FYS4150 H20 - Project 5: Solutions to the SIRS Disease Model with Runge-Kutta and Monte Carlo Methods

Olav Fønstelien

December 19, 2020

Abstract

The report suggests two numerical algorithms for solving the SIRS model for the spread of disease in a population. The first model is deterministic and continuous and is based on the Fourth Order Runge-Kutta method; the second model is stochastic and discrete and is based on the Monte Carlo method. The report shows an example where the algorithms find diverging solution to a problem, but explains how both solutions are correct when interpreted into the proper context – the Runge-Kutta algorithm is most suitable for problems on the macro scale; while the Monte Carlo algorithm is most suitable for problems on the micro scale, dealing with small populations. The computational efficiency of both algorithms also suggests this division between their best applications. The report also gives a review of the mathematical basis for each of the algorithms and an outline for their implementation.

The report presents results from solutions to the SIRS model, including vaccination; dynamic populations; and a case study of the 2020 COVID-19 pandemic. The effects of government-imposed *lockdowns* are explored in the context of seasonal variations in transmission rates. The report concludes that strict lockdowns must be implemented at the right time in order to have effect and reduce the reproduction number $R_t < 1$ – that is; they should be implemented when the *transmission* rate increases due to seasonal variations, and not necessarily when the *infection* rate rises. If implemented too late, seasonal variations are the main driver in later infection rate drops.

Please visit my GitHub repository `https://github.com/fonstelien/FYS4150/tree/master/project5` for the source code developed for this report.

1 Introduction

The spread of disease in a population depends on a large number of social and environmental factors spanning from government policies, culture, national economy and health services, to geography, climate and seasonal variations – in addition to the nature of the pathogen itself. The SIRS model lets us model all of these factors by weighing their effect on a disease's *transmission rate* to the susceptible part of the population; the *recovery rate* among the infectious part; and the *immunity loss rate* among the recovering part of the population.

Mathematically, we describe the SIRS model by a set of three coupled differential equations for the rate of change in each part of the population. It does not have an analytical solution, and hence the need for numerical methods. Numerical methods also let us introduce complex time-dependent transition rates, to predict in detail the outcome of the implementation of for instance a government policy, which further expands the SIRS model's potential.

In this report we will develop two algorithms for solving the SIRS model, one deterministic and continuous algorithm based on the Fourth Order Runge-Kutta method; and one stochastic and discrete algorithm based on the Monte Carlo method. We will begin in Section 2 by properly introducing the SIRS model, and see how we can expand it to explicitly include factors like vaccinations, seasonal variations, and dynamic populations. Then, in Section 3 we will review the mathematical basis for each of the methods; first Runge-Kutta in Section 3.1; then Monte Carlo in Section 3.2. We then move on to presenting the simulation results – in all cases presented side-by-side for both algorithms. Firstly, in Section 4.1 we present first the distribution of the Monte Carlo solution to the SIRS model before we investigate the importance of the basic reproduction number R_0 in the spread of a disease. We expand on these results by introducing vaccines in Section 4.2, and see how a given vaccination rate correspond to the distribution of the population as susceptible, infectious or recovering. Then we move on to Section 4.3 where we model vital dynamics and see where the continuous Runge-Kutta and discrete Monte Carlo model may arrive at diverging solutions for the SIRS model – which may both be correct when interpreted into the proper context. In Section 4.4 we model the 2020 COVID-19 pandemic by tweaking the main parameters in the model by introducing government-imposed *lockdowns* and the slow arrival of vaccines. We conclude the report in Section 5 by summing up our findings and giving some advice for further work.

2 Background: The SIRS Model

The classical *SIRS* model describes the spread of disease in a population of size N, with the transitions between the classes of *Susceptible*, S; *Infectious*, I; and *Recovering*, R. The possible transitions of an individual is cyclic;

$$S \to I \to R \to S, \tag{1}$$

and hence the name. The rates of the transition $S \to I$ is dependent of a transition rate a, and the chance $S/N \cdot I/N$ that each of two interacting individuals are in the classes S and I;

$$\text{rate}(S \to I) = asi, \tag{2}$$

where we have let s, i denote the ratios of susceptibles and infectous in the population. Similarly, the rate of transition from class I to R, and R to S are given by the recovery rate b and the immunity loss rate c. These processes happen spontaneously and need no second individual, such that

$$\begin{aligned} \text{rate}(I \to R) &= bi \\ \text{rate}(R \to S) &= cr \end{aligned} \quad , \tag{3}$$

where again r is the ratio of recovering individuals in the population. From this we can set up the coupled differential equations which describe the rate of transition between the classes in the population:

$$\begin{aligned} s' &= \text{rate}(R \to S) - \text{rate}(S \to I) = cr - asi \\ i' &= \text{rate}(S \to I) - \text{rate}(I \to R) = asi - bi \\ r' &= \text{rate}(I \to R) - \text{rate}(R \to S) = bi - cr \end{aligned} \quad . \tag{4}$$

The classical SIRS model reaches steady state when $s' = i' = r' = 0$. Observing that $s + i + r = 1$, we get the steady-state ratios

$$\begin{aligned} s^* &= b/a \\ i^* &= \frac{1 - b/a}{1 + b/c} \\ r^* &= 1 - s^* - i^* \end{aligned} \quad . \tag{5}$$

Here we note that a disease has established itself within a population if $i^* > 0$, meaning that the steady-state ratio of susceptibles $s^* = b/a < 1$. This ratio is otherwise known as the basic reproduction number R_0, defined as the rate of reproduction in a perfectly susceptible population – that is; a population where $s = 1$, except for the initial infected individual. At a later stage, $s < 1$, and the general reproduction number R_t is given by as/b;

$$\begin{aligned} R_0 &= b/a \\ R_t &= as/b \end{aligned} \quad . \tag{6}$$

See [1].

During the outbreak of disease – be it the annual influenza or the black plague – governments and individuals will try to reduce R_t as far as possible or practicable. This can be done only by suppression, i.e. reducing the transmission rate a, since as long as the disease exists, $s^* = b/a$, as we will see next.

