Exam: Epidemic modelling

Candidate nr. 10037
Department of Physics, Norwegian University of Science and Technology, Trondheim
Norway
TFY4235 - Computational physics
(*Last updated on May 5, 2021*)

Abstract

Short abstract

Contents

PART I

Introduction

The topic of this project is epidemic modelling, and we will study various different models. A short introduction to these are given in this introduction, mostly for introducing notation and presenting some choices in the implementation. For a more detailed introduction, consult the problem sheet [**?**].

0.1 The deterministic SIR model

The SIR model reads

$$\frac{\mathrm{d}S}{\mathrm{d}t} = -\beta\frac{IS}{N} \tag{1a}$$
$$\frac{\mathrm{d}I}{\mathrm{d}t} = \beta\frac{IS}{N} - I/\tau \tag{1b}$$
$$\frac{\mathrm{d}R}{\mathrm{d}t} = I/\tau, \tag{1c}$$

where β is the rate at which susceptible people encounter infected people, and become infected, and τ is the typical duration of the infection. To solve this equation numerically, we recast it as a system of ODE's by defining the vector $\mathbf{v} = [S, I, R]^T$, such that the equations in 1 simplify to

$$\frac{\mathrm{d}\mathbf{v}(t)}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} S(t) \\ I(t) \\ R(t) \end{bmatrix} = \begin{bmatrix} -\beta IS/N \\ \beta IS/N - I/\tau \\ I/\tau \end{bmatrix} = \begin{bmatrix} -\beta v_1 v_2/N \\ \beta v_1 v_2 - v_2/\tau \\ v_1/\tau \end{bmatrix} = \mathbf{f}(\mathbf{v}, t). \tag{2}$$

0.2 The stochastic SIR model

For the stochastic SIR model we have probabilities instead of rates. The probabilities for a susceptible person becoming infected, and an infected person recovering during an interval of Δt are given by

$$P_{S\to I} = 1 - \exp\left(-\Delta t\beta I/N\right) \tag{3a}$$
$$P_{S\to R} = 1 - \exp\left(-\Delta t/\tau\right), \tag{3b}$$

respectively. The transition to next time step is then governed by

$$S(t + \Delta t) = S(t) - \Delta_{S\to I} \tag{4a}$$
$$I(t + \Delta t) = I(t) + \Delta_{S\to I} - \Delta_{I\to R} \tag{4b}$$
$$R(t + \Delta t) = R(t) + \Delta_{I\to R}, \tag{4c}$$
$$\tag{4d}$$

where the Δ's are drawn from binomial distributions:

$$\Delta_{S\to I} = \mathcal{B}(S, P_{S\to I})$$
$$\Delta_{I\to R} = \mathcal{B}(S, P_{I\to R}).$$

0.3 The stochastic SEIIaR model

For the stochastic SEIIaR model we follow essentially the same line of thought as for the stochastic SIR model, except that we include E : exposed people, that cannot infect others, and I_a : infected but asymptomatic people. The probabilities for the different transitions happening during an interval Δt are

$$P_{S\to E} = 1 - \exp\left(-\Delta t\beta\frac{r_s I + r_a I_a}{N}\right) \tag{6a}$$
$$P_{E\to I} = f_s \times (1 - \exp(-\Delta t/\tau_E)) \tag{6b}$$
$$P_{E\to I_a} = f_a \times (1 - \exp(-\Delta t/\tau_E)) \tag{6c}$$
$$P_{I\to R} = 1 - \exp(-\Delta t/\tau_I) \tag{6d}$$
$$P_{I_a\to R} = 1 - \exp(-\Delta t/\tau_I), \tag{6e}$$

with the different parameters $r_a, r_s, f_a, f_s, \tau_E, \tau_I$ given in [**?**]. As before,

$$\Delta_{S\to E} = \mathcal{B}(S, P_{S\to E})$$
$$\Delta_{I\to R} = \mathcal{B}(I, P_{I\to R})$$
$$\Delta_{I_a\to R} = \mathcal{B}(I_a, P_{I_a\to R}).$$

and

$$\Delta_{E\to I}, \Delta_{E\to I_a}, \Delta_{E\to_E} = \mathcal{M}(E, (P_{E\to I}, P_{E\to I_a}, (1 - P_{E\to I} - P_{E\to I_a}))), \tag{8}$$

where $\mathcal{M}$ denotes a multinomial distribution, as described in [**?**]. The changes of the different variables during an interval Δt are then given by

$$S(t+\Delta t) = S(t) - \Delta_{S\to E} \tag{9a}$$
$$E(t+\Delta t) = E(t) + \Delta_{S\to E} - \Delta_{E\to I} - \Delta_{E\to I_a} \tag{9b}$$
$$I(t+\Delta t) = I(t) + \Delta_{E\to I} - \Delta_{I\to R} \tag{9c}$$
$$I_a(t+\Delta t) = I_a(t) + \Delta_{E\to I_a} - \Delta_{I_a\to R} \tag{9d}$$
$$R(t+\Delta t) = R(t) + \Delta_{I_a\to R} + \Delta_{I\to R}. \tag{9e}$$
$$\tag{9f}$$

PART
II

Code overview

The source code for this project is structured into separate files for each sub-problem, i.e. `prob_2A.py`, `prob_2B.py` etc. Additionally, there is a file for plotting for each sub-problem. I have also included a file called `utils.py` which contains some useful utility functions used throughout the exercises. For problem 2A I have used the same ODE-solver as the one I made for exercise 2. This can be found in `ode.py`.

PART III

Results and Discussion

1 Problem 2A: Deterministic SIR model

1.1 a)

The ODE-solver I chose to use is the one I made for exercise 2, as I am very familiar with this, and found it working reasonably good for these kind of purposes. For this problem I use the RK4-method. The implementation is an object-oriented version of the solver methods shown in the lectures and should be easily understood by the documentation in `ode.py`.

I solve the deterministic SIR equations in terms of fractions of people. As seen in figure 1 the asymptotic expressions for $S(t)$ and $R(t)$ as given in the exam-sheet [**?**]. The expressions for $S(\infty)$ and $R(\infty)$ are solved using the non-linear equation solver `fsolve` from `scipy`, with initial guesses 0.5 for both.

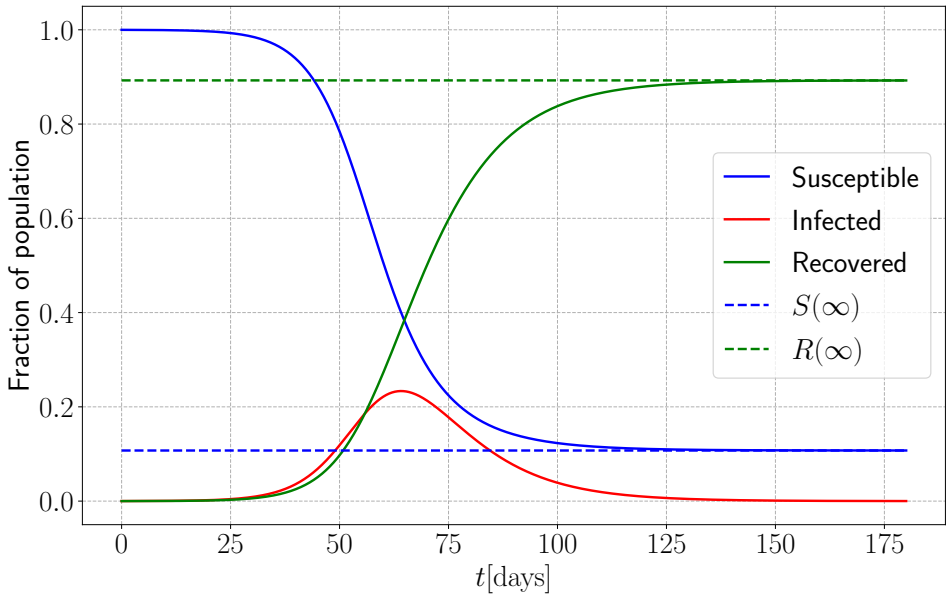

Figure 1: SIR equations with $\beta = 0.25\,\mathrm{day}^{-1}$, $\tau = 10\,\mathrm{day}$.

