ENG3 Final Project

Due June 12, 2020, 5:00pm

Modeling the Covid-19 pandemic

The SIR (susceptible-infected-removed) epidemiological model consists of a simplified mathematical model for the spread of infectious diseases. The idea of the SIR model is simple: consider a population of N individuals. We then separate the population into three groups: susceptible, infected and removed (deceased). The sizes of these subpopulations at time t are denoted by $S(t)$, $I(t)$, and $R(t)$, respectively. Mathematically, this model can be written as a system of three coupled, non-linear ordinary differential equations (ODEs)

$$\frac{dS}{dt} = -\beta SI, \tag{1}$$

$$\frac{dI}{dt} = \beta SI - \gamma I, \tag{2}$$

$$\frac{dR}{dt} = \gamma I, \tag{3}$$

where we assume that the disease transmission rate $\beta = \beta(t)$ and the mortality rate $\gamma = \gamma(t)$ are real, time-dependent functions. Given that the ODEs are nonlinear and coupled, finding an analytical solution for Eqs. (1)-(3) is impractical. Nonetheless, one can search for solutions numerically, using, for example, ODE solvers such as the ones available in Matlab. We will use this classic epidemiological model to model the spread of Covid-19 in Santa Barbara County.

Part 1: Exploring the data (10 points)

Your first task consists of loading the data file `covid_data.mat` provided on GauchoSpace and plotting its contents. The file consists of official data reported by local authorities in the interval from March 15 to May 13 in Santa

Barbara County. It has 2 columns: the first contains the **cumulative number of infected individuals in the population at a certain date**, and the second contains the **cumulative number of deaths recorded on the same date**.

(a) Create a script named `explore_data.m` that loads the file `covid_data.mat` provided.

(b) Still in your `explore_data.m` script, create a figure with two plots: i) number of infected individuals vs time and ii) deaths vs time. Make sure that you:

- Change the line thickness to 3 for each line.
- Use legends "Infected Individuals", "Deaths" to identify each curve.
- Name the x and y axes as "Time (days since March 15)" and "Number of individuals", respectively. You can use `legend('location','best')` to move your legend box to the top left corner, so it doesn't cover your plot.
- Give your plot the following title: "Covid-19 in Santa Barbara County".
- Turn the grid on using the command `grid on`.
- Increase the font size of legends, axis labels and title using the command `set(gca,'FontSize',20)`.

At the end, your plot should look like Fig. (1). **Upload `explore_data.m` to GauchoSpace**.

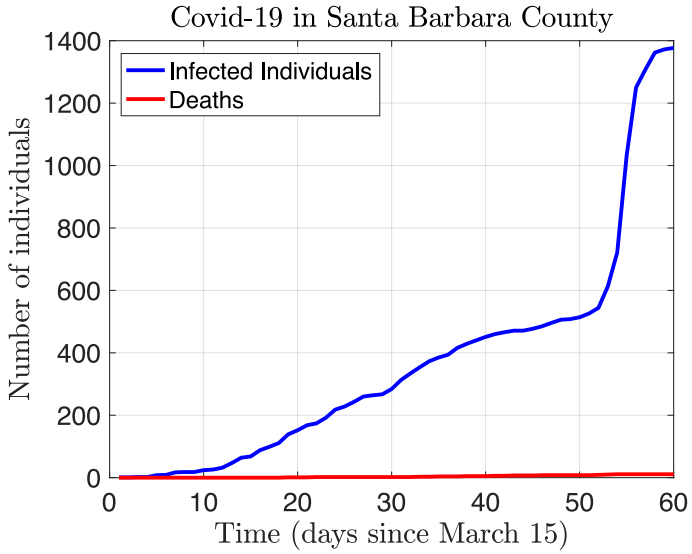

Figure 1: Plot of the number of individuals infected and deaths in Santa Barbara County due to Covid-19.

Part 2: Estimating the parameters from the data (20 points)

Now that you have familiarized yourself with the data, let us come back to the SIR model. Our goal is to use it to predict the growth of the number of infected individuals and as well as the number of deaths in the near future. But the problem is that we don't know the values for the functions $\beta(t), \gamma(t)$ in Eqs. (1)-(3). Despite that, observing the data from places where the Covid-19 epidemic got under control via quarantine measures can give us some hints about these parameters. For example, it is reasonable to consider that both the infection and the mortality rates decreased as more and more people started taking prophylaxis measures, e.g., stay-at-home shelters, self-quarantine, crowd control in public transportation systems and the usage of masks to avoid the spread of the virus.

