

# Disease Modeling - FYS3150 Computational Physics

Nicholas Karlsen

This is an abstract

## INTRODUCTION

## THEORY, ALGORITHMS AND METHODS

### SIRS Model

The SIRS model is a mathematical model from epidemiology describing how infectious disease evolves within a population, and is a part of a family of similar models in epidemiology with various different features. In the SIRS model, the total population ( $N$ ) is divided into three groups

- Susceptible (S) : People who do not have the disease, and are not immune to it.
- Infected (I) : People who are infected with the disease.
- Recovered (R) : People who has recovered from the infection, and have developed immunity.

Where in the simplest case, the permitted traversal from one group to another follows a cyclical pattern  $S \rightarrow I \rightarrow R \rightarrow S$ , hence the name SIRS.

The rate of traversal is governed by a set of coupled differential equations,

$$\begin{aligned} S' &= cR - \frac{aSI}{N} \\ I' &= \frac{aSI}{N} - bI \\ R' &= bI - cR \end{aligned} \quad (1)$$

Where the constants  $a, b, c$  are governing the

- rate of transmission
- rate of recovery
- rate of immunity loss

respectively, with dimension inverse time. For the purposes of this report, we will not consider any particular unit of time, but rather the dynamics of the system, as the timescales at which different diseases operate on vary. However, based on data (INSERT CITATIONS) a lot of common diseases are observed to operate on a scale of days, whilst some operate on a scale of years.

We note here that each term in the system of equations is reflected in opposite, for a different group. For

example, the  $cR$  term in  $S'$  is mirrored in opposite in  $R'$  as  $-cR$ . As defined earlier,  $c$  denotes the rate of immunity loss, so these terms essentially moves people in the R group to the S group. The result of this is that the total population is conserved in this system, such that we have a constant  $N = S(t) + I(t) + R(t)$ .

As shown in Hjorth-Jensen [1], we can then substitute this into the set of equations 1 and derive equations for how the system distributes itself in a steady state

$$\begin{aligned} s^* &= \frac{b}{a} \\ i^* &= \frac{1 - \frac{b}{a}}{1 + bc} \\ r^* &= \frac{b(1 - \frac{b}{a})}{c(1 + \frac{b}{c})} \end{aligned} \quad (2)$$

which notably only depends upon the parameters governing the system. Meaning that for some set of parameters,  $a, b, c$  the system will always converge towards the same state given enough time. Matching the behaviour of what we expect from a Markov chain governed by a regular stochastic matrix as shown in for example, Lay [2, p. 277, theorem 18].

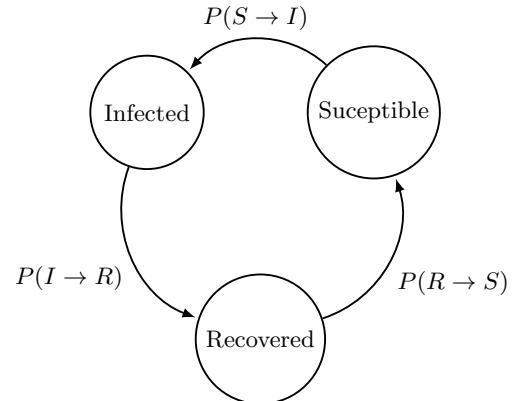


FIG. 1: Markov diagram for the basic SIRS model

### Extending the SIRS model

The model described by Eqn.1, henceforth referred to as the Basic SIRS model can be extended to model other behaviours relevant to the study of an illness. In this project, we will specifically look at 3 extensions to the

model, that is: Vital dynamics, Seasonal variation and vaccinations.

Starting off, we add the effect of vital dynamics to our set of differential equations in the following way

$$\begin{aligned} S' &= cR - \frac{aSI}{N} - dS \\ I' &= \frac{aSI}{N} - bI - (d + d_I)I \\ R' &= bI - cR - dR \end{aligned} \quad (3)$$

where  $d$  denotes the death rate of the population, and  $d_I$  the added death rate of the Infected group of the population. We see here that the new terms are not mirrored in the same way as that constitute the basic model such that the system is generally speaking no longer population conservative if any of these terms are non-zero. With a possible exception being for example if  $d = 0$ ,  $a = 0$ ,  $d_I > 0$  and  $I_0 = 0$ , in which case the population would be conserved.