1.2 b)

For the early developments of the epidemic, the expression for I can be simplified to [**?**]

$$\frac{\mathrm{d}I}{\mathrm{d}t} = (\beta - 1/\tau)I. \tag{10}$$

The solution of this equation is given by

$$I(t) = I(0)\exp\left([\beta\tau - 1]\,\frac{t}{\tau}\right) = I(0)\exp\left([\mathcal{R}_0 - 1]\,\frac{t}{\tau}\right),$$

where we have introduced $\mathcal{R}_0 = \beta\tau$. As seen in figure 2, the fraction of infected people matches this expression quite closely during the first 40 days or so. This confirms the exponential growth in the beginning.

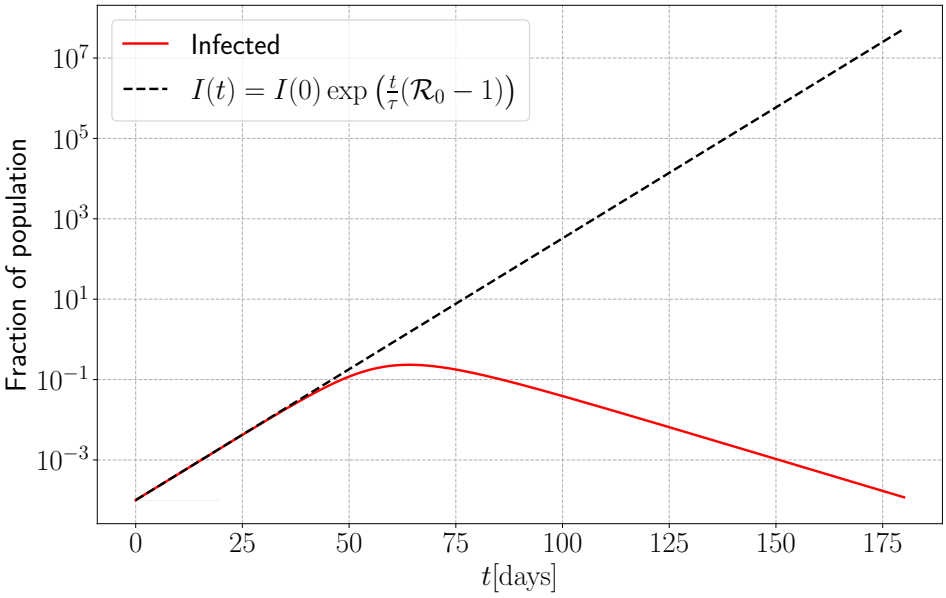

Figure 2: Infected people compared with the analytical approximation at the early stages.

1.3 c) Flattening the curve.

As seen from figure 1, the peak of the infected people is rather close to being at 0.2 when $\beta = 0.25$. Remark also that the peak occurs in the early stage of the epidemic. We use these observations to our favour when finding the *largest* β that will keep I/N[1] smaller than 0.2 during the pandemic. To do this, I choose the following procedure:

Choose a starting value for $\beta = \beta_0$;
Choose a tolerance `tol`;
Run the simulation of the SIR with the same parameters as in 2Aa, except with $\beta = \beta_0$.;
Calculate $\mathcal{I} := \max_{t\in[0,180]} I(t)$.;
while $err := |\mathcal{I} - 0.2| > tol$ **do**
  if $\mathcal{I} > 0.2$ **then**
    $\beta \leftarrow \beta \cdot 2^{\mathtt{err}}$
  else
    $\beta \leftarrow \beta \cdot 2^{-\mathtt{err}}$
  end
end
Run the simulation of the SIR with the same parameters as in 2Aa, except with the new value for β.;
Recalculate $\mathcal{I} := \max_{t\in[0,180]} I(t)$.;

Algorithm 1: Finding the largest beta keeping $\max I$ *less* than 0.2.

[1]Note that I use fractions of people instead of actual numbers of people in this section, so that is why I refer to I as if it is I/N in the procedure below.

This procedure is by no means any advanced piece of algorithm, but it ensures that β is increased when the maximum of I is less than 0.2, and decreased when it is larger than 0.2. Further, it also ensures that the "nudging" of beta in each step is less when the error is less. To get an estimate for the *largest* β one can have to keep I/N *less* than 0.2, one should start off with an initial value β_0 for which the peak is less than 0.2, so that one reaches the limit from below. Using $\beta_0 = 0.2$, one finds with the maximum value of $\beta = 0.28020370$, as given in table 1 together with the deviation $0.2 - \max_{t\in[0,180]} I(t)$ this value gives rise to.

1.4 d) Vaccinations

To find the minimum fraction of vaccinated people preventing an outbreak, $R(0)/N$, I use the following procedure:

Choose a starting value for $R(0) = R_0$;
Run the simulation of the SIR with the same parameters as in 2Aa, except with $R(0) = R_0$;
Calculate the `slope` of the fraction of infected people in a semi-log plot for the first 15[a] days of the simulation;
If the initial slope is negative it indicates that the outbreak dies out by itself;
while *slope*> 0 **do**
  $R(0) \leftarrow R \cdot 2^{\texttt{slope}}$;
  Run the simulation of the SIR with the same parameters as in 2Aa, except with the new value for $R(0)$.;
  Recalculate the `slope` in the semi-log axes.;
end

Algorithm 2: Finding the minimum fraction of initially vaccinated people for outbreaks to be impossible.

[a] As people are typically sick for 10 days ($\tau = 10$) I regard this time scale as long enough to detect whether the outbreak grows exponentially or decays.

The value of $R(0)/N$ preventing an outbreak found from this procedure is 0.59987499, as shown in table 1 together with the slope in the semi-log plot which this fraction yields for $I(t)$. This is obtained from running the procedure with $R_0 = 0.1$.

Table 1: The maximum value of β giving a peak less than 0.2 of the infected fraction, and the minimum value of $R(0)$ (vaccinated) avoiding exponential growth.

Parameter	value	$0.2 - \max_{t\in[0,180]} I(t)$	Initial log-slope
β	0.28020370	$8.319 \cdot 10^{-7}$	—
$R(0)$	0.59987499	—	$-1.74 \cdot 10^{-15}$

2 Problem 2B: Stochastic SIR model

2.1 a)

As seen in the plot in figure 3, the 10 different realisations of the stochastic SIR model , shown in dashed lines in graded colours, seem to lie close to the deterministic solution.

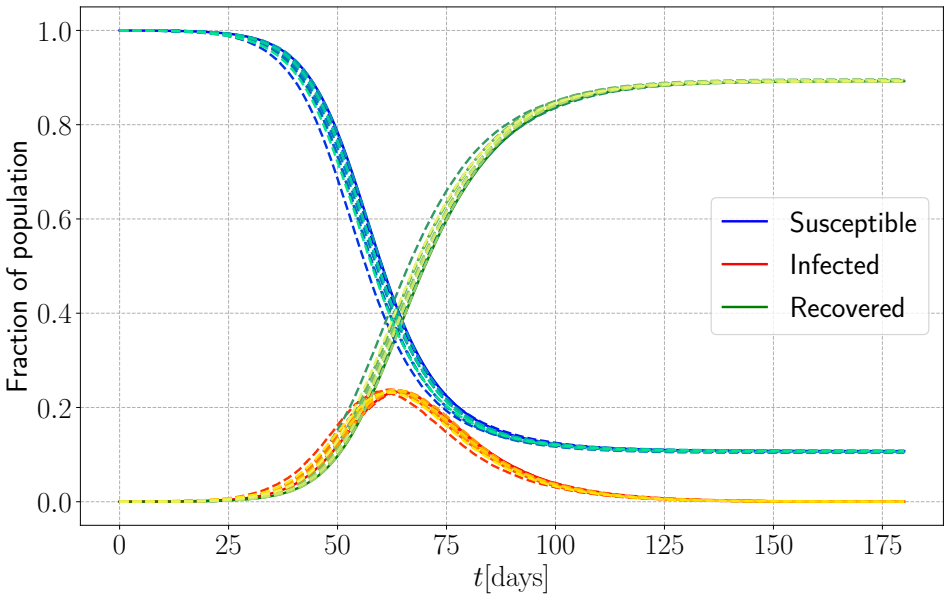

Figure 3: Solution of stochastic SIR equations with $\beta = 0.25\,\mathrm{day}^{-1}$, $\tau = 10\,\mathrm{day}$.