Based on this information, let us postulate that both the infection and the mortality rate decreased exponentially after the quarantine measures were implemented, i.e.,

$$\gamma(t) = 0.0247e^{-0.111t}, \tag{4}$$

$$\beta(t) = \beta_0 e^{-0.0506t}, \tag{5}$$

where β_0 is a constant to be determined. Your task will be to estimate β_0

using the data you plotted in Part 1.

There exist many methods to estimate parameters from a given set of data points. For simplicity, you are going to use a Euclidean norm approach to minimize the error in the total number of cases $C(t) = I(t) + R(t)$. The idea is simple: solve Eqs. (1)-(3) for various β_0 values within a range, and for each of these values, compute the error given by the norm of the difference between the model results $C(t)$ and the data, i.e.,

$$err = \sqrt{\frac{\sum_{i=1}^{Ndays}(I(i) + R(i) - \text{covid\_data}(i,1))^2}{Ndays}}, \tag{6}$$

where $Ndays$ is the number of days covered by the data (i.e., the length of the `covid_data` matrix).

(a) Write a function named `sir_model.m` that takes as input `t` (time) and `X` (vector of S, I, R) and returns the right-hand side of the ODE system given by Eqs. (1)-(3).

(b) Write a script named `sir_main.m` that solves the ODE system for various values of β_0 (use a for loop for that). Use the following range of values for β_0: `beta0 = [0.1:0.001:0.5]./S0`, where `S0` is the initial susceptible population. Compute the error using Eq. (6) for each value of β_0 tested, and store these values. For the initial conditions, consider that the susceptible population of Santa Barbara County was 500,000, and assume that the infection started with 1 person infected and zero deaths. Use a time span of 100 days. Use the `ode45()` function to solve the ODE system.

(c) Still in your script `sir_main.m`, plot β_0 vs err. Make sure that:

- Your plots look like Fig. 2.
- Give your plot the following title: "Error analysis (SIR model vs data)".
- Change the line thickness to 3 for each line.
- Name the x and y axis as "beta0" and "Error", respectively.
- Turn the grid on using the command `grid on`.
- Increase the font size of legends, axis labels and title using the command `set(gca,'FontSize',20)`.

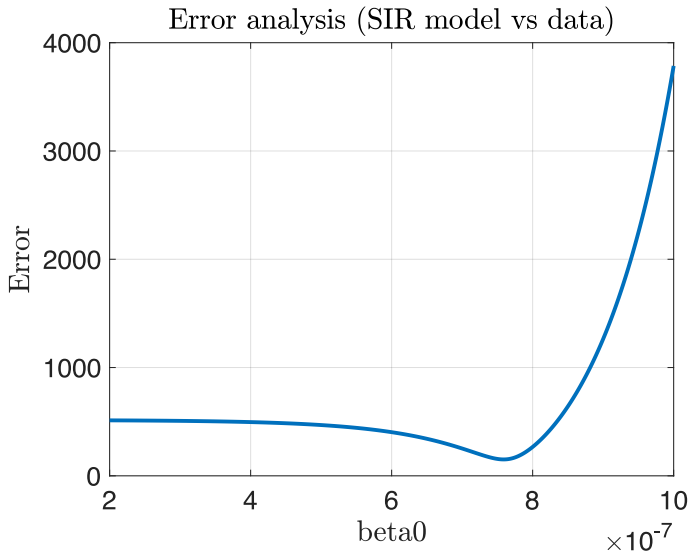

Figure 2: Evolution of the Euclidean norm of the error as a function of β_0.

(d) Still in your script `sir_main.m`, plot i) the total number of cases $C(t) = I(t) + R(t)$ vs. time, and ii) the number of deaths $R(t)$ vs. time, for the best value of β_0 found in part (c). In addition to that, overlap the curves for C and R with the `covid_data.mat` data provided on GauchoSpace, so that you can compare simulations with data. Make sure that:

- Your plots look like Fig. 3.
- Change the line thickness to 3 for each line.
- Instead of using the usual `plot()` command, use `semilogy()` to plot in semi-log scale. This will make your two curves look more distinguishable.
- Give your plot the following title: "SIR model, beta0 = X", where X must be replaced by the respective value of β_0 found in (c).
- Use legends "Cumulative infected (SIR)", "Deaths (SIR)", "Cumulative infected (data)", "Deaths (data)" in your plots.
- Name the x and y axis as "Time (days since March 15)" and "Number of individuals", respectively. Again, you can use `legend('location','best')` to move your legend box to the best available spot, so it doesn't cover your plots.
- Turn the grid on using the command `grid on`.

