Mini-project 1: Contagion

CX 4230, Spring 2023[1]

Spring 2023: Due Feb 19 at 11:59 pm

[1] Last updated: `Tue Feb 7 14:02:35 EST 2023`.

In this mini-project, you will simulate two infectious disease models, one that we give you and one that you devise on your own. This assignment is inspired by Chapter 21 of the book, *Scientific Computing with Case Studies* ("SCCS"), by Dianne O'Leary, which should serve as a useful reference.[2]

[2] Dianne O'Leary. *Scientific computing with case studies*. SIAM, 2009. DOI: 10.1137/9780898717723. Available via the GT Library: [Link]

Here are the basic ground rules:

- You may work individually or in pairs. See section 2.
- You can implement your solution in any programming language.
- You will submit a written write-up (PDF file) *and* your source code. See section 3.
- The assignment is due on Sunday, February 19, at 11:59 pm. See section 3 for the late policy.

1 *Your Assignment: Implement a "Well-Mixed" SIR Model*

Consider the model of a disease presented as "Model 1" in SCCS Chapter 21. In that model, there are three types of people:

1. *susceptible* people, who have never had the disease;
2. *infected* people, who currently have the disease;
3. and *recovered* people, who had the disease once but no longer have it.

The disease can only spread from infected people to susceptible people. An infected person stays sick for some period, but then recovers; once they have recovered, they can never get sick again.

Let's assume the behavior of the disease is governed by the following mathematical model.[3] Denote time by a continuous variable $t \geq 0$, and let the state of the system be represented by the three-dimensional vector,

[3] See the explanation in SCCS Chapter 21 for a more detailed explanation.

$$\vec{x} = \vec{x}(t) = \begin{bmatrix} S(t) \\ I(t) \\ R(t) \end{bmatrix}, \tag{1}$$

where $S(t)$, $I(t)$, and $R(t)$ denote the fractions of the population who are susceptible, infected, or recovered, respectively, at time t. Further assume no one is born or dies, so that $S(t) + I(t) + R(t) = 1$ for all time.[4]

[4] This constraint expresses a *conservation law* for this model.

Suppose the state $\vec{x}$ evolves as an ordinary differential equation,

$$D\vec{x} = \vec{f}(\vec{x}), \tag{2}$$

where $\vec{f}(\vec{x})$ is the vector function,

$$\vec{f}(\vec{x}) = \begin{bmatrix} -\tau S(t)I(t) \\ \tau S(t)I(t) - \dfrac{I(t)}{\kappa} \\ \dfrac{I(t)}{\kappa} \end{bmatrix}. \tag{3}$$

The parameter $\tau \geq 0$ measures how quickly the disease can spread, with higher values corresponding to faster rates of spread. The parameter $\kappa > 0$ measures how quickly an infected person recovers, with higher values corresponding to slower (longer) recovery times.

1.1 *Question 1.1 (5 points)*

What are the fixed points of this system? Classify their stability.

1.2 *Question 1.2 (10 points)*

Similar to "Challenge 21.1" of SCCS, implement a computer simulation of this model. You can use any off-the-shelf ODE solver for this task.[5]

Use your simulator to conduct the following experiment.

- Let the initial conditions be $I(0) = 0.01$ (initial infected population of 1%), $S(0) = 0.99$, and $R(0) = 0$.
- Let $\tau = 0.8$ and $\kappa = 4$.
- Run a simulation until $I(t) < 10^{-4}$, the "stopping condition" for this experiment. At what time t does this condition occur? Report your stopping time results to two digits after the decimal point.[6] Create a plot that shows how $S(t)$, $I(t)$, and $R(t)$ vary in time. A sample plot appears in fig. 1.
- Run another simulation for $\tau = 0.4$ and $\kappa = 4$ with the same stopping condition. Again, note the time t at which the stopping condition occurs. Create another plot showing the state variables over time.
- Run a third simulation for $\tau = 0.8$ and $\kappa = 8$, note the stopping time, and create a third plot.
- Summarize what you observe across the three plots. For example, how do the stopping times compare? If any curves exhibit peaks,

[5] Challenge 21.1 suggests using MATLAB's `ode23s`, but you can use any reasonable equivalent. For example, if you are using Python, you can use Scipy's `integrate.solve_ivp` with its default solver, `'RK45'`.

[6] Since you often do not know how long to run a simulation, many off-the-shelf ODE solvers allow you to define "events," that is, a stopping criteria like this one. So try to use such a facility if your ODE solver has one. If your solver does not have one, explain what you did to determine the stopping time.

how do the peaks change (e.g., how do they shift in time and magnitude)?