To choose a suitable time-step for simulating this model, I run the simulation up to $t = 250$ days and find the deviation between the fraction of susceptible and the semi-analytic expression given in the problem sheet [**?**], and similarly for the fraction of recovered. I compute this using the average of 100 runs of the stochastic model. I include the same run with the deterministic model to see how the deviations relate to each other. This is shown in figure 4. Clearly, using this comparison is *not* a good way of determining the time step we want to use for the deterministic model, as the solution approaches the asymptotic values for all of the chosen step lengths. However, we see from this plot that if we choose a step length $\lesssim 0.01$, the deviations of the stochastic solutions start to flatten out, and are roughly of the same order as those of the deterministic model. I therefore always use a time step smaller than 0.01 for simulating the stochastic models. In particular, I use $\Delta t = 0.005$ in problem B,C,D and Ea. For Eb I use $\Delta t = 0.01$ because the runtime become uncomfortably large when using smaller time steps.

For determining the step length to use for the deterministic model I create a reference solution using a relatively small time step, $\Delta t = 10^{-4}$, and then use the deviations between this and each run at the endpoint as a measure of the absolute error. After having done that, it is mostly a matter of preference how small a step length one wants to use. If I can accept an error of 10^{-8} in the fraction of susceptible after 50 days — corresponding to 1/1000 of an individual — I see from the plot in figure 5 that I will have to use $\Delta t \lesssim 10^{-1}$. Since the runtimes of the simulation for $\Delta t < 10^{-2}$ for 50 days are always less than 1 second, I find that $\Delta t = 0.005$ is probably more than sufficient, and appropriate considering both runtime and accuracy.

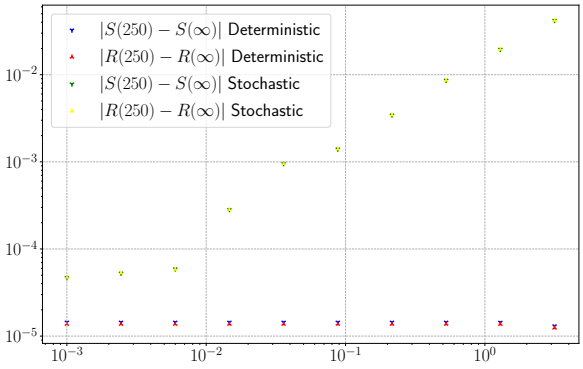

Figure 4: Global error for stochastic model, compared with $S(\infty)$ and $R(\infty)$.

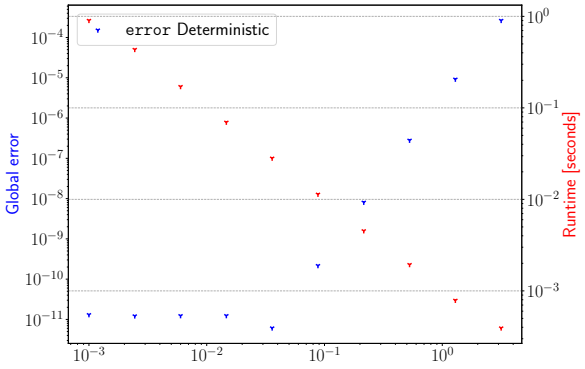

Figure 5: Global error the deterministic model.

2.2 b)

As in the previous exercise, we plot the fraction of infected people together with the analytical model for the early stages. This is shown in figure 6. Here we clearly see that all the realisations are approximately linear in the semi-log plot in the first 40 days, or so, of the simulation as we observed in the previous exercise.

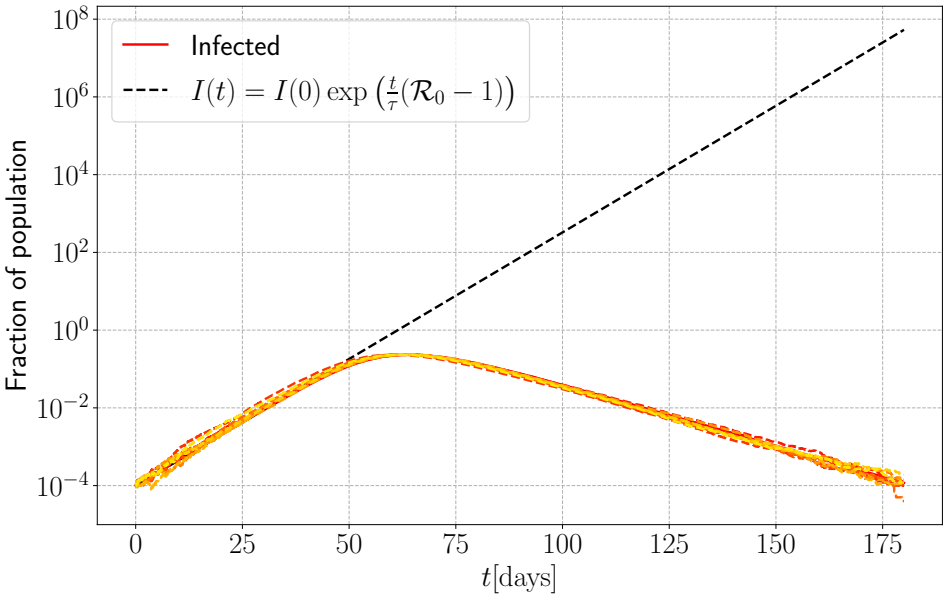

Figure 6: Infected people compared with the analytical approximation at the early stages. Stochastic and deterministic model.

2.3 c) Probability of an outbreak

There will always be a certain probability for an outbreak disappearing by itself using the stochastic model. In the present subsection we estimate this probability for an initial number of infected people $= 1, 2, \ldots, 10$. This is done by the procedure described in algorithm 3. In this procedure, I use the slope of the number of infected people $I(t)$ during the first 30 days as an indicator of whether the outbreak disappears by itself or not. If the slope is positive, I count it as an outbreak, and if not, I count it as disappearing.

Choose the parameters of the model as those given in the exam sheet [**?**], but with $T = 30\,\text{days}$[a]. ;
Choose a batch-size B.;
for $I = 1, 2, \ldots, 10$ **do**
  Initialise an empty vector of length B: $\mathbf{X} = [0, \ldots, 0]$.;
  for $n = 1, \ldots, B$ **do**
    Run the simulation with initial number of infected people = I.;
    Calculate the `slope` in the semi-log axes for $I(t)$.;
    if *slope* <= 0 **then**
      $X_n = 0$
    else
      $X_n = 1$
    end
  end
  Estimate the probability of an outbreak for I initially infected by

$$p := P(\text{outbreak}|I) = \frac{1}{B}\sum_{n=1}^{B} X_n.$$

  Calculate the standard deviation of the estimate by

$$\sqrt{\text{Var}(\hat{p})} = \sqrt{\frac{p(1-p)}{B}}. \tag{11}$$

end

Algorithm 3: Calculating the probability of an outbreak as a function of the initial number of infected people, I.

[a] As the typical infection time is 10 days, I assume 50 days to be sufficient for detecting an outbreak in the stochastic model.

In performing this procedure, I use a batch size of 500 in each sweep. What we are estimating here is essentially a Bernoulli-distributed random variable, X: X can take the realisations 1 or 0 with probabilities p and $1 - p$ respectively, and we assume them to be independent [**?**, p.26]. Then, as the variance of such a distribution is $p(1 - p)$, the variance of the estimator for the probability, namely $\hat{p}$[2], is

$$\text{Var}(\hat{p}) = \sum_{n=1}^{B} \text{Var}\left(\frac{X_n}{B}\right) = \frac{1}{B^2}\sum_{n=1}^{B} \text{Var}(X_n) = \frac{1}{B^2}\sum_{n=1}^{B} p(1-p) = \frac{p(1-p)}{B},$$

from which formula (11) follows.