- Increase the font size of legends, axis labels and title using the command `set(gca,'FontSize',20)`.

Note: The best β_0 value obtained in (c) should be close to $\beta_0 \approx 7.580 \times 10^{-7}$. If you have not been able to accomplish the optimization, please proceed with $\beta_0 = 7.580 \times 10^{-7}$ for Part 3 of this project.

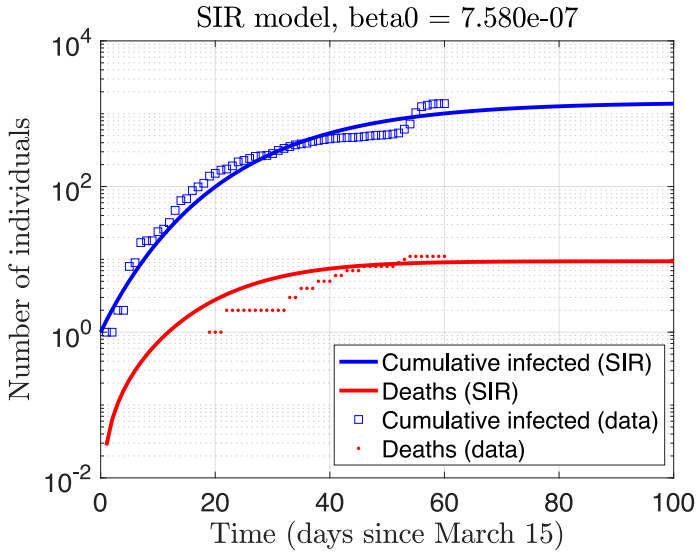

Figure 3: Semi-log plot of the solution of the SIR model with $\beta_0 = 7.580 \times 10^{-7}$.

Upload `sir_model.m` and `sir_main.m` to GauchoSpace.

Part 3: Stochastic simulation of SIR model (20 points)

Let's take a look back at your plots from Part 2-(c). At this point you probably noticed that we can find a β_0 that fits the data reasonably well, but no matter how much you change it, we will never find the perfect fit. One of the reasons for that involves the fact that traditional *deterministic* (ODE) models like the one given by Eqs. (1)-(3) cannot accurately predict the early stages of the infection, since they rely on the hypothesis of a well-mixed population and interactions of millions of individuals. In contrast, for small populations, random events are often relevant, and therefore situations like jumps in the number of cases are prone to occur; deterministic models, on the other hand, disregard these events. To solve these problems, we use *stochastic* models.

Notice that the current number of infected individuals in Santa Barbara County is relatively small compared to the total population. Thus, the deterministic model given by Eqs. (1)-(3) might lead to incorrect predictions. Your goal for this assignment is to solve a stochastic SIR model using a method called the *Gillespie algorithm*.

The Gillespie algorithm belongs to a class of methods called dynamic Monte Carlo methods. As you probably remember from homeworks 6 and 7, Monte Carlo simulations allow us to estimate probabilities of events by testing multiple trials (also called *realizations*). Since each realization involves random numbers, the results of Monte Carlo simulations are never exactly the same. Similarly, each realization (simulation) via the Gillespie algorithm will generate unique curves for $S(t), I(t), R(t)$.

The stochastic SIR model considers the following chains of events (also called *Markov chains*)

$$\mathrm{S} + \mathrm{I} \xrightarrow{\beta(\mathrm{t})} 2\,\mathrm{I}, \tag{7}$$

$$\mathrm{I} \xrightarrow{\gamma(\mathrm{t})} \mathrm{R}, \tag{8}$$

where $\gamma(t), \beta(t)$ are the same functions given by Eqs. (4)-(5). The chain given by Eq. (7) describes the process of infection: each time a susceptible individual meets an infected one, there's a propensity (which is proportional to $\beta(t)$) that dictates how often the susceptible individual becomes infected. Similarly, the chain in Eq. (8) describes the process of infected individuals succumbing to the virus and dying, with a propensity proportional to $\gamma(t)$. The propensity function α_i gives the probability that an event in the chain i will occur in the time interval $(t, t+\tau)$:

$$\alpha_i = \text{number of individuals interacting at chain "}i\text{"} \times \text{rate}, \tag{9}$$

which, for the chains given by Eq. (7)-(8), read

$$\alpha_1 = S(t)I(t)\beta(t), \tag{10}$$
$$\alpha_2 = I(t)\gamma(t). \tag{11}$$

For more details about the implementation of this algorithm, see p. 11 of this final project assignment: Recipe for Stochastic SIR using Gillespie Algorithm.

