

Exam: Epidemic modelling

Candidate nr. 10037

Department of Physics, Norwegian University of Science and Technology, Trondheim
Norway

TFY4235 - Computational physics
(*Last updated on May 4, 2021*)

Abstract

Short abstract

Contents

I	Introduction	2
II	Code overview	3
III	Results and Discussion	4
1	Problem 2A: Deterministic SIR model	4
1.1	a)	4
1.2	b)	4
1.3	c) Flattening the curve.	5
1.4	d) Vaccinations	6
2	Problem 2B: Stochastic SIR model	7
2.1	a)	7
2.2	b)	7
2.3	c) Probability of an outbreak	7
3	Problem 2C: Stochastic SEIIaR model	10
3.1	a)	10
3.2	b) Probability of outbreak dependence on r_s	11
4	Problem 2D: Stochastic SEIIaR Commuter model	14
4.1	a) Commuter model for a two-town system	14
4.2	b) Description of implementation & tests	15
5	Problem 2D: Larger stochastic SEIIaR Commuter model	19
5.1	a) 10 city simulation	19
5.2	b) 356 city simulation	20

PART

I

Introduction

Code overview

PART
II

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. 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 [Nor21]. The expressions for $S(\infty)$ and $R(\infty)$ are solved using the non-linear equation solver `fsolve` from `scipy`.

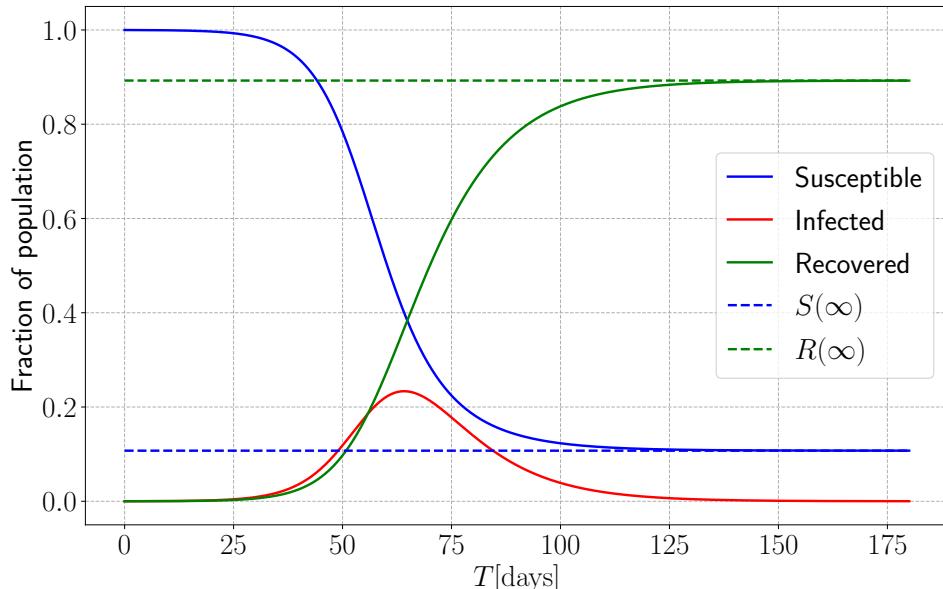


Figure 1: SIR equations with $\beta = 0.25 \text{ day}^{-1}$, $\tau = 10 \text{ day}$.

1.2 b)

For the early developments of the epidemic, the expression for I can be simplified to [Nor21]

$$\frac{dI}{dt} = (\beta - 1/\tau)I. \quad (1)$$

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.

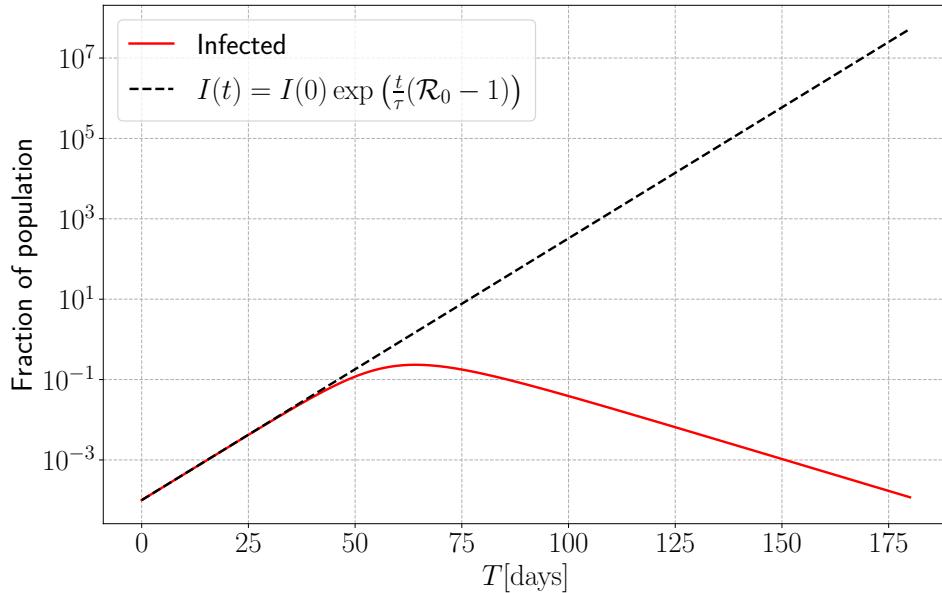


Figure 2: Infected people compared with the analytical approximation at the early stages.