The probability of an outbreak as a function of the initial number of infected people are shown in figure 7 together with the associated standard deviation. As seen from this plot, when there are more than 6 people initially infected, the probability is approximately 1 that an outbreak will happen. However, for e.g. 1 initially infected person, the probability is around 0.65. This ultimately shows that the stochastic model has a more realistic feature to it than the deterministic one, in that these scenarios might occur.

[2] That is, the mean value of X_n.

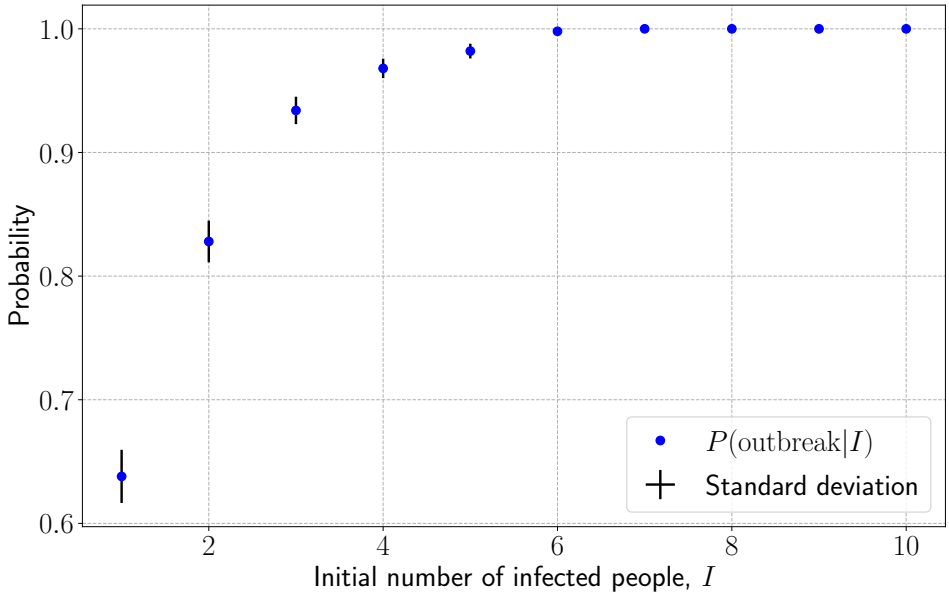

Figure 7: Probability of an outbreak as a function of initial number of infected people.

3 Problem 2C: Stochastic SEIIaR model

3.1 a)

The time-development of all the variables in the SEIIaR model is shown in figure 8, showing 10 realisations of the simulation. To compare it easily with the deterministic model, we contract E, I, I_a into one variable, so that they together represent all the infected people. The solutions of the deterministic model are shown together with the contracted versions of these simulations in figure 9.

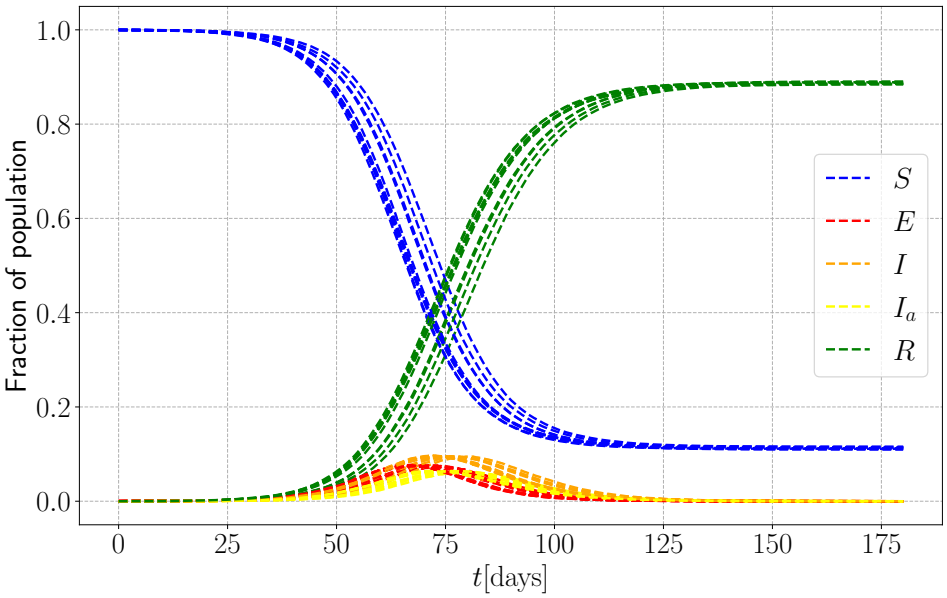

Figure 8: Solution of the stochastic SEIIaR-equations.

One prominent feature to notice here, is that the process overall seems to be *slower*, in the sense that it takes longer for people to get infected and eventually recover, but the peaks and asymptotic behaviours seem to align quite closely. This is probably due to the fact that this model includes a period in which people are infected but not yet able to infect anybody else: an incubation period of typical length $\tau_E = 3\,\text{days}$. This will naturally delay the process, but it should not affect the peak nor the asymptotic behaviour. Ultimately, this shows that the SEIIaR model is more realistic in the sense that people do not get sick right away, and that the behaviour agrees with our expectations. Another thing to mention is that the fraction of infected asymptomatic people I_a are always less than the fraction of infected symptomatic people I. This is as expected since $f_a < f_s$ and $r_a < r_s$.

To test that the implementation is correct — by comparing with the deterministic and stochastic SIR model — we adjust the parameters of the SEIIaR to $\beta = 0.25$, $r_s = 1$, $r_a = 1$, $\tau_E = 0$, $\tau_I = 10$ and start with 10 out of 100 000 people initially infected. The results of doing 1000 such simulations and performing the average of the stochastic models is shown in figure 10. This shows that the two stochastic models are more or less identical, as expected.

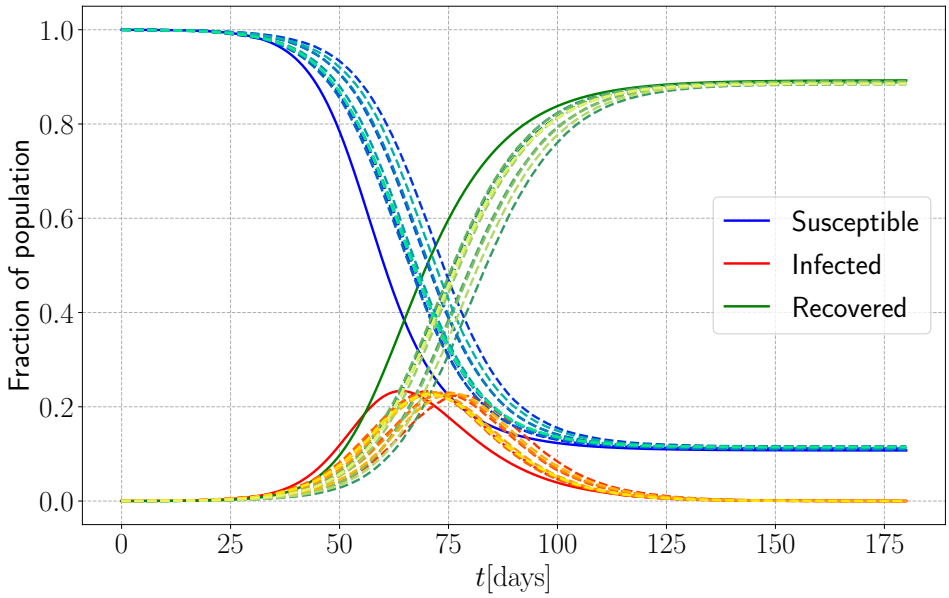

Figure 9: Comparison of the solution of the Stochastic SEIIaR-equations with the deterministic SIR-model. The number of infected people I in the stochastic model is $E+I+I_a$.