If we denote the rate of vaccination within a population f, and limit vaccination to those in the S class only, we can describe the rate transition into R from S as

$$\text{rate}(S \to R) = fs. \tag{7}$$

Introducing vaccination into Equation (4) gives us

$$\begin{aligned} s' &= cr - asi - fs \\ i' &= asi - bi \\ r' &= bi - cr + fs \end{aligned} \quad . \tag{8}$$

Likewise, the steady-state ratios from Equation (5) now become

$$\begin{aligned} s^* &= b/a \\ i^* &= \frac{1 - b/a(1 + f/c)}{1 + b/c} \\ r^* &= 1 - s^* - i^* \end{aligned} \quad , \tag{9}$$

where we again note that a disease needs $b/a < 1$ in order to establish itself, but given a vaccine, the new steady-state infection ratio i^* is reduced due to the f/c term in Equation (9).

Given a disease with know transmission and immunity loss rates, we may predict the needed rate of vaccination f in order to remove the disease from the population – that is; $\mathrm{argmin}_f i^*(f) = 0$. From Equation (9) we see that $f = c(a/b - 1) - i^* a(c/b + 1)$, and $i^* = 0$ is thus achieved when

$$f \geq c(a/b - 1). \tag{10}$$

Likewise, we find how large part of the population to vaccinate – or rather; how large part of the population which has to have antibodies against a pathogen as $i \to i^* = 0$;

$$r \to r^* = 1 - \frac{1}{1 + f/c}. \tag{11}$$

Thus, only after the disease has been eradicated does the $s^* = b/a$ relationship cease to hold. s^* is then given as

$$s^* = \frac{1}{1 + f/c}. \tag{12}$$

At last, we may also introduce vital dynamics – births and deaths – into the SIRS model to be able to study the effect of deadly diseases over longer periods of time. Given a birth rate e, a background death rate d and an incidence death rate d_I, the full form of the SIRS model becomes

$$\begin{aligned} s' &= cr - asi - fs - ds + e \\ i' &= asi - bi - di - d_I i \\ r' &= bi - cr + fs - dr \end{aligned} \quad , \tag{13}$$

where we have assumed that newborns are initially susceptible. The steady-state ratios s^*, i^*, r^* are more complicated to attain, and in this report we will find these with the help of the numerical models which we will developed in the following chapter. This will allow us to introduce time-varying rates, such as transmission rates affected by government policy, or delayed availability of vaccines.

3 Methods

The differential equation for the SIRS model as stated in Equations (4), (8), (13) does not have an analytical solution. In this chapter we will develop two algorithms for solving them; one based on the Fourth order Runge-Kutta method; and one based on the Monte Carlo method.

3.1 Fourth Order Runge-Kutta Algorithm for the SIRS Model

The Fourth Order Runge-Kutta method, or simply RK4, is based on *Simpson's Rule* for numerical integration, by which

$$\int_a^b f(x)dx = \frac{1}{6}\big(f(a) + 4f((b+a)/2) + f(b)\big) + O(h^5), \tag{14}$$

– that is; an approximation of f's slope over $[a, b]$ by the weighted function values at the end points a, b and the midpoint $(a + b)/2$.

Given a first-order differential equation on the form

$$\frac{dy}{dt} = f(t, y), \tag{15}$$

we can use Simpson's Rule to approximate $y(t)$ numerically by y_t as

$$\begin{aligned} y_{i+1} &= y_i + \int_{t_i}^{t_{i+1}} f(t, y)dt \\ &= \frac{h}{6}\big(f(t_i, y_i) + 4f(t_{i+1/2}, y_{i+1/2}) + f(t_{i+1}, y_{i+1})\big) + O(h^5) \end{aligned} \quad , \tag{16}$$

where $h = t_{i+1} - t_i$ is the step size. The midpoint and endpoint values $y_{i+1/2}$ and y_{i+1} are unknown to us. We may approximate $y_{i+1/2}$ as $y^*_{i+1/2}$ by *Euler's Forward Method* with step size $h/2$ such that

$$y_{i+1/2} = y_i + \frac{h}{2}f(t_i, y_i) + O(h^2) \Rightarrow y^*_{i+1/2} = y_i + \frac{h}{2}f(t_i, y_i), \tag{17}$$

but instead of applying this directly into Equation (16), RK4 now splits the middle term in two;

$$4f(t_{i+1/2}, y_{i+1/2}) \to 2f(t_{i+1/2}, y^*_{i+1/2}) + 2f(t_{i+1/2}, y^{**}_{i+1/2}), \tag{18}$$

where $y^{**}_{i+1/2}$ in the second term is an improved approximation of $y_{i+1/2}$ using

$$y^{**}_{i+1/2} = y_i + \frac{h}{2}f(t_{i+1/2}, y^*_{i+1/2}). \tag{19}$$

Following the same pattern, y_{i+1} in the last term is approximated by y^*_{i+1} using $y^{**}_{i+1/2}$ such that

$$y^*_{i+1} = y_i + hf(t_{i+1/2}, y^{**}_{i+1/2}). \tag{20}$$

We end up with an approximation y^{**}_{i+1} for y_{i+1} which is given by

$$y^{**}_{i+1} = y_i + \frac{h}{6}\big(f(t_i, y_i) + 2f(t_i + \frac{h}{2}, y^{*}_{i+1/2}) + 2f(t_i + \frac{h}{2}, y^{**}_{i+1/2}) + f(t_i + h, y^{*}_{i+1})\big). \tag{21}$$

The error is similar to that of Simpson's Rule, $O(h^5)$, which adds up to $O(h^4)$ over the whole run from $t_1 = t_0 + h$ to $t_n = t_0 + nh$. This comes however at the cost of seven evaluations of $f(t, y)$ at each time step, compared with only one for Euler's method, which again has an overall error $O(h)$.

Algorithmically, the steps in RK4 are very simple to implement. An outline is given in Listing 1. The RK4 algorithm runs in $\mathcal{O}(n)$ time and requires no memory of the earlier calculation steps. It is CPU-bound.