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  $\text{tol}$ ;
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  $\text{err} := |\mathcal{I} - 0.2| > \text{tol}$  do
  if  $\mathcal{I} > 0.2$  then
    |  $\beta \leftarrow \beta \cdot 2^{\text{err}}$ 
  else
    |  $\beta \leftarrow \beta \cdot 2^{-\text{err}}$ 
  end
end
Run the simulation of the SIR with the same parameters as in 2Aa, except with
the new value for  $\beta$ ;
Recalculate  $\mathcal{I} := \max_{t \in [0, 180]} I(t)$ ;
Algorithm 1: Finding the largest beta keeping  $\max I$  less than 0.2.

```

¹Note 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 beta 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 this 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 15a days of the simulation;
If the initial slope is negative it indicates that the outbreak dies out by itself;
while slope> 0 do
     $R(0) \leftarrow R \cdot 2^{\text{slope}}$  ;
    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.

^aAs people are typically sick for 10 days ($\tau = 10$) I suppose this time scale is long enough.

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.

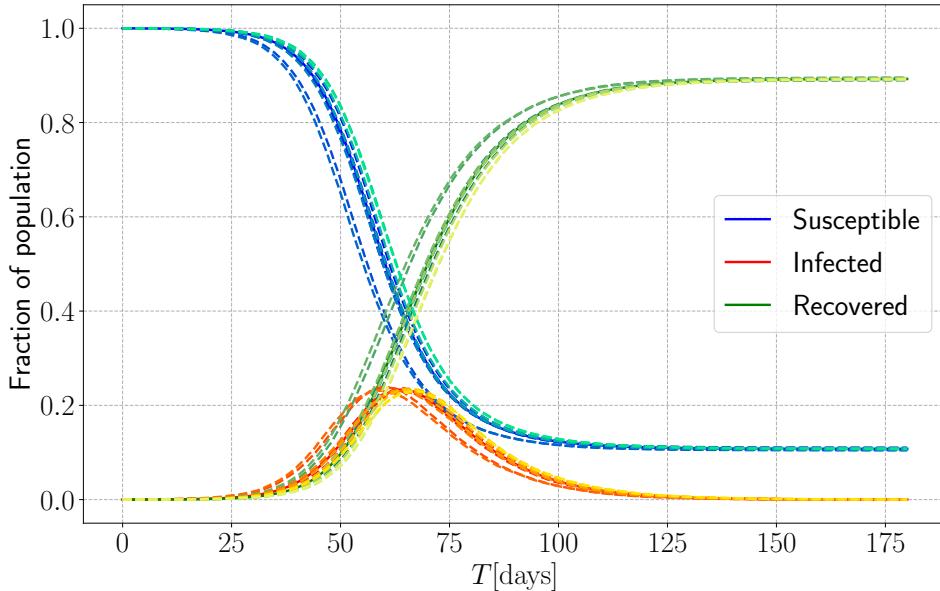


Figure 3: Solution of stochastic SIR equations with $\beta = 0.25 \text{ day}^{-1}$, $\tau = 10 \text{ day}$.

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 4. 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.

2.3 c) Probability of an outbreak

There will always be a certain probability for an outbreak disappearing by itself for the stochastic model. In the present subsection we estimate this probability for an initial number of infected people = 1, 2, ..., 10. This is done by the following procedure:

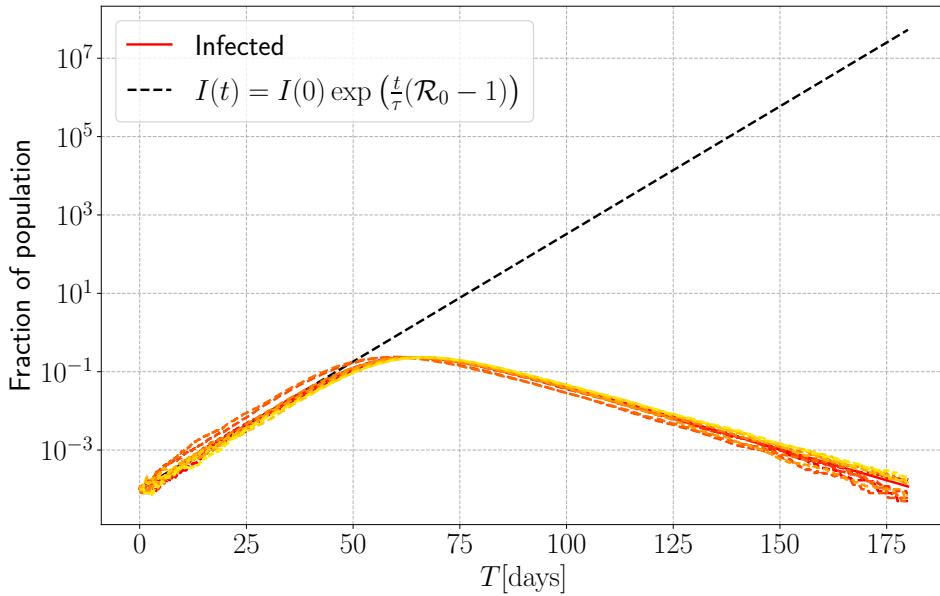


Figure 4: Infected people compared with the analytical approximation at the early stages. Stochastic and deterministic model.

Choose the parameters of the model as those given in the exam sheet [Nor21],
but with $T = 30$ days^a. ;

Choose a batch-size B .;

for $I = 1, 2, \dots, 10$ **do**

Initialise an empty vector of length B : $\mathbf{X} = [0, \dots, 0]$; ;

for $n = 1, \dots, B$ **do**

Run the simulation with initial number of infected people = I ; ;

Calculate the slope in the semi-log axes for $I(t)$; ;

if $slope \leq 0$ **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

$$\sqrt{\text{Var}(\hat{p})} = \sqrt{\frac{p(1-p)}{B}}. \quad (2)$$

end

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

^aAs 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 [Was04, 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} i.e. the expectation value of X_n , 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 (2) follows.

The probability of an outbreak as a function of the initial number of infected people are shown in figure 5 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 less than 0.6. This ultimately shows that the stochastic model has a more realistic feature to it than the deterministic one, in that these scenarios might occur.

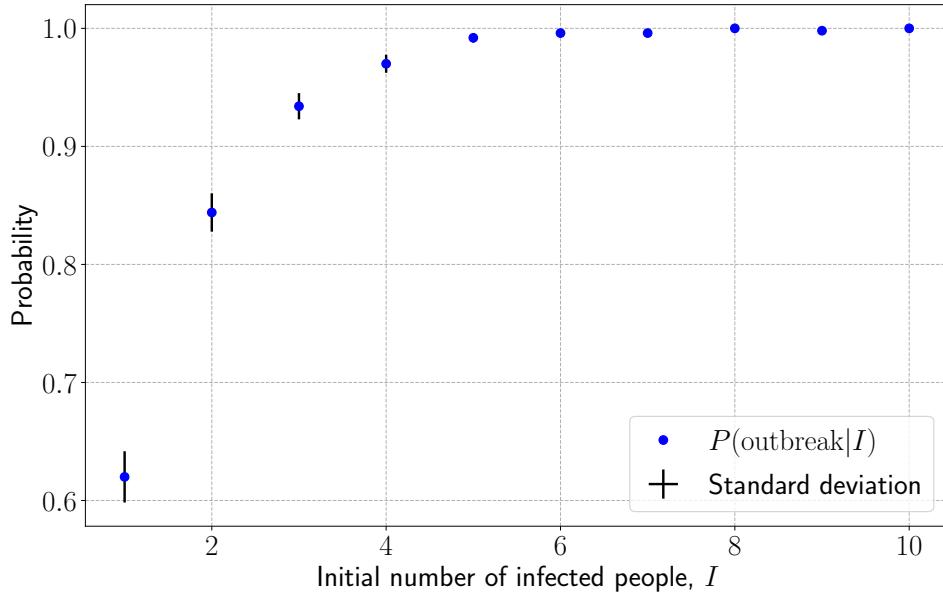


Figure 5: Probability of an outbreak as a function of initial number of infected people.

3 Problem 2C: Stochastic SEIIaR model