Seasonal variation, or simply a periodic behaviour in the rate of transmission, added to the model by no longer having a constant  $a$ , but instead letting it be a function of time as

$$a(t) = A \cos(\omega t) + a_0 \quad (4)$$

Where  $a_0$  denotes the baseline rate of transmission,  $A$  the maximum deviation from the baseline and  $\omega$  the frequency of the variation.

and lastly, we look at a possible way modelling the effects of vaccinations by this time adding a term  $f$ , denoting the rate of vaccination to our basic model in the following way

$$\begin{aligned} S' &= cR - \frac{aSI}{N} - f \\ I' &= \frac{aSI}{N} - bI \\ R' &= bI - cR + f \end{aligned} \quad (5)$$

We see here that this addition of this terms allows someone in the  $S$  group to go immediately to the  $I$  group, which as intended. Further, this transition rate from  $S \rightarrow I$  is not dependent on the population, or size of any population groups unlike any of the other behaviours modelled thus far. Whilst we will save the discussion on the many implications this has later, we note here a particular, significant flaw in this extension to the model. Whilst it is still population conservative if we look only at  $N = S(t) + I(t) + R(t)$ , it does present the possibility of  $S(t) < 0$ , which is entirely un-realistic.

### Monte-Carlo Algorithm

We will now build a Monte-Carlo algorithm for simulating the dynamics of disease modelled by the different

sets of equations we have looked at. These simulations aims to, on average match the behaviour predicted by the differential equations, but in a more realistic way by first and foremost allowing only discrete, transitions from one group to another instead of continuous ones. And further, imposing certain restrictions to prevent unintended behaviours such as any of the groups becoming negative, which as mentioned earlier is a possibility in the system modelling the effects of vaccinations, in Eqn. 5.

In Hjorth-Jensen [1] we are methodology of translating the sets of differential equations to a set of transition probabilities for the basic model

$$\begin{aligned} P(S \rightarrow I) &= \frac{aSI}{N} \Delta t \\ P(I \rightarrow R) &= bI \Delta t \\ P(R \rightarrow S) &= cR \Delta t \end{aligned} \quad (6)$$

and a step-size

$$\Delta t = \min \left\{ \frac{4}{aN}, \frac{1}{bN}, \frac{1}{cN} \right\} \quad (7)$$

where for each step we evaluate the transition probabilities against a random number  $r \in [0, 1]$ . If the transition probability is greater than  $r$ , the transition is made, else, it isn't. This constitutes the Monte-Carlo part of the algorithm, and in order to produce expectation values we take the average of many cycles of systems initiated in the same way, and evolved following these rules.

Shown below in Pseudo-code

### SIRS Monte-Carlo

```

1 for each MC-Cycle:
2   for each  $t_i$ :
3     Compute  $P(S \rightarrow I)$ 
4     Compute  $P(I \rightarrow R)$ 
5     Compute  $P(R \rightarrow S)$ 
6
7     if  $P(S \rightarrow I) > r$  and  $S_i > 0$ :
8        $S_{i+1} \leftarrow 1$ 
9        $I_{i+1} \leftarrow 1$ 
10
11    if  $P(I \rightarrow R) > r$  and  $I_i > 0$ :
12       $I_{i+1} \leftarrow 1$ 
13       $R_{i+1} \leftarrow 1$ 
14
15    if  $P(R \rightarrow S) > r$  and  $R_i > 0$ :
16       $R_{i+1} \leftarrow 1$ 
17       $S_{i+1} \leftarrow 1$ 
```