Listing 1: Fourth order Runge-Kutta algorithm.

```
ALGORITHM RUNGE-KUTTA(f, ti, yi, h)
  Inputs: function f, time ti, value yi at time ti, step size h.
  Outputs: approximation of y at time ti+h.

  k1 = h*f(ti, yi)
  k2 = h*f(ti+h/2, yi+k1/2)
  k3 = h*f(ti+h/2, yi+k2/2)
  k4 = h*f(ti+h, yi+k3)
  OUTPUT yi + (k1 + 2*k2 + 2*k3 + k4)/6

END ALGORITHM
```

The SIRS model as described in Equation (13) can be implemented directly as it stands using the Runge-Kutta algorithm in Listing 1. We will however add an equation for the population growth rate n', which gives us the following set of equations;

$$\begin{aligned} s' &= cr - asi - fs - ds + e \\ i' &= asi - bi - di - d_I i \\ r' &= bi - cr + fs - dr \\ n' &= e - d - id_I \end{aligned} \quad . \tag{22}$$

We must select a suitable time step h, and update all approximations for each iterations, using only the values from the earlier iteration. However, we might update the last equation, say that for r_i, based on the results of the others, since we need that $r_i = n_i - s_i - i_i$ at all times. This improves numerical stability. A possible implementation is outlined in Listing 2.

Listing 2: SIRS model Runge-Kutta algorithm.

```
ALGORITHM SIRS-RUNGE-KUTTA(fs, fi, fn, t0, s0, i0, n0, n, h)
  Inputs: functions fi, fi, fn, initial time t0,
      initial values s0, i0, n0, time steps n, step size h.
  Outputs: approximation of s, i, r, n at time
       t=t0+h, t0+2*h, ..., t0+n*h
  s = s0
  i = i0
  n = n0
  t = t0
  FOR k IN 1...n DO
    s = RUNGE-KUTTA(fs, t, s, h)
    i = RUNGE-KUTTA(fi, t, i, h)
    n = RUNGE-KUTTA(fn, t, n, h)
    r = n - s - i
    t = t + h
    OUTPUT s, i, r, n
  END FOR

END ALGORITHM
```

Possible extensions to the Fourth Order Runge-Kutta algorithm include the *Runge-Kutta-Fehlberg* algorithm, which applies adaptive time step for error control and optimization. See [2] for more on this issue as well as numerical methods for solving differential equations in general.

3.2 Monte Carlo Algorithm for the SIRS Model

The Runge-Kutta method inherently implicates continuous values for the ratios of susceptibles s, infectious i, and recovering r in the population. The Monte Carlo method, on the other hand, lets us calculate the discrete changes in each class, and thereby simulate a possible trajectory for the spread of the disease and the effect it has on the population.

Before we run the simulation, however, the expectation value at each step in the trajectory is exactly the solution to the differential equation describing the SIRS model;

$$\mathbb{E}(y_i) = y(t_i) \tag{23}$$

The Monte Carlo method will therefore give us an estimate $\bar{\mu}_i$ of the solution if we calculate its mean value. The estimate follows the t-distribution and if we run n simulations it has standard deviation $\bar{\sigma}_i$ given by

$$\bar{\sigma}_i^2 = \frac{1}{n-1}\sum_{j=1}^{n}(y_j - \bar{\mu}_i)^2. \tag{24}$$

From this we can calculate a confidence interval for the *true* $\mu_i = y(t_i)$, i.e. the solution to the equations. A $100 \cdot (1-\alpha)$ % confidence interval is given by

$$\bar{\mu}_i - t_{\alpha/2,\nu}\frac{\sigma}{\sqrt{n}} < y(t_i) < \bar{\mu}_i + t_{\alpha/2,\nu}\frac{\sigma}{\sqrt{n}}, \tag{25}$$

where $\nu = n-1$ denotes the t-distribution's degrees of freedom. See [3]. As $\nu \to \infty$, the t-distribution converges towards the normal distribution, but for small n, i.e in the 10's or 20's, it does matter. A CI_{90} has a multiplier of 1.64 for the normal distribution, while it is 1.83 for the t-distribution with $\nu = 9$. Corresponding values for CI_{99} are 2.58 and 3.25.

We implement the Monte Carlo method by first discretizing the SIRS model in Equation (13);

$$\begin{aligned} S' &= cR - \frac{aSI}{N} - fS - dS + eN \\ I' &= \frac{aSI}{N} - bI - dI - d_I I \\ R' &= bI - cR + fS - dR \end{aligned} \quad , \tag{26}$$

where we have let the upper-case letters S, I, R, N denote discrete variables and their derivatives. Next, we translate the transition rates rate$(X \to Y)$ into probabilities that a transition will happen within a time step Δt of our simulation;

$$\begin{aligned} P(S \to I) &= \frac{aSI}{N}\Delta t \\ P(S \to R) &= fS\Delta t \\ P(S \to D) &= dS\Delta t \\ P(B \to S) &= eN\Delta t \\ P(I \to R) &= bI\Delta t \\ P(I \to D) &= dI\Delta t \\ P(I \to D_I) &= d_I I\Delta t \\ P(R \to S) &= cR\Delta t \\ P(R \to D) &= dR\Delta t \end{aligned} \quad . \tag{27}$$

Here, we have let D denote deaths from other causes than the disease; D_I deaths from the disease; and B denote births, which again are all assumed into class S, i.e. susceptible. Since we cannot have any half deaths or half births, and since at any point in the simulation we do not know the development in the next steps, we must adapt Δt such that at any time at most one individual is likely to transition from one class to another. To find Δt, observe that for all but the new infections, the highest probability for the transition out of one class X into another class Y occurs when all N individuals are in X. For new infections we have the highest rate when exactly half of the population are in each class S, I. Consequently, the time step Δt must satisfy

$$\begin{aligned} \mathrm{argmax}_X P(X \to Y) &= qN\Delta t \leq 1 \\ \mathrm{argmax}_{S,I} P(S \to I) &= \frac{aN}{4}\Delta t \leq 1 \end{aligned} \quad . \tag{28}$$

Remembering that we may also have time-varying transition rates, such as seasonal vaccination or policy-suppressed transmission rates, we must adapt the time step at each iteration in the simulation such that

$$\Delta t_i = \frac{1}{N} \cdot \min\left(\frac{4}{a}, \frac{1}{b}, \frac{1}{c}, \frac{1}{d}, \frac{1}{d_I}, \frac{1}{e}, \frac{1}{f}\right). \tag{29}$$

Now, having decided on the time step for iteration i, we draw a random number $\varepsilon_i \sim \mathcal{U}(0,1)$, and transition one individual from class X to Y if $\varepsilon_i < P_i(X \rightarrow Y)$. Since all transitions *within one iteration* of the simulation are independent, we may reuse ε_i when we do all the transition tests in that iteration.