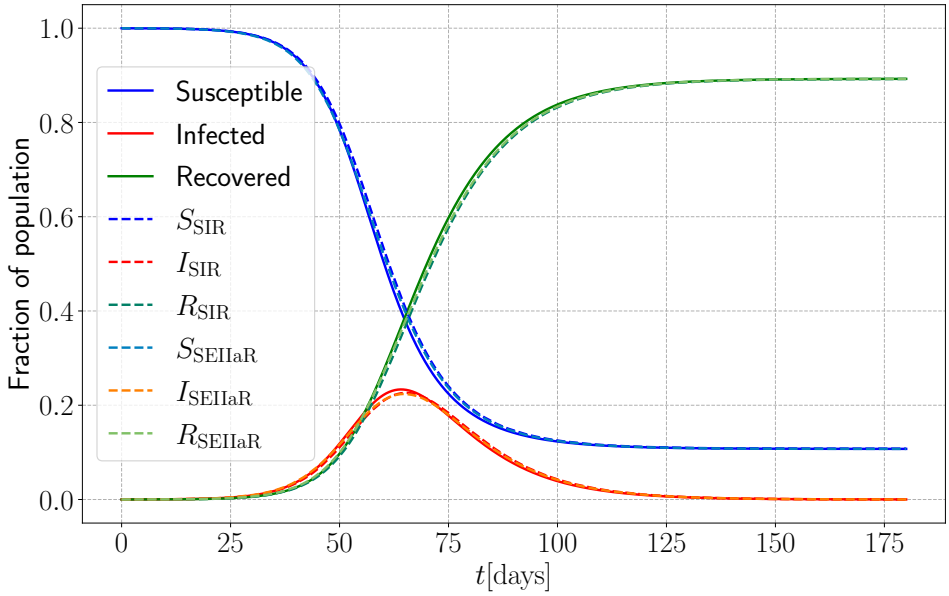

Figure 10: Solution of the stochastic SEIIaR-equations compared with the stochastic and deterministic SIR-equations for the case of identical parameters.

3.2 b) Probability of outbreak dependence on r_s

The variable r_s describes how infectious a person is when he is in the infection state. Reducing this constant below 1 can therefore correspond to increasing the degree of self-isolation when people are symptomatic. We investigate the probability of an outbreak as a function of this self-isolation-rate r_s by the following procedure (which is essentially the same as that shown in $2Bb$ expect we are now finding the probability as a function of r_s):

Choose the parameters of the model as those given in the exam sheet [**?**], but with $T = 30\,\text{days}$[a]. ;
Choose a batch-size B.;
Choose a number of values n of r_s to try;
Select n values of r_s, equally spaced between 0.001 and 1 :

$$\mathbf{R} = [R_1, \dots R_n]$$

for $i = 1, 2, \dots, n$ **do**
  Initialise an empty vector of length B: $\mathbf{X} = [0, \dots, 0]$.;
  for $n = 1, \dots, B$ **do**
    Run the simulation with $r_s = R_i$.;
    Calculate the `slope` in the semi-log axes for $I(t)$.;
    if *slope* <= 0 **then**
      $X_n = 0$
    else
      $X_n = 1$
    end
  end
  Estimate the probability of an outbreak for I initially infected by

$$p \coloneqq P(\text{outbreak}|r_s) = \frac{1}{B}\sum_{n=1}^{B} X_n.$$

  Calculate the standard deviation of the estimate by

$$\sqrt{\text{Var}(\hat{p})} = \sqrt{\frac{p(1-p)}{B}}.$$

end

Algorithm 4: Calculating the probability of an outbreak as a function of r_s.

[a] As the typical infection time is still 10 days, I assume 50 days to be more than sufficient for detecting an outbreak in the stochastic model.

The results of this calculation using a batch size of $B = 500$ and 100 values of r_s are shown in figure 11. The behaviour is as expected: when people are hardly infectious when symptomatic the probability of an outbreak is close to 0. This is also probably also a result of the fact that $r_a = 0.1$, i.e. it is not particularly likely to infect anyone when you are asymptomatic. If r_a and f_a was higher, one would expect the probability distribution to stagnate at some finite value when $r_s \to 0$, or at least go to 0 slower. This is demonstrated in the same figure, where we set $r_a = 1$ and perform the same test as above. This again is an intuitive confirmation that the model behaves as expected, and therefore is correctly implemented.

We see from this plot that it an outbreak is almost certain to *not* grow exponentially when $r_s \lesssim 0.2$, and there is roughly a 40 % chance of the outbreak not growing exponentially when $r_s = 0.4$.

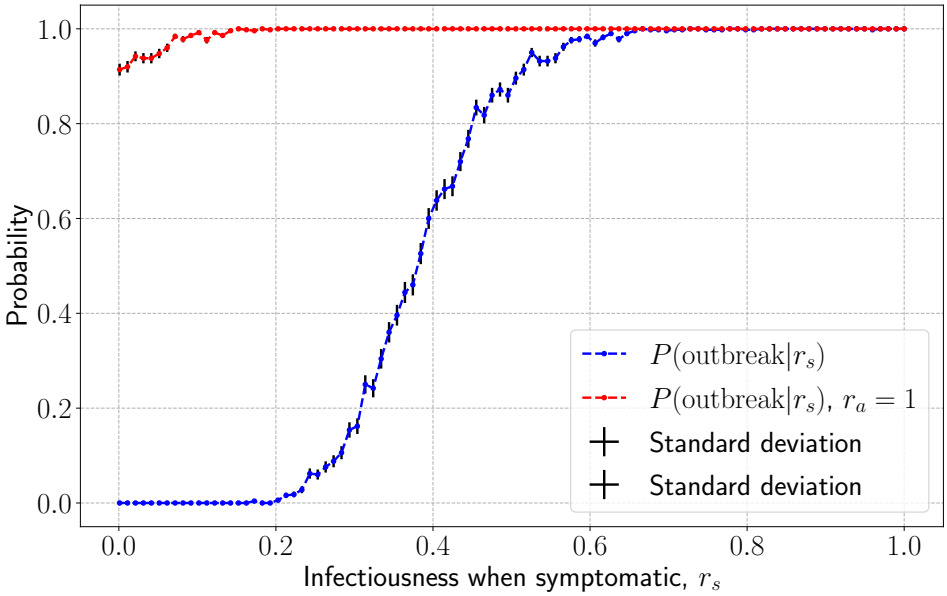

Figure 11: Probability of an outbreak as a function of r_s.

4 Problem 2D: Stochastic SEIIaR Commuter model

4.1 a) Commuter model for a two-town system

I set up a population structure as described by the matrix

$$\mathbf{M} = \begin{bmatrix} 9000 & 1000 \\ 200 & 99800 \end{bmatrix},$$

and simulate the time evolution of each of the five states for 180 days, as described in [?], with 25 initially exposed people working and living in town 1. The results of this simulation is shown in figure 12. Notice that we here use the number of people as scale on the vertical axis, so that we easier can distinguish the two towns from each other. We observe that the evolution of the epidemic in town 2 is delayed compared to that of town 1. This is as expected, as only 25 people in town 1 are exposed, and they can only infect people in town 2 *via* someone that works in town 2 during the night, or someone that works in town 1 while living in town 2 during daytime.

Also here, note that at each time there are fewer asymptomatic infected than symptomatic infected. This is as expected since $f_a < f_s$, and $r_a < r_s$.