Extending algorithm for the model extensions to the basic model is done quite easily by following the same logic outlined in Hjorth-Jensen [1]. If we consider the system the addition of vital dynamics, we have three additional permitted transitions, one for each group.  $S \rightarrow DEAD$ ,  $I \rightarrow DEAD$  and  $R \rightarrow DEAD$ . The tran-

sition probabilities for each time-step is then given by

$$\begin{aligned} P(S \rightarrow DEAD) &= dS\Delta t \\ P(I \rightarrow DEAD) &= (d + d_I)I\Delta t \\ P(R \rightarrow DEAD) &= dR\Delta t \end{aligned} \quad (8)$$

where the time step is now given by

$$\Delta t = \min \left\{ \frac{4}{aN}, \frac{1}{bN}, \frac{1}{cN}, \frac{1}{dN}, \frac{1}{(d + d_I)N} \right\} \quad (9)$$

Where the last entry of the set,  $\frac{1}{(d + d_I)N}$  can be omitted for a slight performance increase as it will always be greater than  $\frac{1}{dN}$ .

Out of the 4 variations of the SIRS model discussed in this project, this model in particular requires some extra attention, as the step size is a will vary differently in each Monte-Carlo cycle as the step size is dependent on N. This means, that when computing the expected path produced by multiple cycles of the Monte-Carlo algorithm, we interpolate between points to produce averages for some selection of times.

#### 4th-order Runge-Kutta method

The 4th-order Runge-Kutta (RK4) is one part, of a family of methods for solving ordinary differential equations (ODE). In short, the Runge-Kutta methods consists of taking a weighted average of different finite-steps, in particular, for RK4 we have the steps

$$\begin{aligned} k_1 &= f(t_i, y_i) \\ k_2 &= f\left(t_i + \frac{h}{2}, y_i + \frac{h}{2} \cdot k_1\right) \\ k_3 &= f\left(t_i + \frac{h}{2}, y_i + \frac{h}{2} \cdot k_2\right) \\ k_4 &= f(t_i + h, y_i + h \cdot k_3) \end{aligned} \quad (10)$$

where  $h$  denotes the step-size. A weighted average of these steps are then used to compute the  $i+1$ 'th element in the following way.

$$y_{i+1} = y_i + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4) + \mathcal{O}(h^5) \quad (11)$$

where  $\mathcal{O}(h^5)$  is the error in this single step, and the a global error of  $\mathcal{O}(h^4)$ . For more details, refer to Hjorth-Jensen [3, p. 250].

## RESULTS AND DISCUSSIONS

### Implementation

Before we look at the results, i would like to make a quick remark in regards to the code, and its structure. My initial plan was to implement the entirety of

the project in python, however, upon starting to consider how i might implement the vital dynamics, with its dynamic step sizes i decided to implement this, and the subsequent models in Julia<sup>1</sup>, which generally speaking is much faster than pure python. A particular benefit of Julia being that there exists APIs that lets us call Julia functions directly from python, and in reverse. These APIs allowed me to very easily offload the heavier Monte-Carlo simulations to Julia, then performing the subsequent plotting and analysis in python.

### SIRS: Basic model

We start by looking at the results produced by both the RK4 solution to Eqn. 1 as well as the Monte-Carlo results for this system, for 100 cycles. The results are presented in a quite compact form in Fig. 2, where all 4 sub-plots share a common legend. In this, and subsequent plots the colours blue, red and green indicate the S, I and R groups respectively and individual MC cycles are plotted with a transparency of 5%, such that common paths will display a greater hue than less common ones. Overlaid on top of these individual MC cycles are the ODE result, as well as the Mean Monte-Carlo result. For the sake of readability the Mean and ODE results have not been colour coded with individual colours for each group, but their belonging should be quite clear from the individual cycles on which they overlay.