An outline of the resulting SIRS model Monte Carlo algorithm is given in Listing 3. As for the Runge-Kutta algorithm in Listing 2 we update `R` by summing over the other classes, both for efficiency and for numerical stability.

Listing 3: SIRS model Monte Carlo algorithm.

```
ALGORITHM SIRS-MONTE-CARLO()
  Inputs: initial population distribution S0, I0, R0, N0,
  time-varying transition rates a, b, c, d, dI, e, f,
  initial time t0, end time tn.
  Outputs: approximation of S, I, R, N at time
  t1=t0+dt1, t2=t0+dt2, ..., tn.

  S = S0
  I = I0
  R = R0
  N = N0
  t = t0
  WHILE t <= tn DO
    dt = get_time_step(N, t) // (Equation (29))
    epsilon = draw_random_number()
    {collect incremental transitions from
     all combinations in Equation (27).
     Make all transitions where probability < epsilon}
    S = S + {transitions into class S}
    I = I + {transitions into class I}
    N = N + {sum of births and deaths}
    R = N - S - I

    OUTPUT S, I, R, N, t
    t = t + dt
  END WHILE

END ALGORITHM
```

Even if the number of steps can not be determined before the algorithm is run, the Monte Carlo SIRS model algorithm runs in linear time as long as we increase the problem along one axis only – either the population size N; the simulation time t_{sim}; or the number of runs n. Consequently, time is given as $\mathcal{O}(nNt_{sim})$. Monte Carlo is CPU bound and requires no memory of the earlier calculation steps.

4 Results

We will now go through the results and see how our two algorithms for solving the SIRS model perform. We will in all cases present corresponding results for both models side-by-side.

The results are structured in the order of increasing complexity. We begin in Section 4.1 by examining the importance of the ratio of the recovery and transmission rates in the establishment of a disease within a population. We saw this as the basic reproduction number R_0 in Section 2. After looking at these static conditions, we will see how seasonal variations affects the dynamics of the disease over the year.

In Section 4.2 we will introduce the effect of a vaccine and further study its effect on the spread of a disease. We will see how the vaccination rate must be adjusted for the disease to gradually disappear, or even be kept at a constant level within the population, which is the case with many diseases. We will introduce seasonal variations here as well, and see how the dynamics that this introduces very much resembles the seasonal transmission rate variations.

After that, in Section 4.3, we will perform a longer simulation to see how infection levels of a deadly disease affects the long-term growth of a population with a fixed birth and background death rate.

At last we will bring it all together when we try to reproduce the 2020 COVID-19 pandemic, with corresponding government policies, transmission rate seasonality and – looking into the future – the advent of an efficient vaccine.

Note that unless otherwise is stated, the disease parameters for transmission rate, recovery rate and immunity loss rate can be assumed to be $a = 4, b = 1, c = 0.5$. For the Monte Carlo simulations, population size can be assumed to be $N = 400$ and initial number of infectious individuals $I = 100$.

4.1 The Importance of R_0

As we saw in Section 2, Equation (6), success or failure on a infectious disease's term depends on its basic reproduction number R_0, which is given by its recovery vs. transmission ratio a/b. Figure 1 shows Runge-Kutta (RK4) and Monte Carlo (MC) simulations for four different ratios. We see that gradually increasing the recovery rate also gradually decreases the steady-state infectious fraction i^*, as we would expect from Equation (5);

$$i^* = \frac{1 - b/4}{1 + 2b}. \tag{30}$$

For $b = 1$, the disease establishes itself in 25 % of the population at any given time; doubling the recovery rate reduces this to 10 %. At $b = 3$, a mere 3.5 % remain infected, while at $b = 4$, corresponding to exactly $i^* = 0$, the disease gradually disappears without ever getting a foothold.

With only 400 individuals in the population simulated by the MC algorithm in Figure 1, the system still seems very deterministic. We see that only in the most dynamical periods do we have a noticeable shade around the curves,

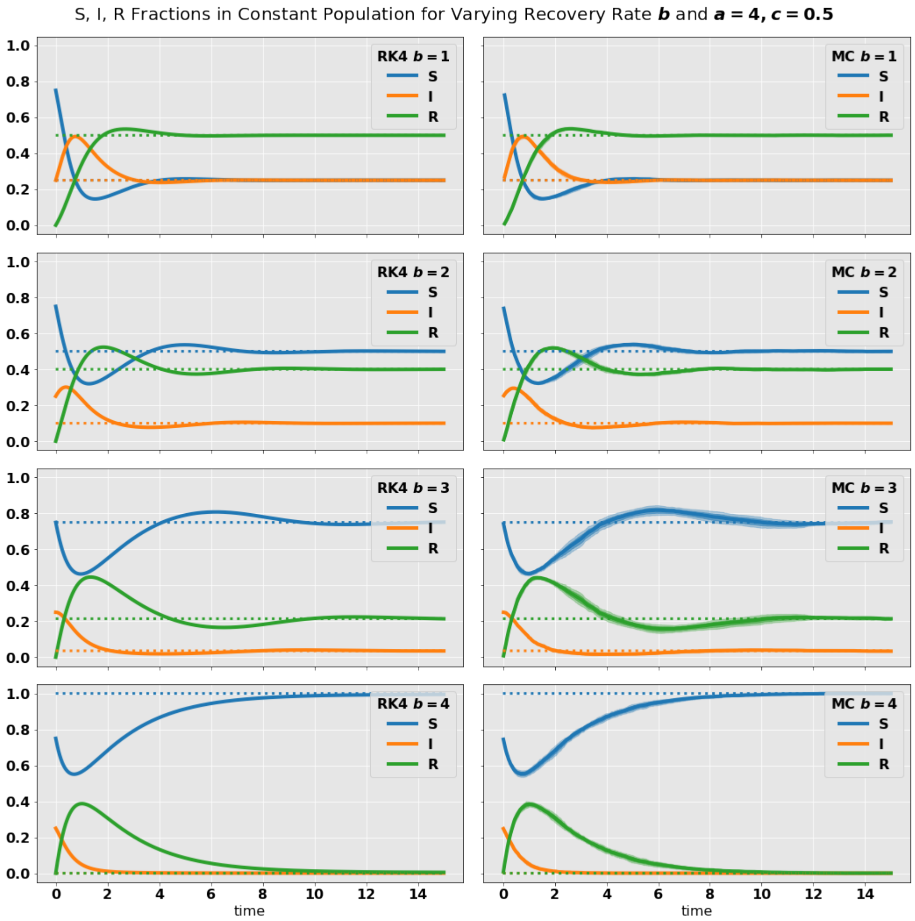

Figure 1: Runge-Kutta (RK4) simulations in the left column and Monte Carlo (MC) in the right for a population of size 400 with initially 100 infected individuals. Results from both algorithms largely overlap and converge towards the expected values (dotted same color). MCs are mean of 10 simulations with CI_{99} bands indicated as same-color shades. We see that the CI band is wider for the areas with high dynamic.