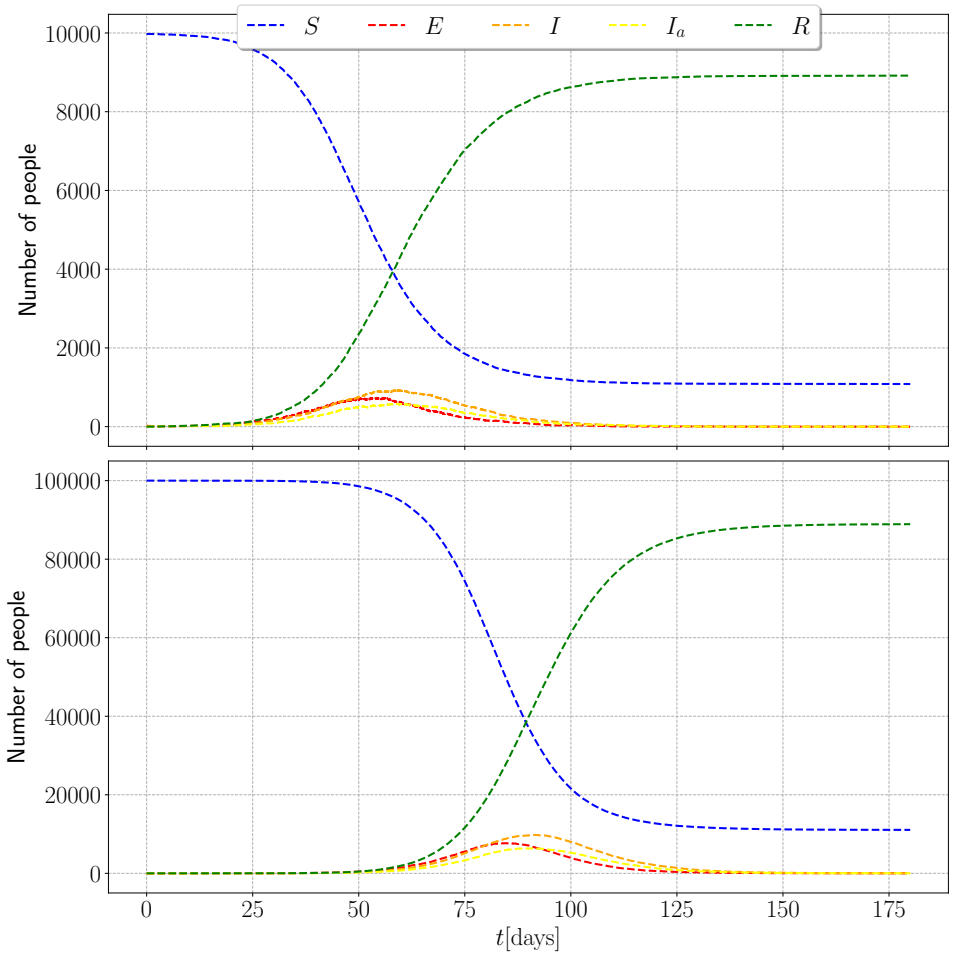

Figure 12: Solutions of Stochastic SEIIaR commuter model for the 2-city case.

4.2 b) Description of implementation & tests

To implement the Stochastic SEIIaR Commuter model, I choose the following procedure:

Choose a population structure $\mathbf{M} \in \mathbb{N}^m \times \mathbb{N}^m$, i.e. an $m \times m$ matrix;
Set an end time t_N and a time-step Δt;
Set an initial state $\mathbf{X}_0 \in \mathbb{N}^m \times \mathbb{N}^m \times \mathbb{N}^5$, where entry $X_{0,ij}$ is the vector $\mathbf{v}$ of the variables S, E, I, I_a, R for group (i, j) in the matrix $\mathbf{M}$.;
Create an empty array for holding the state at each point in time, $\mathbf{X}$, and set its first element to $\mathbf{X}_0$.;
Calculate the probabilities:

$$\begin{aligned} P_{E\to I} &= f_s \times (1 - \exp(-\Delta t/\tau_E)) \\ P_{E\to I_a} &= f_a \times (1 - \exp(-\Delta t/\tau_E)) \\ P_{I\to R} = P_{I_a\to R} &= 1 - \exp(-\Delta t/\tau_I) \end{aligned}$$

Calculate the number of days $D = \texttt{int}(t_N)$, and the number of steps per half day $S = \texttt{int}(1/(2\Delta t))$.;
for $d = 0, \ldots, D-1$ **do**
  $i \leftarrow 2 \times d \times S$;
  for $j = 0, \ldots, S-1$ **do**
    Calculate the number of people in each town, β,

$$N_\beta = \sum_{\alpha=1}^{m} M_{\alpha\beta}$$

    Find the number of infected people of each kind at the previous time step for each town β,

$$I_\beta, I_{a,\beta} = \sum_{\alpha=1}^{m} X_{(i+j)\alpha\beta 3}, \sum_{\alpha=1}^{m} X_{(i+j)\alpha\beta 4}$$

    Calculate the probability for transitioning between S and E for each town, β, [a]

$$P_{S\to E,\beta} = 1 - \exp\left(-\Delta t \beta_0 \frac{r_s I_\beta + r_a I_{a,\beta}}{N}\right)$$

  end
  Do a normal SEIIaR step for each group of people $\alpha, \beta = 1, \ldots, m$, exactly as described in the introduction to update to the present state $X_{(i+j+1)\alpha\beta\gamma}$;
  $i \leftarrow i + S$;
  for $j = 0, \ldots, S-1$ **do**
    Repeat the procedure in the loop above, but with $\alpha \leftrightarrow \beta$, i.e. perform the sums over the *third* not *second* axis of $\mathbf{X}$, and the *second* of $\mathbf{M}$. ;
  end
end
Algorithm 5: Description of implementation of the SEIIaR commuter model.

[a] Note that here I have used β_0 to denote the beta value describing the value of the model, to not confuse it with the summation variables.

Listing 1 shows the same algorithm as described above implemented in python.

```
@nb.njit()
def SEIIaR_commuter_step(X,Pse,Pei,Peia,Pir,Piar):

    Dse           = np.random.binomial(X[0],Pse)
    Dei,Deia,Dee = np.random.multinomial(X[1], (Pei,Peia,1-Pei-Peia) )
    Dir           = np.random.binomial(X[2], Pir)
    Diar          = np.random.binomial(X[3], Piar)

    return np.array([X[0] - Dse,
                     X[1] - Dei - Deia + Dse,
                     X[2] - Dir + Dei,
                     X[3] - Diar + Deia,
                     X[4] + Dir + Diar])

@nb.njit()
def SEIIaR_commuter(M,X_0,tN,dt):

    # set this to ones initially, but change it
    # for each step, as it depends on the number of infected.

    m    = np.shape(M)[0]
    Pse  = np.ones(m)

    Pei  = fs * (1 - np.exp(-dt/tau_E))
    Peia = fa * (1 - np.exp(-dt/tau_E))
    Pir  = 1 - np.exp(-dt/tau_I)
    Piar = 1 - np.exp(-dt/tau_I)

    T = np.arange(0,tN+dt,dt)
    n = len(T)

    X          = np.zeros((n,m,m,5),dtype = np.int64)
    X[0,:,:,:] = X_0

    # The loop below assumes that the simulation is
    # runned for a whole number of days, with 0.5 divisible by dt,
    # so that the number of steps are evenly split into night and day.

    assert( int(0.5 / dt) * dt  == 0.5 )

    step_length = int(1/(2*dt))
    days        = int(tN)

    for day in range(days):

        i = day * 2 * step_length # current start index

        for j in range(step_length):

            # Daytime simulation

            N = np.sum(M,axis = 0)
            I = X[i+j,:,:,2:4]
            I = np.sum(I, axis = 0)
            Pse = 1 - np.exp(- dt * beta * 1/N * ( rs * I[:,0] + ra * I
    [:,1] ))
            for k in range(m):
```

```
                for l in range(m):
                    X[i+j+1,k,l,:] = SEIIaR_commuter_step(X[i+j,k,l,:],
Pse[k],Pei,Peia,Pir,Piar)

        i += step_length

        for j in range(step_length):

            # Night simulation

            N = np.sum(M,axis = 1)
            I = X[i+j,:,:,2:4]
            I = np.sum(I, axis = 1)
            Pse = 1 - np.exp(- dt * beta * 1/N * ( rs * I[:,0] + ra * I
[:,1] ))
            for k in range(m):
                for l in range(m):
                    X[i+j+1,l,k,:] = SEIIaR_commuter_step(X[i+j,l,k,:],
Pse[k],Pei,Peia,Pir,Piar)

    return T, X
```

Listing 1: SEIIaR commuter algorithm implemented in python

To check that the system behaves as expected, we simulate the same scenario, only with another matrix which now represents *no* flow of workers to the other areas during daytime:

$$\widetilde{\mathbf{M}} = \begin{bmatrix} 10000 & 0 \\ 0 & 100000 \end{bmatrix}. \tag{12}$$

The time-evolution of the different variables are shown in figure 13, in which it is apparent that the exposed people in the small city never infect those in the large city, as we expect. Furthermore, we see that the epidemic in area 1 evolves slower in this case — the peak of the infected is delayed by approximately 25 days. This is also as expected, since more potential carriers of infection are available when the two towns mix during daytime. The expected behaviour is also observed when the exposed start out in area 2, but a plot of this is not included, for brevity.