In these plots, the system was initiated in the state  $S_0 = 300, I_0 = 100$  and  $R_0 = 0$  and the parameters  $a = 4, c = 0.5$  are held constant throughout all 4 sets of data. We see that for the MC simulations, only 3 out of the 4 cases converges towards the predicted set of steady states, indicated by axis ticks on the right x-axes. In the case of  $b = 3$ , we observe that neither the S or R groups converge towards their respective steady states. To investigate this further, we then refer to Fig. 3, where we note how the standard deviation of the MC results seem to rise towards a peak from the beginning, until  $t \approx 10$ , before it drops. To make sense as to why there is such a great uncertainty in this area, we refer to  $b = 1$  and  $b = 2$ . We observe that the standard deviation in all 3 groups are quite consistent for  $b = 1$ , then looking at the standard deviation for  $b = 2$  we see that standard deviation for the I group is similar to  $b = 1$ , but the R, and particularly S group has a significantly higher standard deviation. We observe here, what I believe to be the main cause of uncertainty, and deviance from the ODE solution for the Monte-Algorithm. That is, whenever one (or more) of the groups tends towards either the upper

---

<sup>1</sup> This also meant that i had to learn Julia, having only a "hello world"-level of understanding prior to this project.

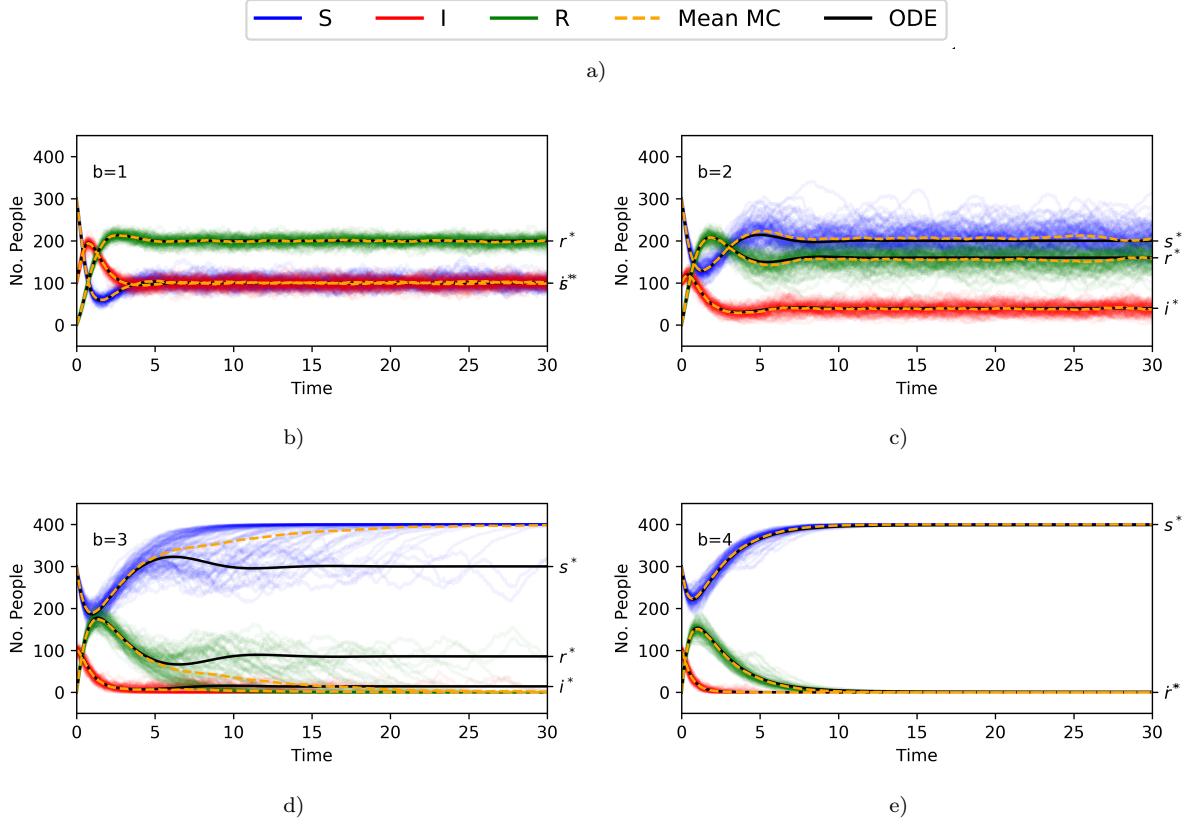


FIG. 2: The basic SIRS model simulated by the Monte-Carlo simulation for 100 cycles, as well as the ODE solutions with initial state  $S_0 = 300, I_0 = 100, R_0 = 0$ , constants  $a = 4, c = 0.5$  and  $b = 1, 2, 3, 4$ .