denoting the CI_{99} interval. This impression is also largely confirmed when we look at Figure 2. We see the distribution for each of the curves S, I, R in the $b = 3$ case, sampled since steady-state at $t = 10$ until $t = 10^4$. The distribution closely resemble the anticipated normal distribution for all populations, but the distribution is markedly sharper for I. This is given by Equation (27) in that a deviation from the expectation value μ has a more forceful pull back towards μ for I than for the other. From Equation (29) we see that $a = 4, b = 3, c = 0.5$

give the transition probability deltas

$$\begin{aligned}\Delta P(S \to I) &= \frac{3 \cdot (299 \cdot 15 - 300 \cdot 14)}{400} \cdot \Delta t = 2.1375\Delta t \\ \Delta P(I \to R) &= 3 \cdot (13 - 14)\Delta t = -3\Delta t \\ \Delta P(R \to S) &= 0.5 \cdot (85 - 86)\Delta t = -0.5\Delta t\end{aligned} \quad . \tag{31}$$

Consequently, we will have a larger variation in S and R populations, while any deviation from the expectation value in the I population will quickly be sent on to R.

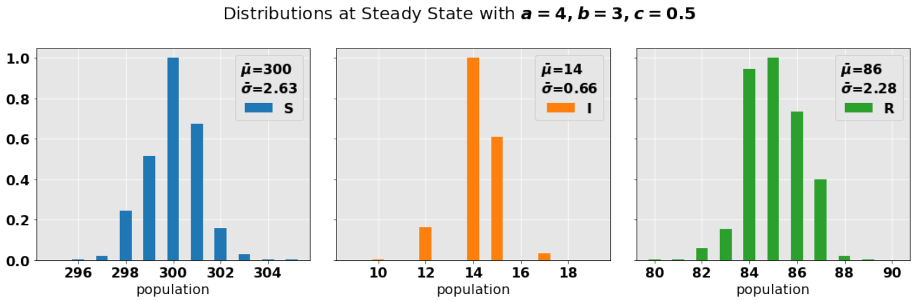

Figure 2: Population distribution for steady-state Monte Carlo simulation for the $b = 3$ case in Figure 1. Population size is 400 and simulation. The distributions largely follow the normal distribution.

By introducing seasonality in a, we can simulate the dynamics of the disease over the year. In Figure 3 we see the $b = 1$ case from earlier with seasonal transmission rate as the dotted grey line. The rate has a peak at $t = 0$, and a period $T = 10$. The resulting pattern of infection closely follows the infection rate phase, but the shape is different. Due to the degree of immunity, as well as infection, and hence decrease of available susceptible individuals, the infection curve quickly rises before it slowly decreases and hits the bottom in the middle of the year.

Lastly, we note that even though both the Runge-Kutta and the Monte Carlo method run in linear time, Runge-Kutta is much more efficient in solving the SIRS model. Number of iterations n can be selected freely, and the algorithm gives good results for as little as 50-100 steps per time unit. The Monte Carlo method, on the other hand, runs much slower both due to the re-sampling of the time steps, and due to the requirement of smaller time steps for a population of 400 with these disease parameters.

4.2 Introducing Vaccines

In Section 2 we saw that, given a disease with parameters a, b, c, the level of infection can be controlled by vaccinating a corresponding portion of the pop-

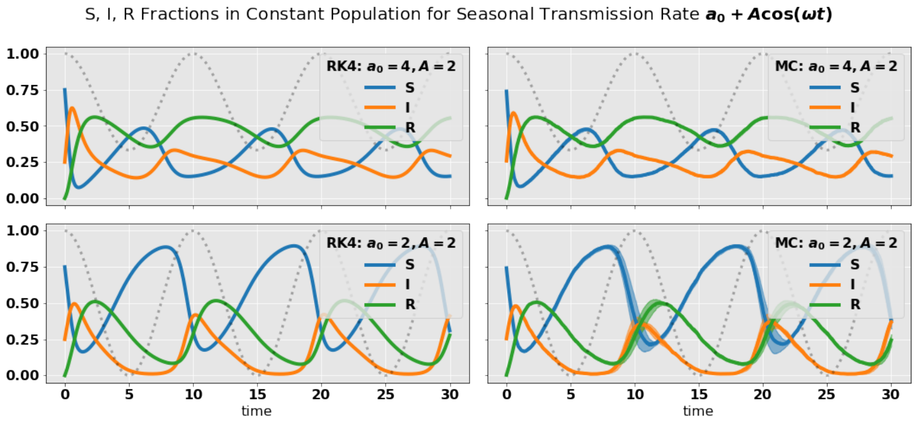

Figure 3: RK4 (left) and MC (right) simulation of seasonal variations in the transmission rate with $b = 1, c = 0.5$. The infection rate curve, shown as a dotted gray line, is closely followed by the infection rate.

ulation given by Equation (11). To completely get rid of the disease, we must have a rate of vaccination $f \geq c(a/b - 1)$.

In Figure 4 we see the effect of using vaccines against a disease with parameters $a = 4, b = 1, c = 0.5$. We see that when we increase the level of immunity in the population from 50 %, which is the level without vaccines, to 65 % we decrease the infection level by the same 15 %, from 25 % to 10 %. Further increasing the immunity to 75 % removes it all together. Thus, we see that by *moving* individuals from the S class into R class, we are in reality *removing* them from the I class. This corresponds to what we found in Equation (9).