3.1 a)

The time-development of all the variables in the SEIIaR model is shown in figure 6, 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 7.

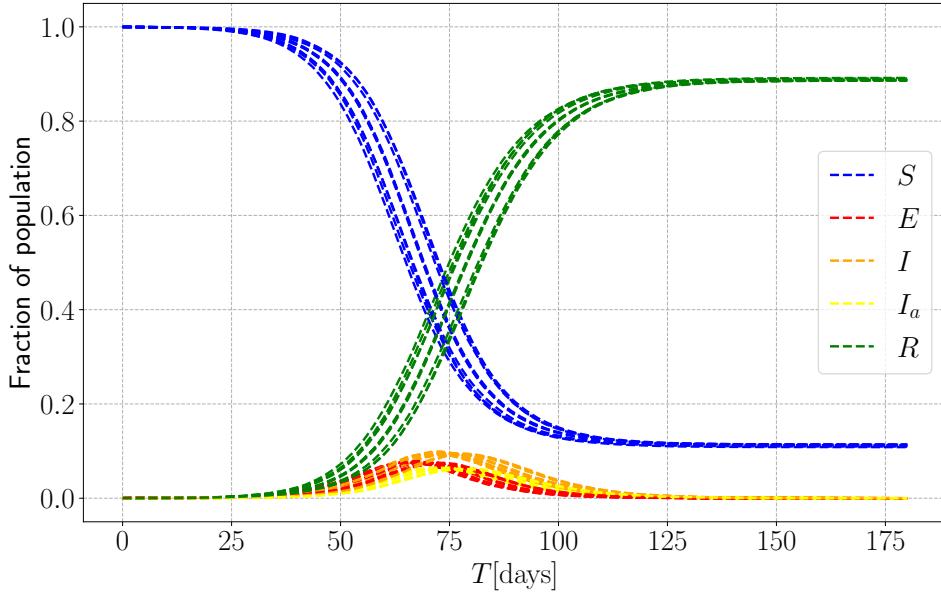


Figure 6: Solution of the stochastic SEIIaR-equations.

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 $\tau_E = 3$ 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 adds another layer of realism to our model in the sense that people do not get sick right away. 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 the probabilities of transitioning to either of these states are the same.

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 8. This shows that the two stochastic models are more or less identical, as expected.

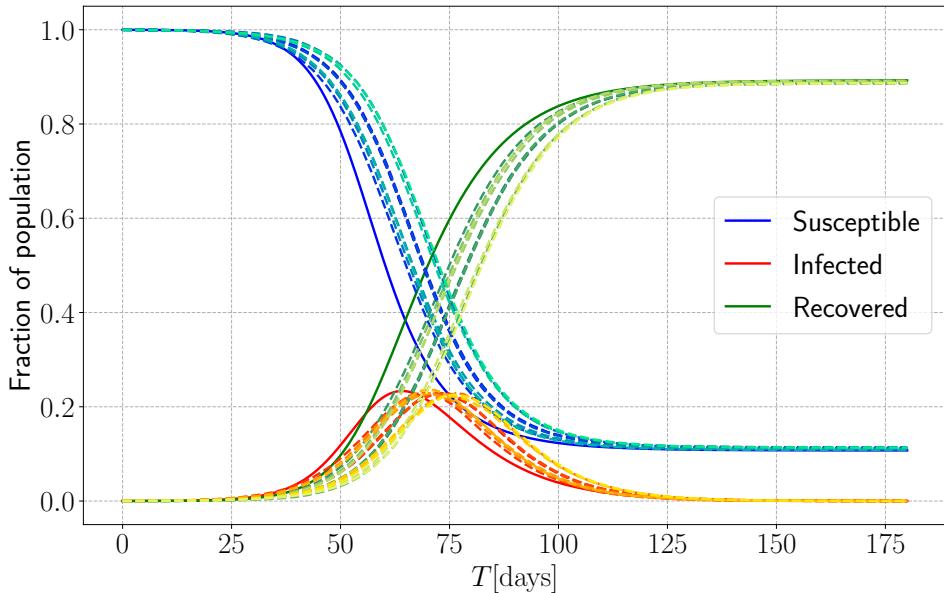


Figure 7: Comparison of the solution of the Stochastic SEIIaR-equations with the deterministic SIR-model. The number of infected people I in the stochastic model is $E + I + I_a$.

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 emulating 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):

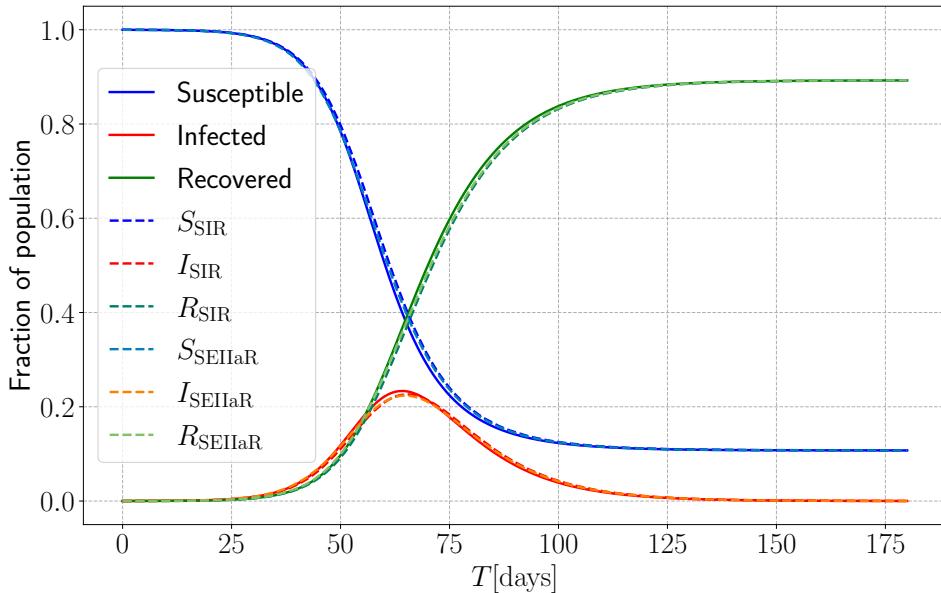


Figure 8: Solution of the stochastic SEIIaR-equations compared with the stochastic and deterministic SIR-equations for the case of identical parameters.

Choose the parameters of the model as those given in the exam sheet [Nor21],
but with $T = 30$ 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 $slope \leq 0$ **then**

| $X_n = 0$

else

| $X_n = 1$

end

end

Estimate the probability of an outbreak for I initially infected by

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