(a) Using the recipe provided on p. 11, write a function named `ssir_model.m`, that takes as input `tspan` (time span) and `X0 = [* ; * ; *]` (column vector of initial conditions for S, I, R) and returns one realization (i.e., column vectors containing $S(t), I(t), R(t)$, resulting from the algorithm provided in p. 9) of the Gillespie algorithm, and a time vector `tVec`. Use the functions $\gamma(t), \beta(t)$ given by Eqs. (4)-(5), and **use the value of β_0 that you found best fit the data in Part 2-(c), i.e.,** $\beta_0 = 7.580 \times 10^{-7}$.

(b) Write a script named `ssir_main.m` to compute `Nr = 5` realizations of the stochastic SIR model by calling your function `ssir_model` 5 times (use a for loop for that). For the initial conditions, assume that the susceptible population of Santa Barbara County was 500,000, and assume that the infection started with 1 person infected and zero deaths. Use a time span of 100 days.

(c) Still in your script `ssir_main.m`, plot i) the total number of cases $C(t) = I(t) + R(t)$ vs. time, and ii) the number of deceased individuals $R(t)$ vs. time, for each of the 5 realizations computed in Part 3-(b). In addition to that, overlap the curves for C and R with the `covid_data.mat` data provided on GauchoSpace, so that you can compare simulations with data. Make sure that:

- Your plots, legends and title look like Fig. 4. Note: since the method is stochastic, your curves will be different than the ones shown in Fig. 4.
- Instead of using the usual `plot()`, use the command `semilogy()` to plot in semi-log scale. This will make your two curves look more distinguishable.
- Give your plot the following title: "sSIR model".
- Use legends 'Cumulative infected (sSIR)', 'Deaths (sSIR)', 'Cumulative infected (data)', 'Deaths (data)' in your plots. Note: Since all the realizations have the same legends, use `'HandleVisibility','off'` in your `semilogy()` commands for realizations 2 to `Nr`, to avoid legend repetition.

- Name the x and y axis as "Time (days since March 15)" and "Number of individuals", respectively. Again, you can use `legend('location','best')` to move your legend box to the best available spot, so it doesn't cover your plots.
- Turn the grid on using the command `grid on`.

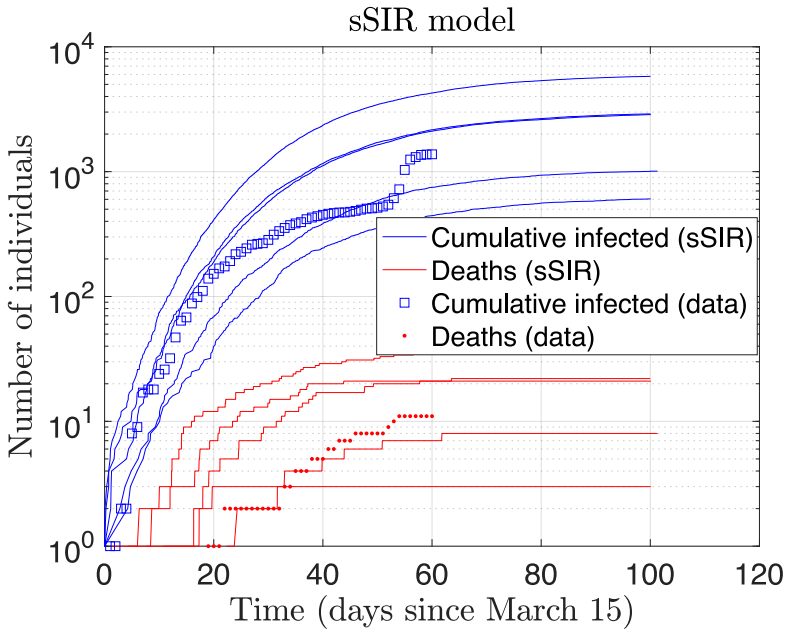

Figure 4: Semi-log plot of the solution of the stochastic SIR model for 5 realizations (each line corresponds to a realization).