Similar to what we saw for seasonal variations in the transmission rate in the former section, a seasonal vaccination rate provokes an annual ebb and flow in the infection rates, as we see in Figure 5. We note, however, that even if we on average apply the vaccination rate which would eventually remove the disease in the steady-state case, the disease remains in the seasonal case.

4.3 Introducing Vital Dynamics

Let us now study the effect of a disease on the long-term development of the population by introducing an incidence death rate d_I from the disease, along with a background death rate d and birth rate e. We assume a constant rates $d = 0.01$ and $e = 0.02$, and apply various d_I for a disease with otherwise fixed, constant parameters $a = 4, b = 1, c = 0.5$. In Figure 6 we see that as for the constant population cases in Section 4.1, the fraction of infectious individuals finds a steady state. The level does not correspond to the levels we saw earlier, though, due to the imbalance introduced by d, e, d_I in the steady-state equations. In the figure we see that the Monte Carlo and Runge-Kutta results correspond quite well, except for the $d_I = 1.00$ case, where Runge-Kutta

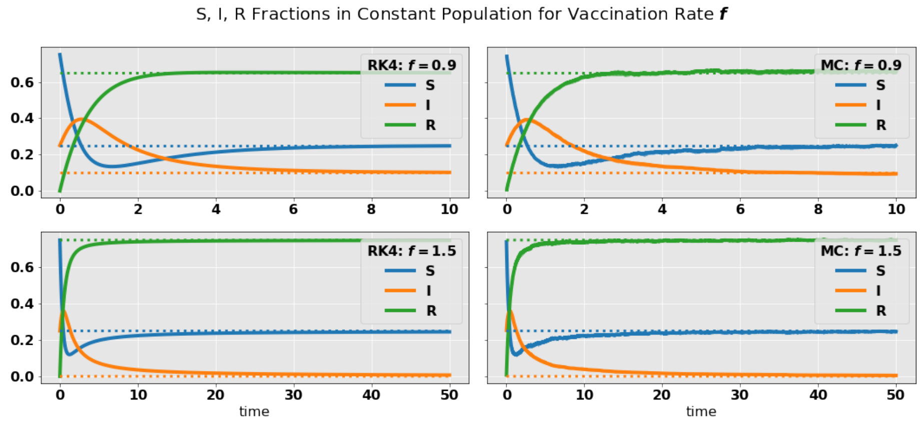

Figure 4: RK4 (left) and MC (right) simulation of the effect of vaccination in a population. Disease parameters are $a = 4, b = 1, c = 0.5$, and for MCs, $N = 400$. Vaccination removes individuals from the I class.

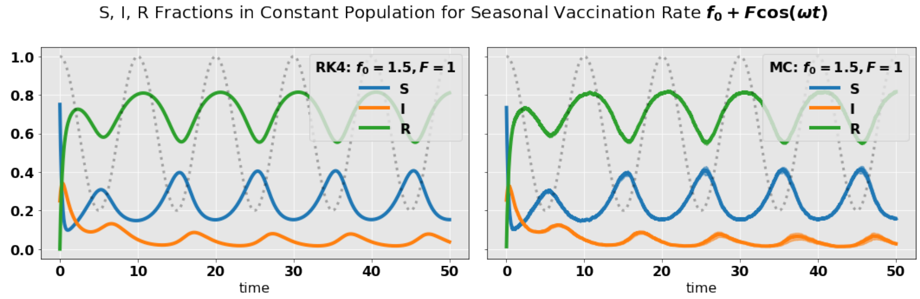

Figure 5: RK4 (left) and MC (right) simulation of the effect of seasonal vaccination in a population. Vaccination rate is indicated by the dotted gray line and has a peak at $t = 0$ and period $T = 10$. As earlier, disease parameters are $a = 4, b = 1, c = 0.5$, and for MCs, $N = 400$. We see that seasonality in vaccination rates translates into seasonality in infections.

presents a constant fraction, while Monte Carlo predicts a collapse, and hence eradication of the disease, at around $t = 25$. The simulation has been run 30 times, and the CI_{99} follows the mean towards zero.

The explanation for the collapse in infections predicted by Monte Carlo becomes clear when we look at the development of the populations. In the $d_I = 1.00$ case, the population is all but extinguished, and it is in this scenario that Monte Carlo predicts the last infected individual to either recover or die, while there still may be a small population left which then starts to grow ever so slowly. This outcome is not possible in a Runge-Kutta simulation since the populations are only represented by continuous fractions, which would not go

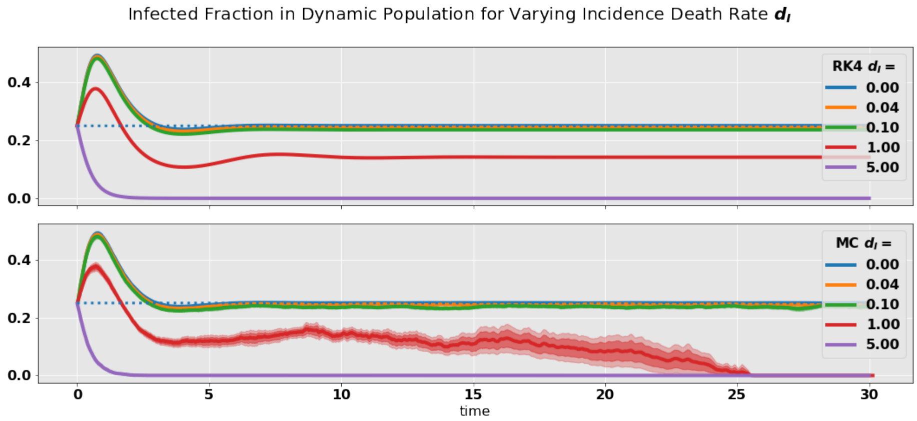

Figure 6: RK4 (left) and MC (right) simulation of the effect vital dynamics and incidence death rate f_I on a population. RK4 finds a steady state for any d_I, as does MC, with one exception. For $d_I = 1.00$, the RK4 and MC simulations deviate drastically. The explanation is found in Figure 7.

to zero for these disease parameters.

The small population at the end of the $d_I = 1.00$ case also explains the high variance in the mean value, since it reflects incremental changes on a small scale. Increasing the population size N to 2000 makes the CI$_{99}$ narrower, and this time the disease does not vanish, so there is some threshold for this given the parameters (result not shown).