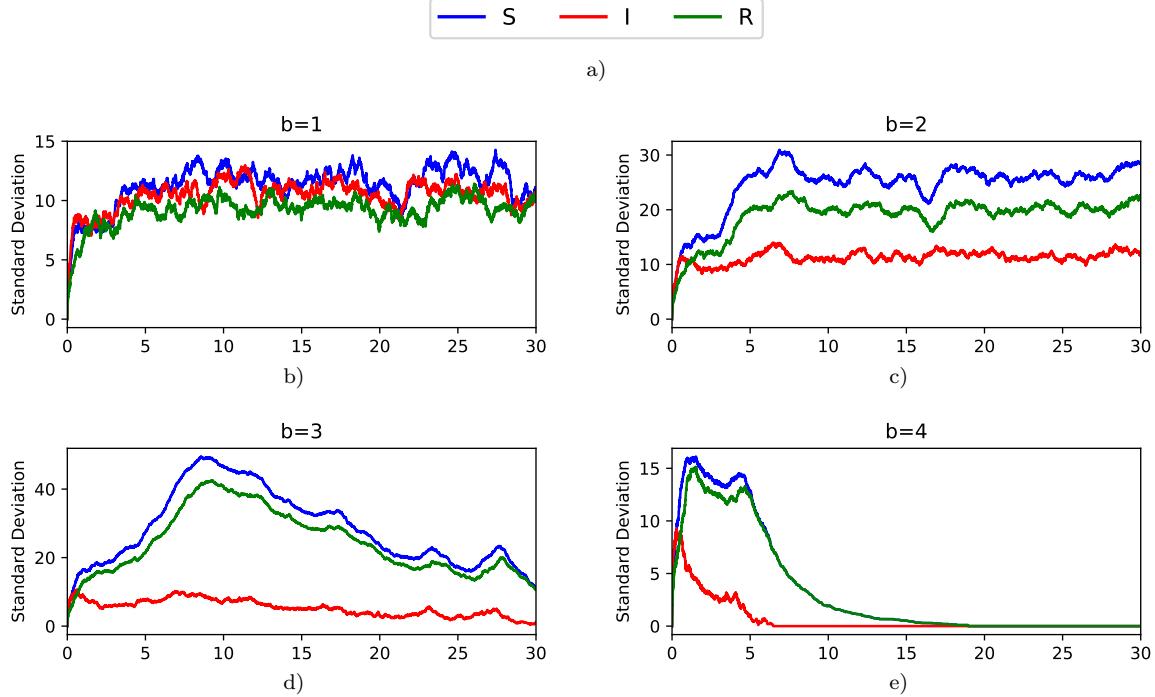


FIG. 3: The standard deviations for each  $b$  in the Monte Carlo solutions shown in Fig. 2, consisting of 100 cycles each.