(d) In a similar manner as you did for your script `ssir_main.m`, create another script, named `ssir_hist.m`, to compute $Nr = 2000$ realizations of the stochastic SIR model. At the end of each realization `i`, store the last value of the total number of cases, i.e., `C(i) = I(end) + R(end)`. This value represents a prediction of the number of cases in Santa Barbara County after `tspan` days (i.e., 100 days). With these values, create a histogram of the array C using the command `histogram(C)`. Your histogram should look like Fig. 5. Based on your histogram, write a comment in your script `ssir_hist.m` indicating, using your own words, what you think is reasonable upper bound for the number of cases.

Hint: for an empirical distribution, you can compute the *cumulative distribution function* using the command `cdfplot(C)`. The plot gives an estimate of the probability F that the number of cases C will be smaller than or equal to a value X.

Upload `ssir_model.m`, `ssir_main.m`, `ssir_hist.m` to GauchoSpace.

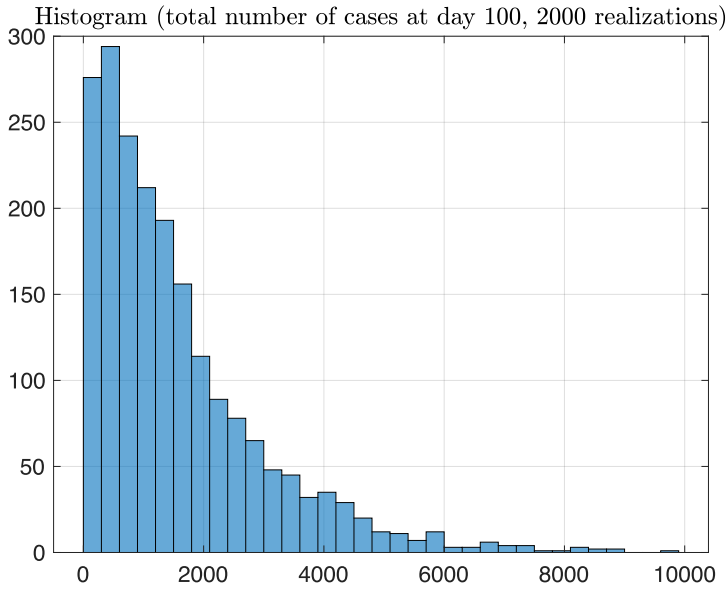

Figure 5: Histogram of the total number of cases $C = I + R$ at day 100, for 2000 realizations of the stochastic SIR model.

Recipe for Stochastic SIR using the Gillespie Algorithm:

While `t < tspan` , do the following:

1. Generate two random numbers, r_1, r_2, uniformly distributed in the interval $(0, 1)$.

2. Compute the propensity function α for each chain. Since we have two chains, the propensities are given by

$$\alpha_1 = S(t)I(t)\beta(t), \tag{12}$$
$$\alpha_2 = I(t)\gamma(t), \tag{13}$$

   where $\gamma(t), \beta(t)$ are the functions given by Eqs. (4)-(5). Use the value of β_0 that you found best fit the data in Part 2-(c).

3. Compute the total propensity $\alpha_0 = \alpha_1 + \alpha_2$.

4. Compute the time τ when the next event (i.e., either contamination or death) will take place.

$$\tau = \frac{1}{\alpha_0} \ln\left(\frac{1}{r_1}\right) \tag{14}$$

5. Compute the number of individuals for each group S, I, R at time $t + \tau$ as the following:

$$\text{if } \; r_2 \in [0, \alpha_1/\alpha_0) : \begin{cases} S(t+\tau) = S(t) - 1, \\ I(t+\tau) = I(t) + 1, \\ R(t+\tau) = R(t), \end{cases} \tag{15}$$

$$\text{else if } \; r_2 \in [\alpha_1/\alpha_0, 1) : \begin{cases} S(t+\tau) = S(t), \\ I(t+\tau) = I(t) - 1, \\ R(t+\tau) = R(t) + 1. \end{cases} \tag{16}$$

6. Make sure that the system always includes at least one infected person, i.e., check if $I(t + \tau) > 0$. If $I(t + \tau)$ becomes 0, include another infected person at time $t + \tau$ by setting $I(t + \tau) = 1$.

7. Save your new time as a point in the time vector: `tVec(i) = t + tau` (so you can plot $S(t), I(t), R(t)$ vs t later).