Another interesting case is the $d_I = 5.00$, which represents a disease so deadly that $R_0 < 1$, and hence the disease disappears by itself. For $d_I = 0.04$, the population size remains constant, meaning that the birth surplus $e - d = 0.01$ correspond exactly to the deaths $d_I i = 0.04 \cdot 0.25 = 0.01$.

4.4 Bringing it All Together: COVID-19 Case Study

We will now combine what we have learned from looking at the seasonality of vaccines and transmission rates in the above results with government policies and non-availability of vaccines, using the COVID-19 pandemic as a case study [4]. The science on this disease is still uncertain, but we will assume a strong seasonal factor in its transmission rate which peaks at new year; $a(t) = 3 + 2.25\cos(5\pi t)$, where one year is 10 time units. We will also assume an influenza-like vaccination seasonality once that gets going, with a peak during the autumn months, such that $f(t) = 1 + 0.5\cos(5\pi(t + 2.5))$. Further, and critically, we will assume government-imposed *social lockdowns* put into force at different times during the onset of the pandemic and see how that affects the outcome, along with the effect of the *strictness* of those lockdowns. Other parameters are kept as before; $b = 1, c = 0.5, N = 400$, but we assume $I = 20$ or 5 % at the onset.

Even if COVID-19 is a deadly disease, potentially, the incident death rate is

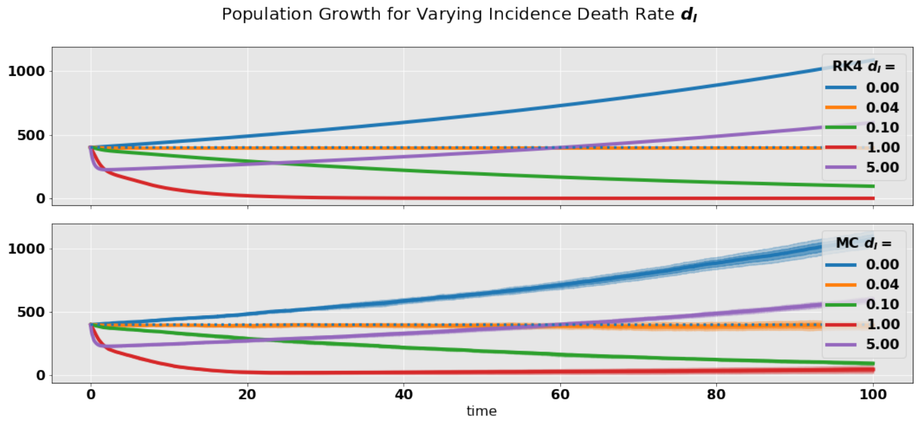

Figure 7: RK4 (left) and MC (right) simulation of the effect of incidence death rate d_I on the population. Constant population is achieved when $e - d = d_I i$, as we see for $d_I = 0.04$. For $d_I = 1.00$, the MC simulations consistently predict eradication of the disease, something which is not possible for RK4.

still low enough for that we can ignore it and set all vitality rates to zero.

In Figure 8 we see in the upper pane the first COVID-19 lockdown as a bathtub shaped gray-dotted line, and the resulting transmission rate $a(t)$ for the cases of ’immediate’ lockdown ($t = 0$); ’early’ lockdown ($t = 0.5$); ’late’ lockdown ($t = 1$); and ’none’. Each lockdown has the same length $\Delta t = 5$ as well as the same strictness, meaning that they reduce the seasonal infection rate by 90 %, but as explained are implemented at different times.

In the middle pane, the first lockdown is ’immediate’ in all cases ($t = 0$), and with a new lockdown implemented in early autumn at $t = 7.5$ with duration $\Delta t = 7$. This time the lockdowns have different strictness ranging from ’strict’ (-90 %); ’medium’ (-70 %); ’loose’ (-20 %); and ’none’.

Finally, in the lower pane we see the vaccination rate. The assumption here is that once this becomes widely available at $t = 15$, it follows a seasonal pattern.

Figure 9 shows the result of each of the lockdown policies in Figure 8, simulated with Runge-Kutta and Monte Carlo. We see in the upper panes that the only policy which has a considerable effect is an ’immediate’ implementation. This limits the peak to about 20 %, from an initial infection ratio of 5 %. Later implementation has little to no effect, since the transmission rate – which is the main driver – is already coming down fast. All cases reach 40-50 %. We see, however, that the later lockdowns have the same effect as the immediate towards the summer, when the infection has all but vanished and restrictions are relaxed.

In the lower pane the course of the pandemic over the next few of years is simulated. First, we see that if only ’loose’ lockdown policies are implemented, they have almost no effect at all. However, ’strict’ policies will not reduce the infections to less than some 20 % in the first wave, and 10 % in the second.

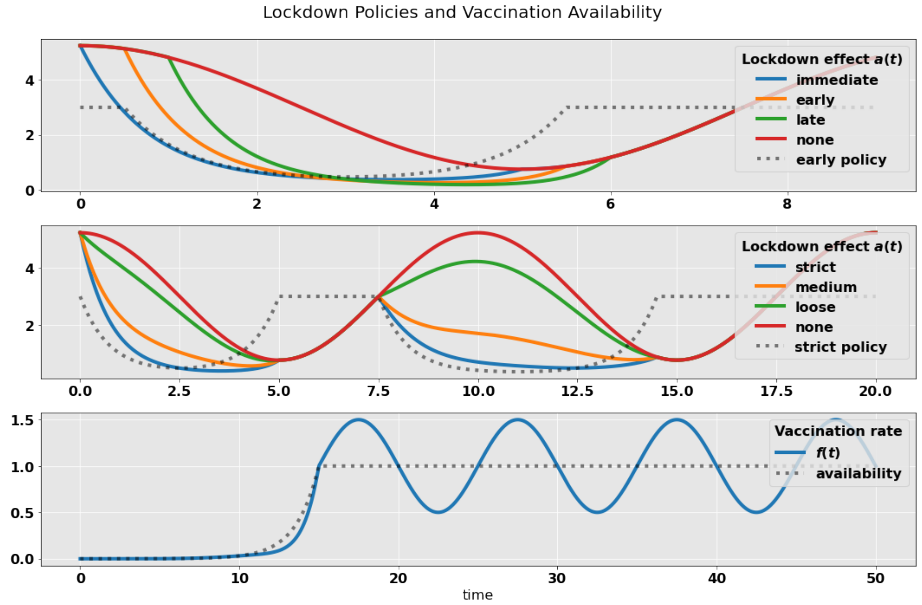

Figure 8: COVID-19 policies and vaccines. The upper two panes show the effect of government-imposed lockdowns (gray-dotted) on the seasonal transmission rate $a(t)$. We see the effect of lockdown timing (middle) and lockdown strictness (middle). In the lower pane, the assumed seasonality of the vaccination is shown, but with a delayed onset.