or lower population limits, the uncertainty of the others

seem to rise locally, and this error is then propagated for-

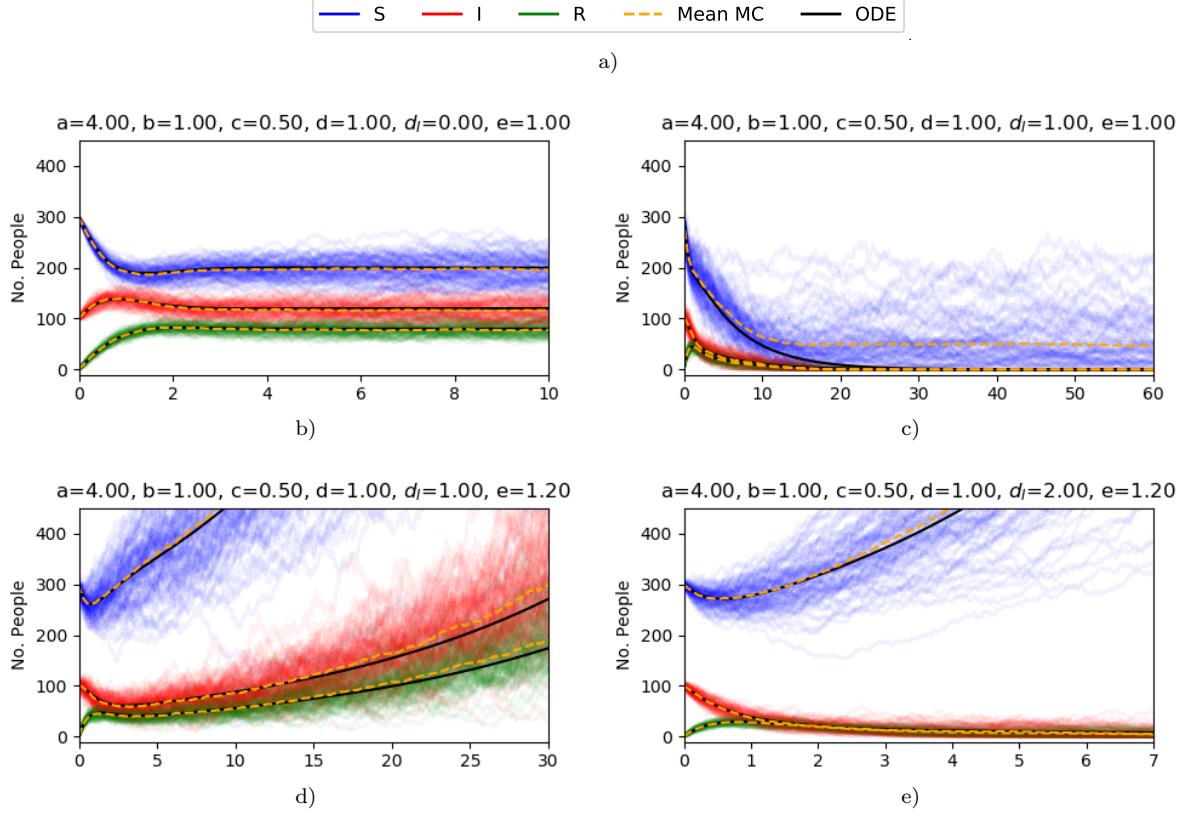


FIG. 4: The SIRS model with vital dynamics for initial state  $S_0 = 300, I_0 = 100, R_0 = 0$  and constants  $a = 4, b = 1, c = 0.5$ , and 100 MC cycles.