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 .

^aAs 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 9. 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 $r_s \rightarrow 0$, 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.

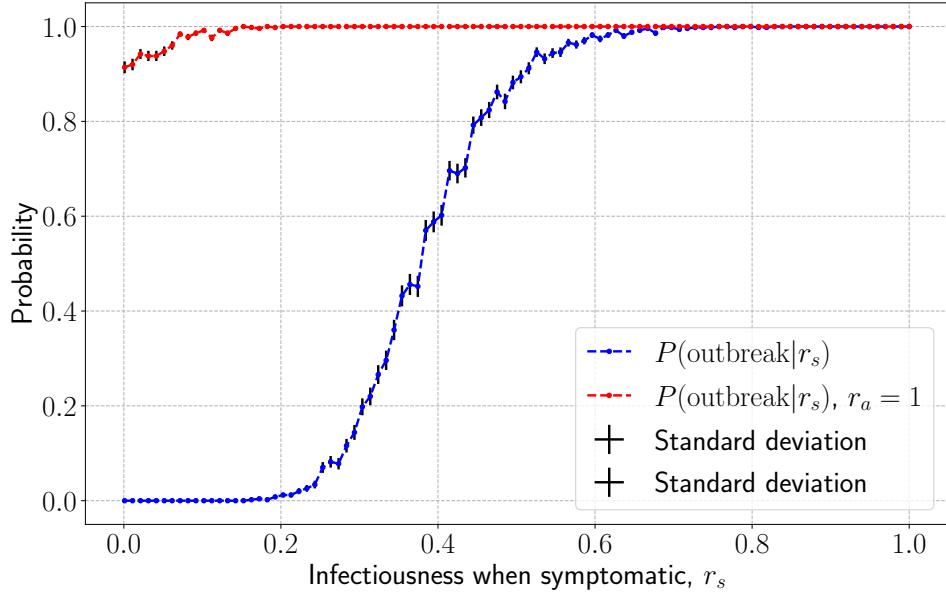


Figure 9: Probability of an outbreak as a function of r_s .

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 [Nor21], with 25 initially exposed people working and living in town 1. The results of this simulation is shown in figure 10. 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.

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 the probability of transitioning to either of these states are equal.

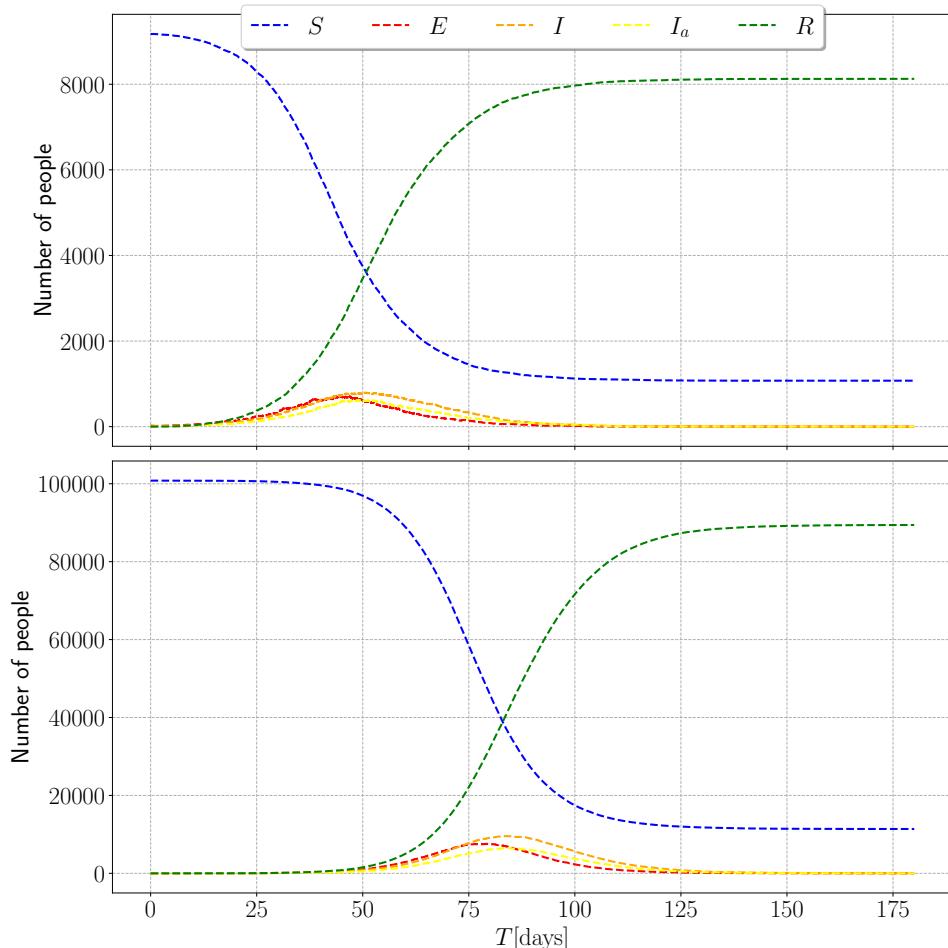


Figure 10: Solutions of Stochastic SEIIaR commuter model for the 2-city case.

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:

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

Calculate the number of days $D = \text{int}(t_N)$, and the number of steps per half day $S = \text{int}(1/(2\Delta t))$;

for $d = 0, \dots, D - 1$ **do**

$i \leftarrow 2 \times d \times S$;

for $j = 0, \dots, S - 1$ **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

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

end

Do a normal SEIIaR step for each group of people $\alpha, \beta = 1, \dots, m$, exactly as described in the previous section.;

$i \leftarrow i + S$;

for $j = 0, \dots, S - 1$ **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.

^aNote 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.