To do a second test we let 9999 of the people living in town 1 work in town 2, and leave only one working in town 1[3]. That is, we use the population matrix in equation (13)

$$\overline{\mathbf{M}} = \begin{bmatrix} 1 & 9999 \\ 0 & 100000 \end{bmatrix}. \tag{13}$$

We simulate the same situation, with 25 of the 9999 people initially exposed. This should make the evolution of the variables approximately in sync. This is indeed what we observe, in figure 14, as the number of infected people in each town are closely synced.

[3]The reason I leave one worker in town 1 is that having 0 people in area 1 at daytime makes the solver break down, as we divide by 0 in the expression for $P_{S\to E}$

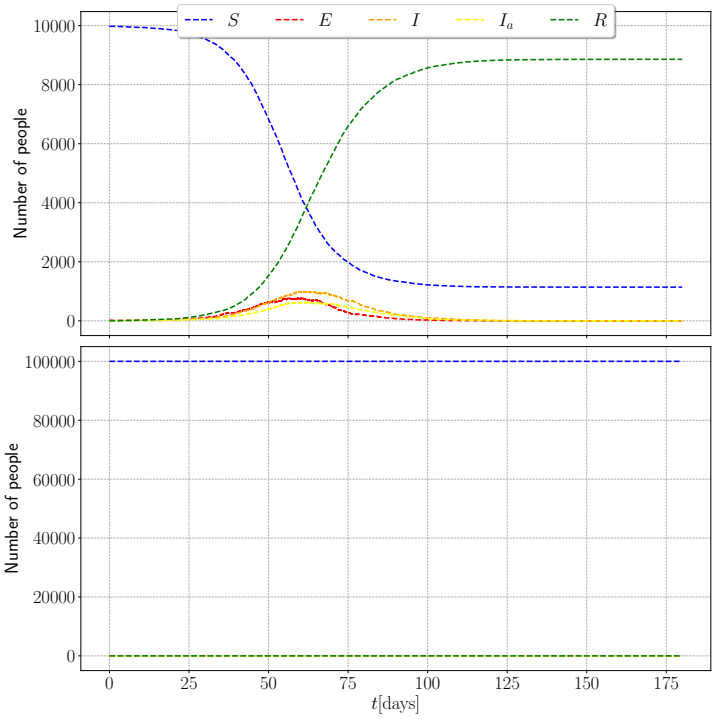

Figure 13: Solutions of Stochastic SEIIaR commuter model for the 2-city case, with the matrix $\widetilde{\mathbf{M}}$ in equation (12).

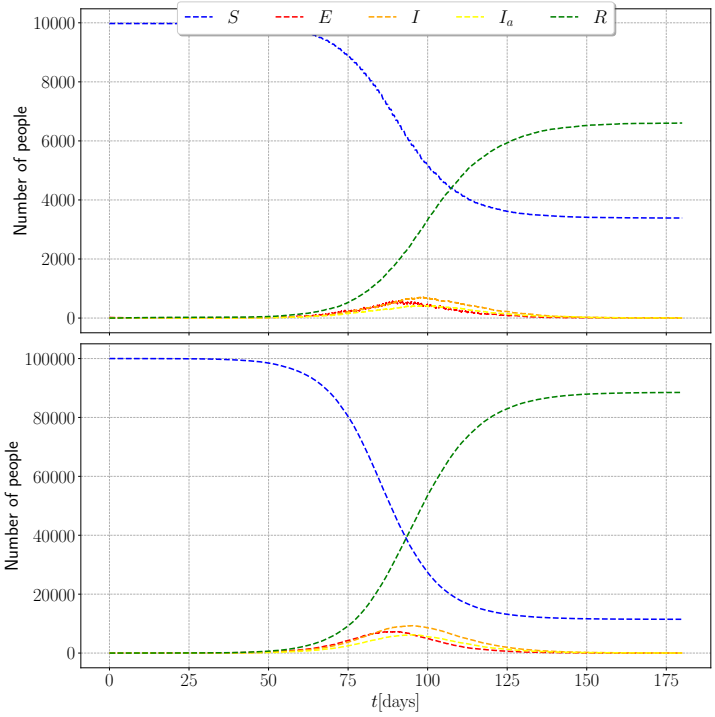

Figure 14: Solutions of Stochastic SEIIaR commuter model for the 2-city case, with the matrix $\overline{\mathbf{M}}$ in equation (13).

5 Problem 2D: Larger stochastic SEIIaR Commuter model

5.1 a) 10 city simulation

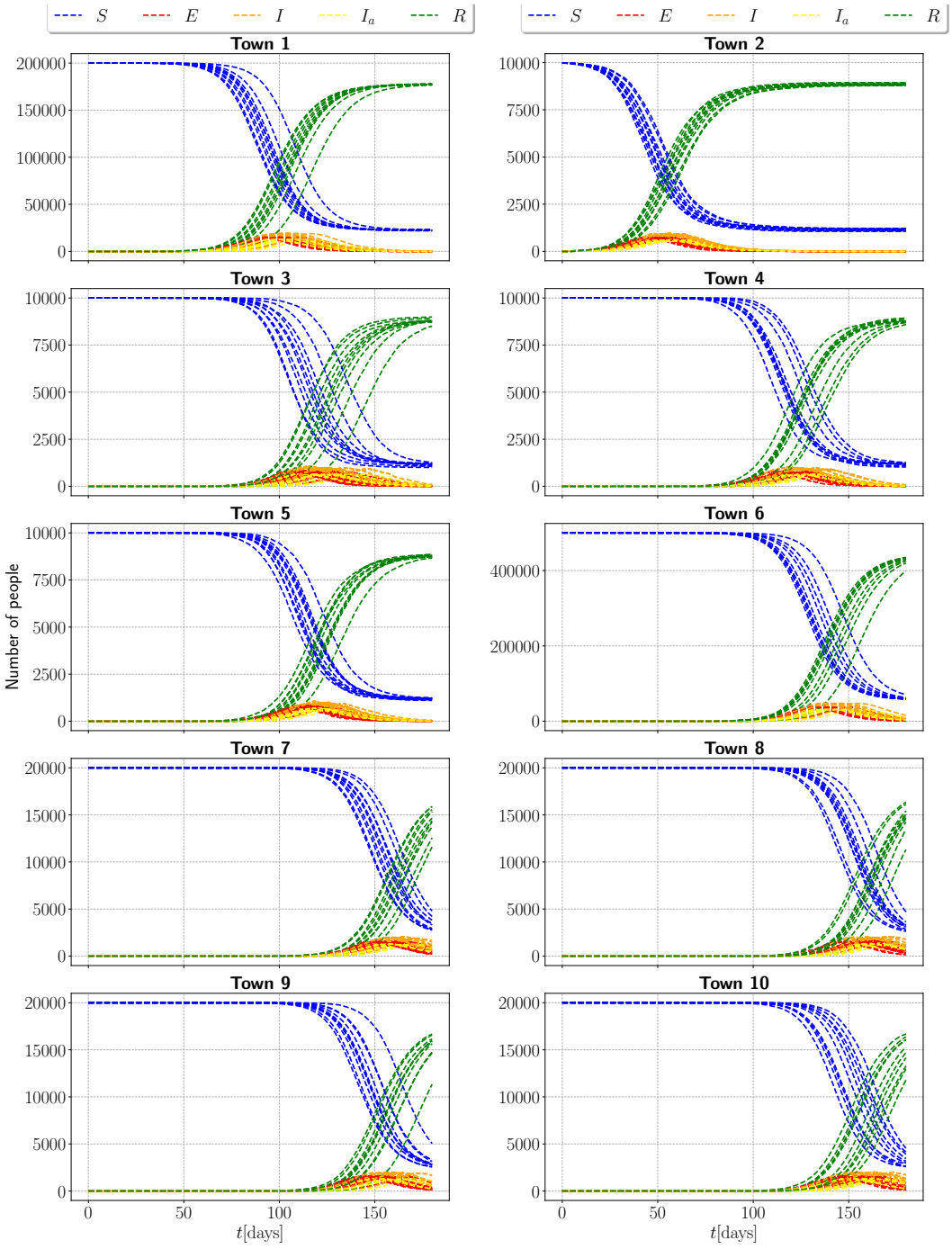

Figure 15: Solutions of Stochastic SEIIaR commuter model for the 10-city scenario.