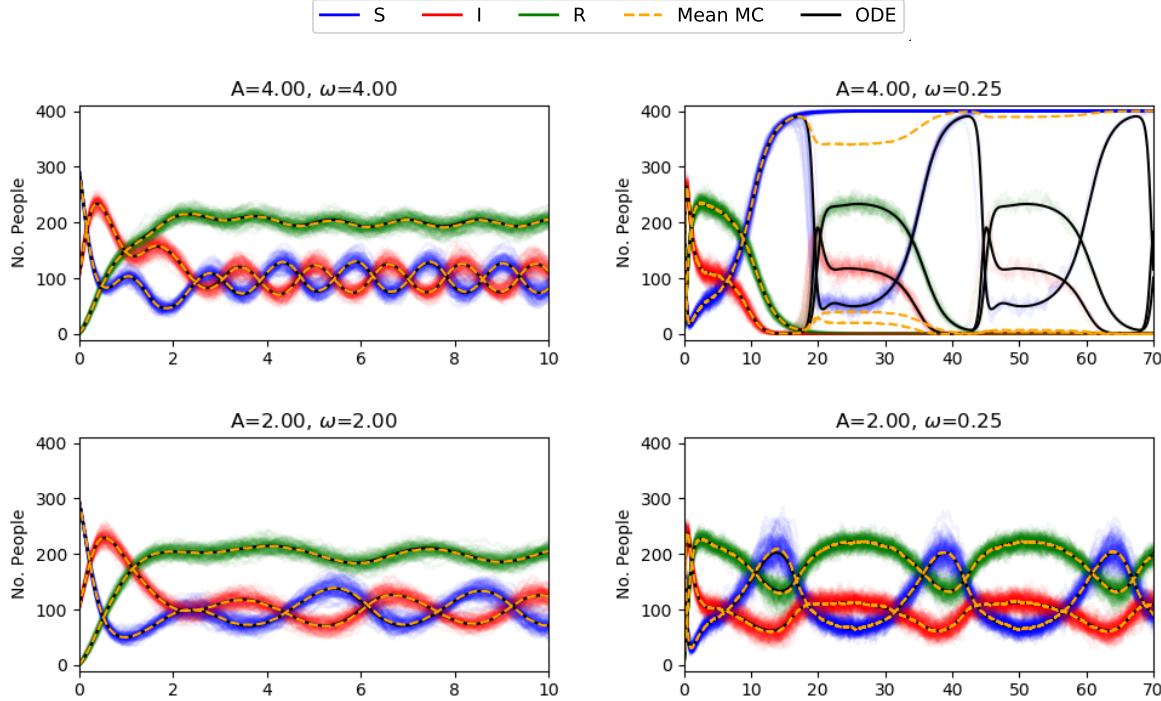


FIG. 5: The SIRS model with seasonal variation for initial state  $S_0 = 300, I_0 = 100, R_0 = 0$  and constants  $a = 4, b = 1, c = 0.5$ , and 100 MC cycles.