```

1 @nb.njit()
2 def SEIIaR_commuter_step(X,Pse,Pei,Peia,Pir,Piar):
3
4     Dse          = np.random.binomial(X[0],Pse)
5     Dei,Deia,Dee = np.random.multinomial(X[1], (Pei,Peia,1-Pei-Peia) )
6     Dir          = np.random.binomial(X[2], Pir)
7     Diar         = np.random.binomial(X[3], Piar)
8
9     return np.array([X[0] - Dse,
10                  X[1] - Dei - Deia + Dse,
11                  X[2] - Dir + Dei,
12                  X[3] - Diar + Deia,
13                  X[4] + Dir + Diar])
14
15 @nb.njit()
16 def SEIIaR_commuter(M,X_0,tN,dt):
17
18     # set this to ones initially, but change it
19     # for each step, as it depends on the number of infected.
20
21     m      = np.shape(M)[0]
22     Pse   = np.ones(m)
23
24     Pei   = fs * (1 - np.exp(-dt/tau_E))
25     Peia  = fa * (1 - np.exp(-dt/tau_E))
26     Pir   = 1 - np.exp(-dt/tau_I)
27     Piar  = 1 - np.exp(-dt/tau_I)
28
29     T = np.arange(0,tN+dt,dt)
30     n = len(T)
31
32     X      = np.zeros((n,m,m,5),dtype = np.int64)
33     X[0,:,:,:]= X_0
34
35     # The loop below assumes that the simulation is
36     # runned for a whole number of days, with 0.5 divisible by dt,
37     # so that the number of steps are evenly split into night and day.
38
39     assert( int(0.5 / dt) * dt == 0.5 )
40
41     step_length = int(1/(2*dt))
42     days       = int(tN)
43
44     for day in range(days):
45
46         i = day * 2 * step_length # current start index
47
48         for j in range(step_length):
49
50             # Daytime simulation
51
52             N = np.sum(M, axis = 0)
53             I = X[i+j,:,:,:2:4]
54             I = np.sum(I, axis = 0)
55             Pse = 1 - np.exp(- dt * beta * 1/N * ( rs * I[:,0] + ra * I
56             [:,1] ))
56             for k in range(m):

```

```

57         for l in range(m):
58             X[i+j+1,k,l,:] = SEIIaR_commuter_step(X[i+j,k,l,:], 
Pse[k],Pei,Peia,Pir,Piar)
59
60             i += step_length
61
62         for j in range(step_length):
63
64             # Night simulation
65
66             N = np.sum(M, axis = 1)
67             I = X[i+j,:,:,:2:4]
68             I = np.sum(I, axis = 1)
69             Pse = 1 - np.exp(- dt * beta * 1/N * ( rs * I[:,0] + ra * I
[:,1] ))
70             for k in range(m):
71                 for l in range(m):
72                     X[i+j+1,l,k,:] = SEIIaR_commuter_step(X[i+j,l,k,:], 
Pse[k],Pei,Peia,Pir,Piar)
73
74     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:

$$\widetilde{\mathbf{M}} = \begin{bmatrix} 10000 & 0 \\ 0 & 100000 \end{bmatrix}. \quad (3)$$

The time-evolution of the different variables are shown in figure 11, 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.

