie_model(…):
```

Apply the Gillespie algorithm for the discrete-variable stochastic model.

```
    return t, n
```

What to submit: In addition to uploading your program (a single .py file), write a technical memo to summarize your approach and analysis, including helpful graphs and your observations and conclusions. Your audience for this technical memo is a group of scientists at the ECDC.

Guidelines for preparing your technical memo:

MEMORANDUM

Date: October 26, 2018
From: *firstname lastname*
To: Surveillance and Response Support, European Centre for Disease Prevention and Control
Subject: Modeling the H3N2 influenza epidemic on Elba

The contents of your technical memo begin here. The key points to consider for the content of your technical memo:

1. You must have a proper memorandum header similar to the example shown above.
2. This memo is **limited to one page, single-column, single-spaced.**
3. This memo should include the following: **purpose, approach**, **results and discussion**, and **conclusion**. Remember to keep each paragraph brief – only the important points matter!
4. Figures or tables which support your work should be included as additional pages in an "**Attachments**" section. Clearly label all figures/tables and reference them in your discussion.
5. Cite any external sources you use, if necessary.

Several scenarios and questions ECDC scientists wish to explore are listed below. Your memo should provide answers to these questions.

- How many people will die if there are 10,000 doses of vaccine initially available, 15,000 additional doses of vaccine available on day 20, and if 30% of the residents refuse to be vaccinated (9600 free riders)? How does the outcome change if the additional vaccine is not available until day 30? Day 40?
- If there are 9600 free riders, what's death toll should we expect if 50,000 doses of vaccine are available at time zero (and no more at any later time)?
- More generally, they are interested in understanding how these variables affect the relationship between death toll and time. Does the expected number of deaths depend more strongly on certain variables than others?