We use the framework developed in the previous section to simulate a larger system of towns, now with 10 of them. The population matrix for this system is given in equation (11) in the problem sheet [**?**]. We initialise the system with all people susceptible, except for 25 exposed in town 2. The time evolution of 10 realisations of the different states is

shown in figure 15.

Figure 15 clearly shows that the epidemic evolves fastest in town 2, where it began. This is as expected. Furthermore, we see that the second fastest evolution is in town 1, which is the closest connection to town 2 in the sense that the commuters of town 2 only travels to town 1. This is also a reassuring fact, indicating a correct implementation. Interestingly, for the towns with less connections to town 2 — e.g. town 9 and 10 — we see that the evolution lags approximately 100 days behind, and there seems to be a wider spread between each of the 10 realisations. The increasing spread may be explained by the fact that small delays in each realisation in the beginning become exceedingly large for another town, as some time must naturally pass for the infections to be exported here.

5.2 b) 356 city simulation

In this problem we use the full population structure as handed out along with the problem set. We are here only interested in the number of municipalities with more than 10 infected people as a function of time (=: $\mathcal{N}(t)$), so we modify the commuter solver shown in listing 1 to only keep the current and previous state of the system at each time step, and to calculate $\mathcal{N}$ for each time step. This is done to avoid using too much memory, which becomes a real problem if we try to allocate space for some thousand matrices of size $365 \times 365 \times 5$. Although simulations of this system with a small time step tends to take very long time, making the solver more "greedy" at least opens the door to running the 10 simulations in parallel.

$\mathcal{N}(t)$ for 10 realisations of this simulation is shown in figure 16.

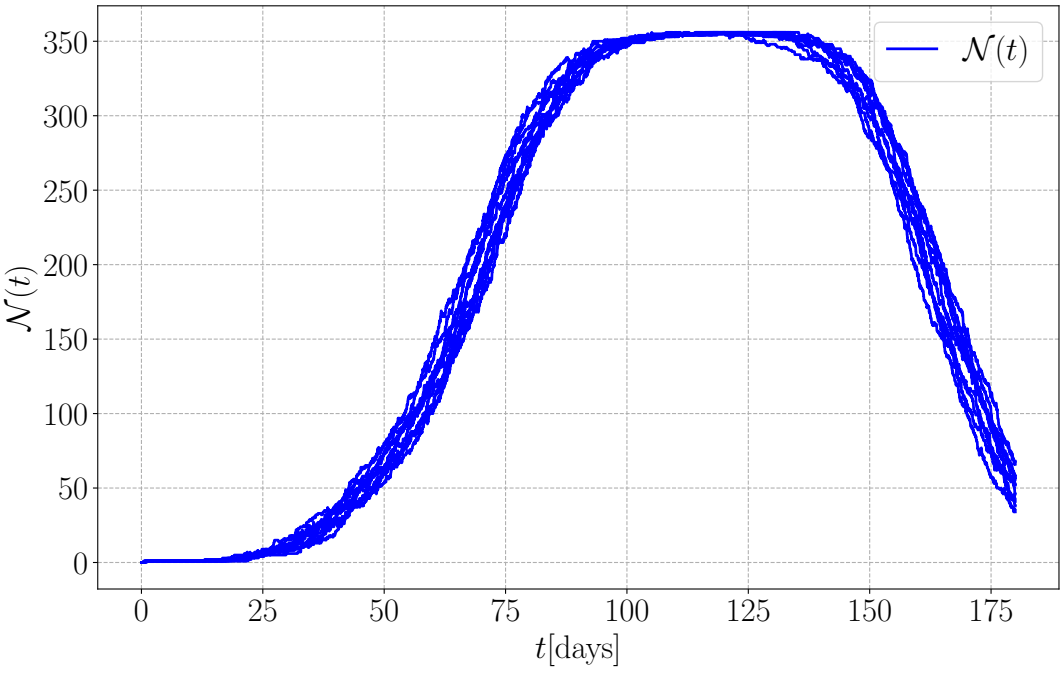

Figure 16: Number of municipalities with more than 10 infected people a a function of time.

5.3 c) Reduced number of commuters in 356 city simulation

To study the effect of making more people work from home, we change the population matrix by dividing the off-diagonal elements by 10, rounding to the nearest integer, and adding the remaining number of people to the diagonal. We run the same simulation as above, and display the number municipalities with more than 10 infected people as a function of time, $\mathcal{N}(t)$, in figure 17. Clearly, the effect of reduced travelling is to flatten this curve. In figure 17 we see that $\mathcal{N}$ saturates after about 100 days, while in this case it never even reaches the case where $\mathcal{N}$ is 356. The increase in $\mathcal{N}(t)$ is seen to be much slower in this case, and the peak of the curve occurs after roughly 150 days.

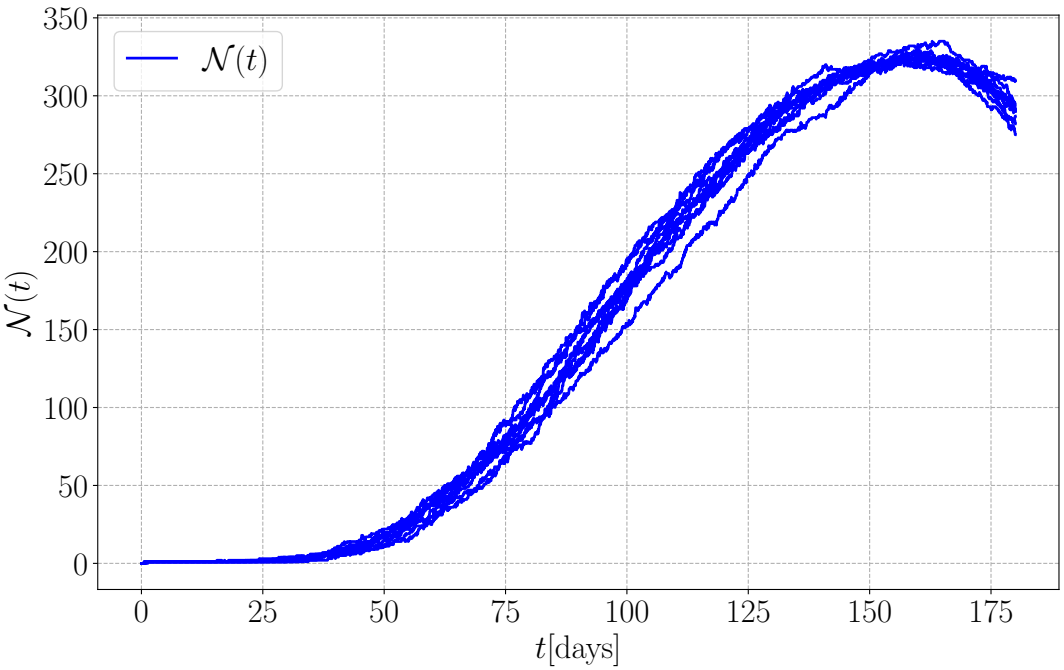

Figure 17: Number of municipalities with more than 10 infected people a a function of time, with reduced travelling.

References

[Nor21] Tor Nordam. Exam 2021 - Computational physics TFY4235, 2021.

[Was04] Larry Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer Texts in Statistics. Springer, New York, 2004.