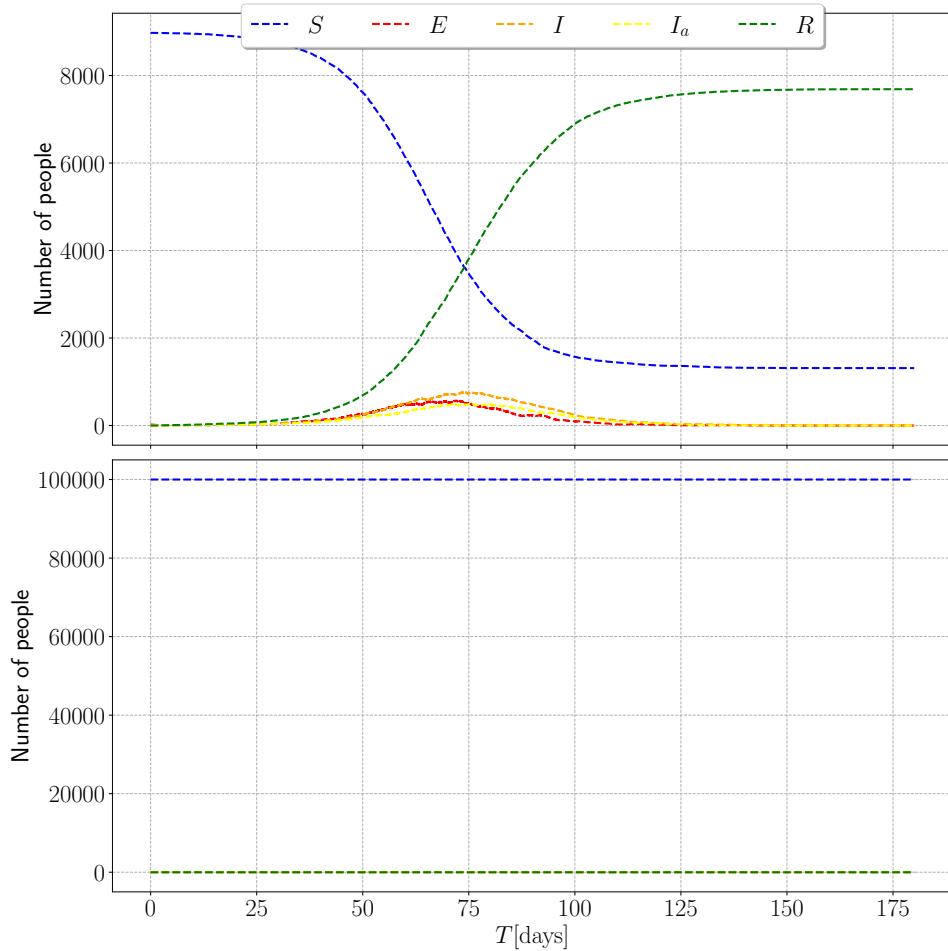


Figure 11: Solutions of Stochastic SEIIaR commuter model for the 2-city case, with the matrix $\widetilde{\mathbf{M}}$ in equation (3).