wards in time, even though the standard deviation in the

MC simulations goes down again. We will investigate this

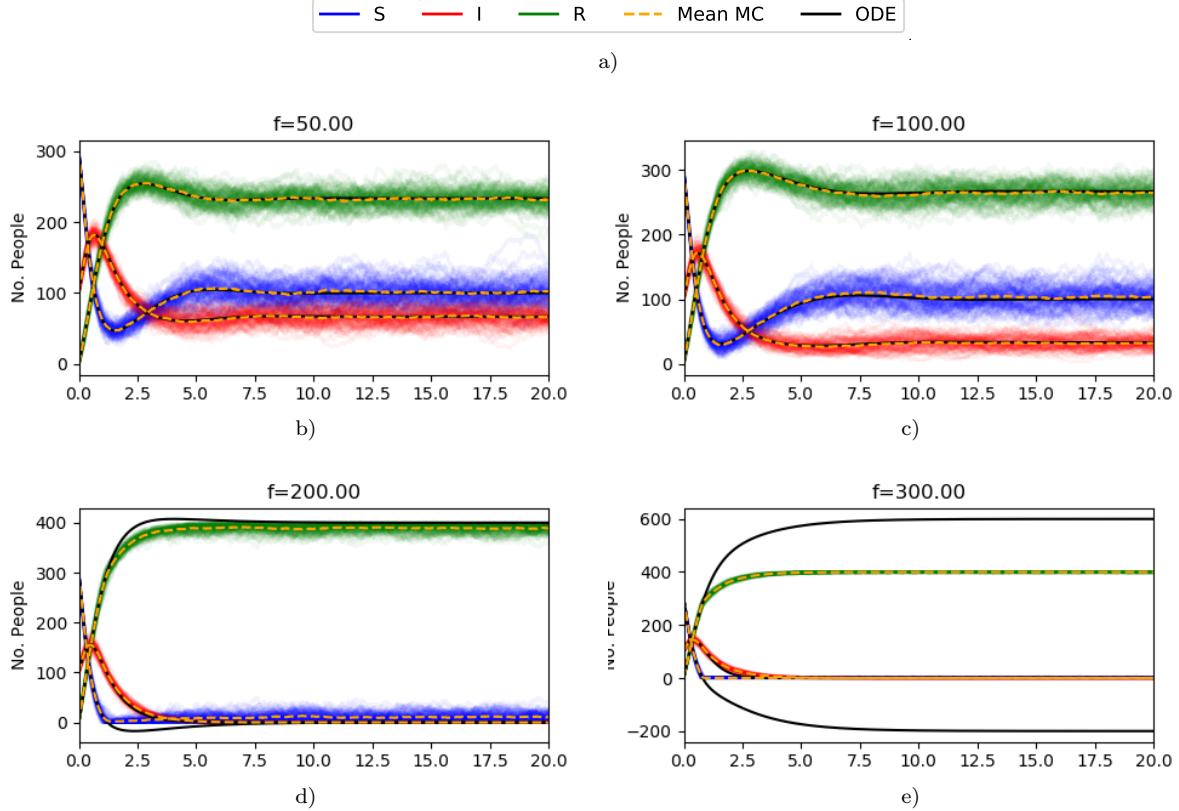


FIG. 6: The SIRS model with vaccinations for initial state  $S_0 = 300, I_0 = 100, R_0 = 0$  and constants  $a_0 = 4, b = 1, c = 0.5$ , and 100 MC cycles

hypothesis more thoroughly later in the model including seasonal variation, but for now we note that the uncertainty, and deviation from the ODE in the MC solutions isn't necessarily tied to the number of MC cycles, but the way in which the system evolves, and that we need to take into account the paths of each set of MC simulations when assessing any errors on a per-simulation basis rather than a per-system or per-number of iterations basis as is often the case in numerical methods<sup>2</sup>

ADD CONVERGENCE CHECK FOR B=3

#### SIRS: Vital Dynamics

If we now turn to Fig. 4 we see the plots for the vital dynamics system. For this model, we observe that the MC mean path follows mostly the ODE solution, deviating the most in Fig. 4d, which out of the four data sets ends up hoovering the most around the lower-limit of the permitted group sizes.

In a qualitative sense, we observe that the system is particularly sensitive to an increase in the birth

rate, which very quickly dominates over the death rate. What's also interesting, is that in a growing population, a highly deadly disease may be short-lived in this in this model.

#### SIRS: Seasonal Variation

In Fig. 5, we see results that are somewhat similar to what we saw in the basic SIRS model, but instead of converging towards a constant steady state, the system instead tends towards a periodic steady state. We see also that the frequency  $\omega$  plays into the amplitude of the variation within the groups in their steady state just as much, if not more than  $A$ . We also observe that there seems to be a natural frequency, in which we get a peak variation within groups once the steady state has been reached and we effectively observe a resonance<sup>3</sup>.

<sup>2</sup> At least in my own, limited experience.

<sup>3</sup> There should be a .gif showing this quite clearly in the README.md file in the main directory for this project.

## SIRS: Vaccinations

Finally, we turn to Fig. 6. Here we again see mostly good agreement between the MC and ODE results, except in Fig. 6e, which exemplifies how MC simulations may improve upon the systems of differential equations which they aim to emulate the behaviour of by imposing restrictions on the transitions to better reflect reality.

## CONCLUSIONS

We note also, a problem in the algorithm. That is; in what order do we evaluate the probabilities? Whether to consider  $P(S \rightarrow I)$  or  $P(R \rightarrow S)$  first is completely arbitrary, but may lead to a certain bias in smaller populations or when a group is at its lower limit due to the restriction that groups may not go below zero, and that multiple transitions can lead to the deduction of a group for each step. For this, it would perhaps be interesting to investigate how the model might be improved by randomly selecting the order in which the transitions are evaluated.

It would also be interesting to experiment with the be-

haviour of a compound model involving all of the different extensions that we have looked at, described by

$$\begin{aligned} S' &= cR - (A \cos(\omega t) + a_0) \frac{SI}{N} - dS + eN - f \\ I' &= (A \cos(\omega t) + a_0) \frac{SI}{N} - bI - (d + d_I)I \\ R' &= bI - cR - dR + f \end{aligned} \quad (12)$$

Whilst the implementation of this compound model is entirely trivial with a proper understanding of its individual constituents, the analysis of it is likely a much more involved, and out of the scope of this project.

- 
- [1] M. Hjorth-Jensen, “Disease modeling, fys3150 project 5.” (2018).
  - [2] D. C. Lay, *Linear Algebra and Its Applications*, 5th ed. (Pearson, 2016).
  - [3] M. Hjorth-Jensen, *Computational Physics - Lecture notes* (2015).