Macm 316 Computing Assignment #6

Submit on Crowdmark by Wednesday, August 5, 2020, 11pm

Upload one .pdf file with up to 4 pages: Pages 1-2 is your typed report (your discussions, data and figures on two pages); Pages 3-4 is a listing of your code(s). The assignment is due at 11:00pm. You will receive a Crowdmark link for the upload.

The deadline for this assignment is a Wednesday. I moved it to take into account the long weekend Aug 1–3. You are free to upload sooner!

A (simplified) Covid-19 model. The so-called SEIR model is a standard way to model the spread of infectious diseases. It is an example of a compartmental model. We will look at this model in its most basic form, and solve it numerically for various parameters.

The population of N individuals is assigned to different compartments, and individuals can move between compartments. The SEIR model uses four compartments, S, E, I, R, representing segments of the population, so $S+E+I+R=N$ (a simplification, meaning the total population is constant, with some leeway in interpreting the term "total population").

S. Susceptible individuals. Those are healthy, and not immune individuals who may become infected upon contact with an infected individual.

E. Exposed individuals. These are individuals who have been infected, but because of the incubation period of the virus are not yet infectious themselves. They will transition to the infected group;

I. Infected individuals. These are individuals who have been infected, and can pass on the infection to susceptible individuals.

R. Removed individuals. (In an optimistic scenario referred to as recovered individuals.) These include people who have recovered and are now immune, and people who have died.

Note that for Covid-19 we do not know yet whether recovered people are immune, and if so, for how long. This model assumes they become immune. Interactions between people in compartments are proportional to the number of people in each compartment – this gives a “quadratic” term. Transfers from one compartment to another that do not involve interactions are assumed to occur proportional to the number of people in one compartment, corresponding to linear terms.

The equations are as follows (the variable t is time, say measured in days):

$$\begin{aligned}
\frac{dS}{dt} &= -\beta S\frac{I}{N} && \text{susceptibles become exposed due to interaction with } I\text{, contact rate of } \beta\\
\frac{dE}{dt} &= +\beta S\frac{I}{N} - \alpha E && \text{incoming } S \text{ minus exposed moving to } I,\ \alpha = \text{ 1/incubation period}\\
\frac{dI}{dt} &= \alpha E - \gamma I && \text{incoming exposed - “removed”},\ \gamma = \text{ 1/infectious period}\\
\frac{dR}{dt} &= +\gamma I
\end{aligned}$$

That was the "easy" part. The difficulty, even within the constraint of this model, is to get somewhat realistic parameters – a major data collection problem in the real world. The other interesting part is, that some of these parameters will also be time dependent – the effect of physical distancing and lockdown or reopening. You probably have heard about the now infamous parameter R_0, the reproduction number. For our model we have $R_0 = \beta/\gamma$. (Given that we have "removed" individuals R, the R_0 naming convention is somewhat unfortunate.)

A reasonable set of parameters is

1. $\alpha = 1/5.2$, incubation period of roughly 5 days.
2. $\gamma = 1/10$, infectious period of 10 days. The actual period is probably longer, but this value takes into account that sick people go to hospital or stay home, rather than continuing to circulate among the general population).
3. $R_0 = 3.5$, for now assumed constant (which would be bad news).

Note that the parameter $\beta = R_0\gamma$.

When things become interesting is when we work with time-varying parameters, in particular $R_0 = R_0(t)$ (and, by association, $\beta = \beta(t)$). When you look at the model, β is the contact rate, which can be controlled. If we all stayed in a remote log cabin by ourselves, $\beta = R_0 = 0$; if we practice physical distancing and wear masks when physical distancing is not possible, then R_0 might stay below 1; if we hang out in bars or at large gatherings, then β and R_0 will go through the roof – leading to large numbers of Is, infected and infectious people.

We could model the effect of intervention with data for R_0 as follows (three profiles are given):

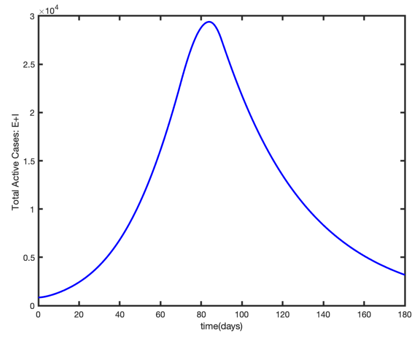
Days (since outbreak)	1..20	21..70	71..84	85..90	91..110	111..1000	after 1000
R_0	3.5	2.6	1.9	1.0	0.55	0.55	0.5
R_0	3	2.2	0.7	0.8	1.00	0.90	0.5
R_0	3	2.2	0.9	2.5	3.20	0.85	0.5

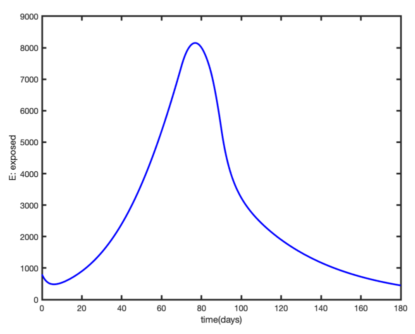
What is missing? We need N, the number of people. For British Columbia we use $N = 5$ Million. We also need initial conditions with $I(0)$ and/or $E(0)$ non-zero. So, "to get started" let us take