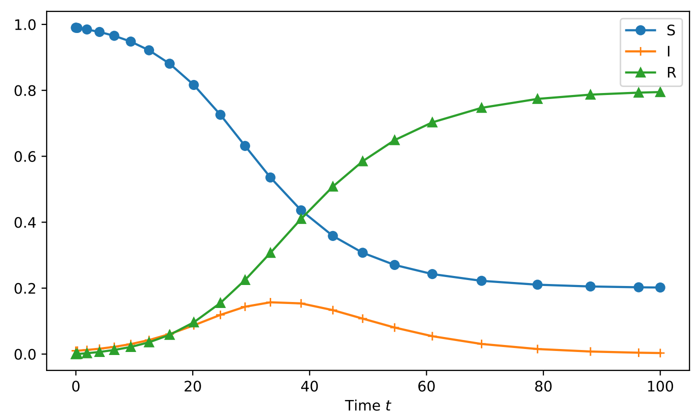

Figure 1: A sample plot of the state variables, $S(t)$, $I(t)$, and $R(t)$, in the unvaccinated model for some choice of τ and κ (not shown).

1.3 *Question 1.3 (pairs only; 10 points)*

If you are working in a team of two, then conduct this additional experiment using your simulator from Question 1.2. Try a variety of values of $\tau \in (0, 4]$ and $\kappa \in [1, 5]$.[7] For each combination of τ and κ, use your simulation to determine at what time t the number of infections falls below 10^{-4}. Create a 2-D contour plot or heatmap of t as a function of τ and κ and summarize in words what you observe.

[7] Say, 8–10 values for each, or a total of 64–100 combinations.

1.4 *Question 1.4 (10 points)*

Suppose there is a vaccine that can be given to *susceptible* people. A susceptible person who receives a vaccine can no longer become infected. However, this vaccine is very expensive, so you cannot vaccinate all susceptible people.

Modify the model to introduce a "policy" of vaccination. That is, introduce a fourth state variable, $V(t)$, that is the fraction of the population who are vaccinated at time t, such that the conservation law, $S(t) + I(t) + R(t) + V(t) = 1$, holds. Again, only susceptible people can become infected; if you vaccinate people, $S(t)$ should decrease and $V(t)$ should increase by the same amount. Your policy would be implemented as some formula for $\frac{dV}{dt}$, which you get to design, as well as modifications to the other derivative formulas to preserve the conservation constraint. Explain your approach.

Run three experiments to compare your policy against the three unvaccinated experiments from Question 1.2.

1.5 Question 1.5 (pairs only; 10 points)

Run an additional set of experiments to compare your policy against the unvaccinated scenarios of Question 1.3. We will give extra credit to the team with the best policy, that is, one that appears to require the fewest vaccinations while minimizing the total number of infected individuals.

2 Teaming

You may work individually or in pairs. Teams of two have additional requirements for the assignment, as noted above.

To "declare" your team, do the following:

- Visit the Canvas "People" page for Mini-project 1 ("MP1").[8]
- If you are working individually: Pick an unused team number and add yourself.
- If you are working in a team of two: Both of you should pick an unused team number and add yourself to the same unused team number.

[8] Link: `https://gatech.instructure.com/courses/310452/groups#tab-34531`

3 How to Submit

Create a *private* git code repository at `github.gatech.edu` (*not* `github.com`!) For teams, only one person needs to create the repository. Place both your code and a PDF report summarizing your results in this repo. You'll submit the URL to this repository on the page for this assignment in Canvas. In the repository itself, create a `README.md` file that tells us a) your individual or team number and member(s) of the team; b) which file is your PDF report; and c) what the organization of your code is (so we can inspect and evaluate it).

Important! Since your repository will be private, please be sure to add *everyone* from the teaching staff to your repo: the two TAs, Albert Chen (`achen355`) and Varun Lakshmanan (`vlakshmanan30`), and Prof. Vuduc (`rvuduc3`). Otherwise, we will not be able to see and grade your submission and you will get a zero.

Late submissions. There is a 10% penalty if you submit up to one day late (Monday, February 20), a 25% penalty if you submit two days late (Tuesday, February 21), and 50% penalty if you submit three days late (Wednesday, February 22). No assignments will be accepted after that time.

References

Dianne O'Leary. *Scientific computing with case studies*. SIAM, 2009. DOI: 10.1137/9780898717723. Available via the GT Library: [Link].