5 Problem 2D: Larger stochastic SEIIaR Commuter model

5.1 a) 10 city simulation

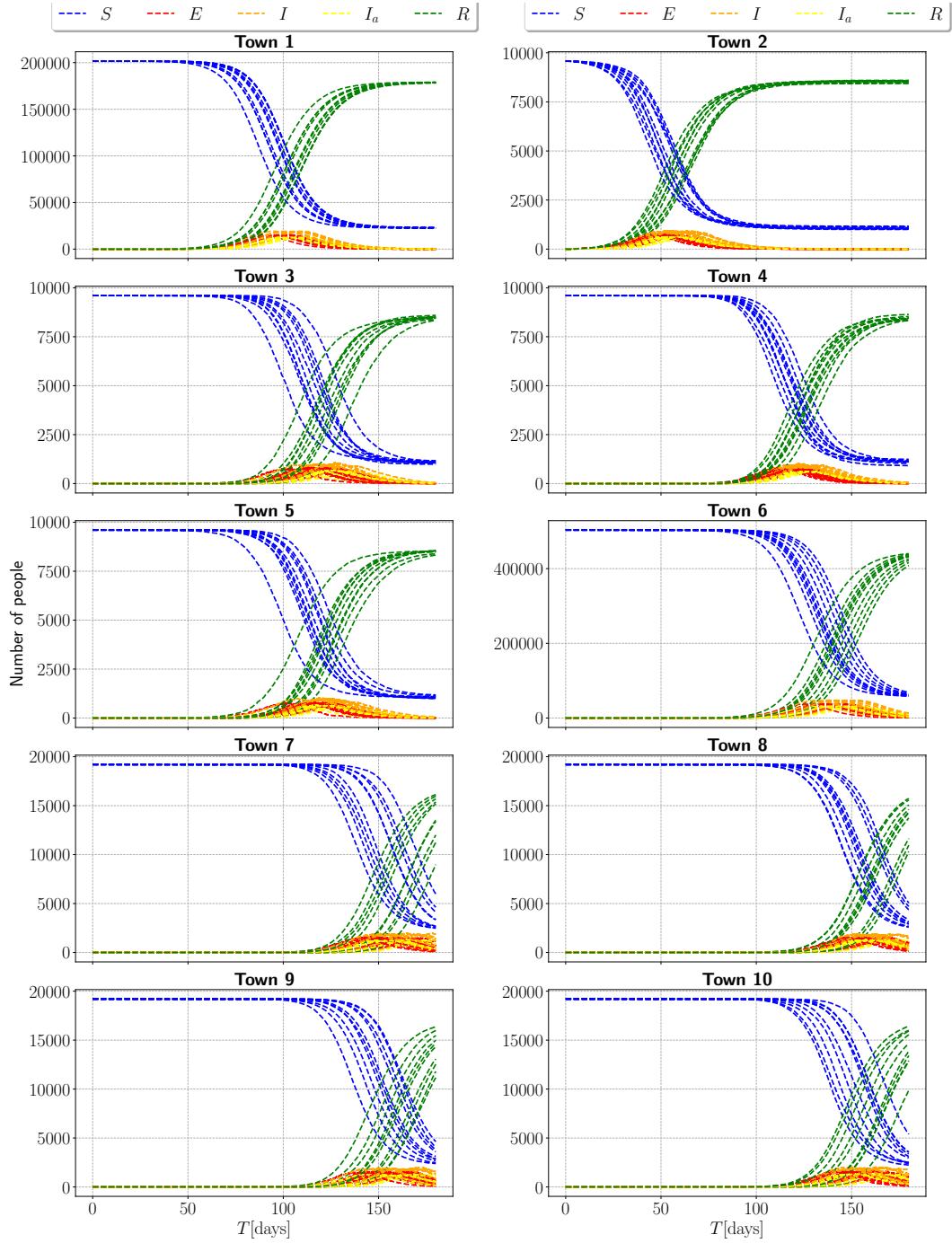


Figure 12: Solutions of Stochastic SEIIaR commuter model for the 10-city scenario.

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 [Nor21]. 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 12.

Figure 12 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 more 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 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 arrays some thousand matrices of size $365 \times 365 \times 5$.

$\mathcal{N}(t)$ for 10 realisations of this simulation is shown in figure 13.

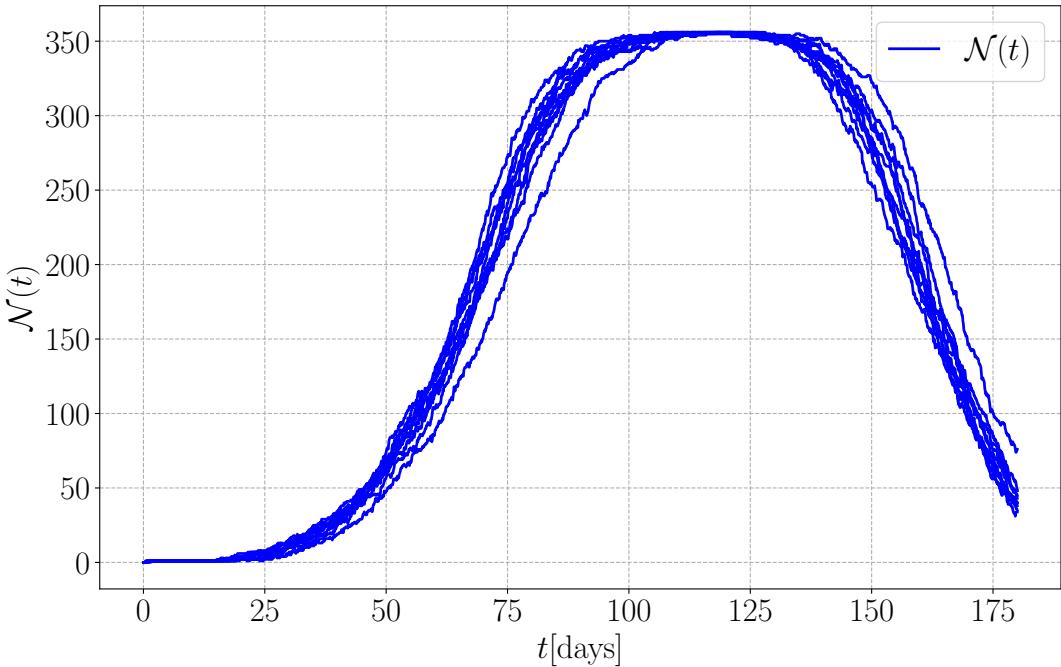


Figure 13: Number of municipalities with more than 10 infected people a a function of time.

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.