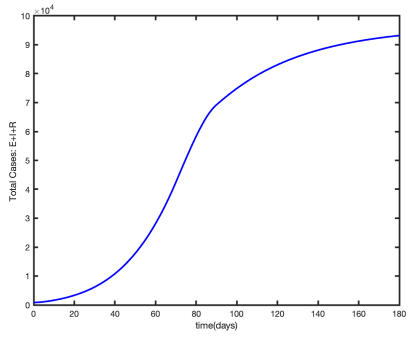
$$I(0) = 40; \qquad E(0) = 20\,I(0); \qquad R(0) = 0, \qquad S(0) = N - I(0) - E(0) - R(0).$$

Once again, $R(0)$ is the initial value of $R(t)$, and is nor related to the reproductive number R_0.

Last point: we run the simulation for 6 months, so take $T_{final} = 180$. Sample output:







Now you have all the information to become a member of the modeling team for the infectious disease epidemic.

Your tasks

1. Use one of Matlab's codes, for example, `ode45` or `ode15s`, to solve the SEIR equations numerically. This will require you to write a function that evaluates the right hand side of the differential equations.
2. Run your codes with $N = 5$ Million people (BC scenario), and $\alpha = 1/5.2$ and $\gamma = 1/10$.
3. At first, use constant R_0 and run your code with different values of R_0: $3.5, 2.5, 1.25, 0.9$. For the constant R_0 case summarize your observations, but only report the results corresponding to $R_0 = 3.5$. $R_0 = 3.5$ is your Scenario 1.
4. Your Scenarios 2, 3, and 4: Run your code with the time dependent R_0 values given in the table above. You may simply use the values as a piecewise constant function, or interpolate the values using Matlab's `interp1` function. The first row corresponds to a scenario with some lock-down measures; the second row shows an effective response; the third row shows the consequences of allowing a lapse in prevention measures. You may want to plot your R_0 values agains time.
5. Assuming a 4% death rate among the $R(t)$ population, compare the number of deaths per 1 Million people for each of your four scenarios. A table will do!
6. Show your plots of number of active cases and total number of cases for your four scenarios.
7. BC has a total of about 5600 acute care beds, and 200 ICU (intensive care unit) beds. Assume that 3500 of acute care beds, and 160 ICU beds are available for Covid cases, and 8% of infected people need to be hospitalized, and 1% of infected people must be admitted to the ICU. Compare the four scenarios above in terms of hospital capacity: Plot the number of Covid patients in hospital and in ICUs, and compare to the capacity,

Some programming notes – just some ideas, you are allowed to do things differently.

1. Rewrite in vector form. To get the system into a format for Matlab to process, we write

$$\mathbf{y}(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ y_4(t) \end{pmatrix} = \begin{pmatrix} S(t) \\ E(t) \\ I(t) \\ R(t) \end{pmatrix}, \qquad \frac{d\mathbf{y}}{dt} = \begin{pmatrix} -\beta y_1(t)\dfrac{y_3(t)}{N} \\ +\beta y_1(t)\dfrac{y_3(t)}{N} - \alpha y_2(t) \\ \alpha y_2(t) - \gamma y_3(t) \\ +\gamma y_3(t) \end{pmatrix}$$

2. Assume your function describing the equation is

```
function  yprime = seir(t,y)
```

The easiest way to get the parameters (α, γ and R_0) into the function is to "hard code" them. You then have to change your function every time you run your code with different parameters.
Alternatively, you can pass parameters in a variable called `par`.

```
%Main program
par.alpha = 1/5.2;
par.gamma = 1/10;
par.rzero = 3.5;
par.N = 5.0e6;
function  yprime = seir(t,y,par)
```

You can then refer to those values inside your function `seir.m`, as they are being passed along. To call, for example, `ode45` you would write

```
rtol = 1.e-6; atol=1.e-5;
options = odeset('AbsTol', atol,'RelTol',rtol,'MaxOrder',5);
[t,y] = ode45(@(t, y) seir(t,y, par) , [0,180], yinit, options)
```

`atol` and `rtol` are error tolerances for solving the differential equations.
`yinit` is the vector of initial values $[S(0), E(0), I(0), R(0)]^T$. If you want Matlab to give you more information, you could use the following options:

```
options = odeset('AbsTol', atol,'RelTol', rtol,'MaxOrder',5,'Stats','on');
```

3. A remaining issue is how to implement a time-varying R_0 factor. One possibility is to pass an array of length at least 180, that has an entry for each day. For the first time-varying data set:

```
maxd=200; rtzero=zeros(1,maxd);
rtable = [ 1 3.5; 21 2.6; 71 1.9; 85 1.0; 91 0.55; 1001 0.5];
for j=1:5,
  constdays = min(rtable(j+1,1),maxd+1) - rtable(j,1);
  rtzero( rtable(j,1):rtable(j,1)+constdays-1 ) = rtable(j,2)*ones(1,constdays);
end;
par.rtzero = rtzero;
```

4. You can test your code by setting $\gamma = 0$. Then you are essentially just solving an exponential decay equation.