Measured towards this, the 'medium' policy performs relatively well at 30 and 20 %. The later trajectories are less dependent on the policy decisions of the first year of the pandemic. As soon as the vaccines are introduced, which happens slowly during the first half of the second year, the disease is predicted to gradually vanish, luckily.

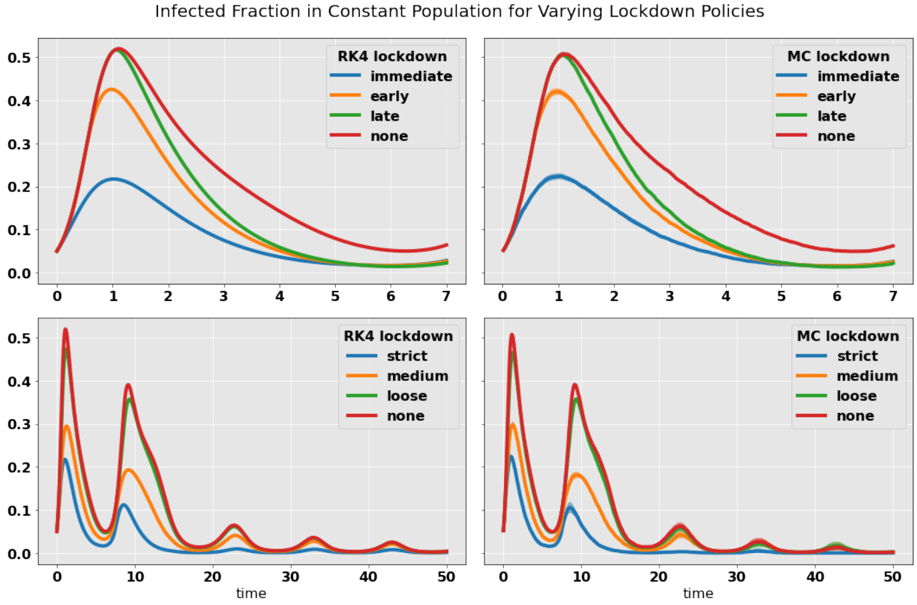

Figure 9: The effects of the timing and strictness of government-imposed lockdowns, and the advent of vaccines at around $t = 15$. The upper pane shows the effect of timing of a 'strict' lockdown the on the infection ratio. The lower pane shows the effect of the strictness of 'immediate' lockdown at onset, and in the autumn of the first year. Finally, vaccines are introduced gradually with full effect around $t = 15$. This causes the disease to vanish in the following years.

5 Discussion and Conclusion

We have seen that two widely different methods, the Fourth Order Runge-Kutta and the Monte Carlo, each solves the set of differential equations describing the SIRS model. RK4 is deterministic and the accuracy of its solution is determined only by the step size h. MC, on the other hand, is stochastic and gives an estimate $\bar{\mu}_i$ for each function value $y(t_i)$. Consequently, the methods may give different solutions, especially when applied on a small, discrete problem, such as the SIRS model with $N = 400$. We saw that in one case, MC would predict with high confidence that a disease would vanish from the population, while RK4 continued to keep a fixed fraction of the population in the I class. Both of these solutions are valid, but valid for a somewhat different problem: The continuous model in the RK4 case; and the discrete model in the MC case.

As the population increase, the differences between the two methods decrease. For large populations, a continuous model of the problem will in many cases serve us best, since MC runs in $\mathcal{O}(N n_{sim} t_{sim})$ time, where N is the population, n_{sim} is the number of simulations to estimate $y(t_i)$, and t_{sim} is the simulation time. In solving the SIRS model, this has tended to be slower than the RK4, which runs in $\mathcal{O}(n)$ time.

Relating to the SIRS model, we have reproduced known benchmarks of disease modeling such as the importance of the basic reproduction number $R_0 = b/a$ (recovery vs. transmission rates); the relationship between disease the parameters a, b and the immunity loss rate c and vaccination rate f; and the somewhat surprising fact that vaccination empties the I class rather than the S class. Also, when performing the COVID-19 case study (with given assumptions), we have seen the importance of timing when it comes to government-imposed lockdown – a strict lockdown implemented too late has nearly no effect, while a lighter lockdown implemented at the right time has a significant effect. The underlying pattern here is that when faced with a pathogen for which there is no vaccine available and which presents a strong seasonal transmission rate a, the only option for forcing $R_t = as/b < 1$ is to enforce policies to reduce a sufficiently during the peaks, and then relax the policies when circumstances are more beneficial. We also saw that as soon as a vaccine becomes readily available, the disease can be suppressed until eradication.

For future work relating to disease modeling, it would be interesting to investigate how the Ising model, or even the related Potts model, could be applied, and to compare the results against the benchmarks of this report [2]. Combining them could even let us introduce *locality* into our model – creating clusters of high and low densities of the classes S, I, R in the population. This could be used for further policy planning with regards to restricting person-to-person interactions, or even the distribution of vaccines. It would also be interesting to find similar cases in the social studies, where a continuous model like RK4 would be applied on the macro scale, while a discrete model like the Monte Carlo model which we have developed here would be used on the micro scale.

References

[1] Gregg N Milligan and Alan D Barrett. *Vaccinology: An Essential Guide.* Wiley Online Library, 2015.

[2] Morten Hjort-Jensen. Computational Physics, Lecture notes Fall 2015. `https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/\Lectures/lectures2015.pdf`, [Online; accessed 20-October-2020].

[3] Jay L Devore and Kenneth N Berk. *Modern mathematical statistics with applications, second edition.* Springer, 2012.

[4] Coronavirus disease (COVID-19) pandemic World Health Organization. `https://www.who.int/emergencies/diseases/novel-coronavirus-2019`, [Online; accessed 18-December-2020].