Numerical Studies of SIRS Model

Simulation of the spread of an infectious disease according to the fourth order Runge Kutta method and Monte Carlo method

Elias Tidemand Ruud, Bendik Nyheim & Mira Mors

Computational Physics I FYS3150/FYS4150

Department of Physics
University of Oslo
Norway
December 2020

Abstract

The spread of an infectious disease was simulated by first applying the simple SIRS model to four isolated populations (A, B, C, D) that had different rates of recovery. In addition, the model was extended by adding vital dynamics, a seasonally varying rate of transmission and by the possibility of vaccination. The fourth-order Runge Kutta method (RK4) and the Monte Carlo method (MC) were used to solve the differential equations. RK4, with $\Delta t = 0.0025$, and MC, with 1000 cycles, yield the same results for group A and D that had a low and high recovery rate respectively. In general, the values computed by RK4 proved to be in better agreement with the analytic expectation values. The relative error was of magnitude less than or equal to 10^{-3}. It was further shown that a low death rate is favorable when the recovery rate is high, but when the recovery rate is low, a high death rate saves more lives in the long run. To effectively prevent the spread of a disease, vaccination should occur before the rate of transmission reaches its predicted maximum.

Contents

1 Introduction

Epidemics have always posted a threat to humanity. Even if medical care through e.g. vaccinations and antibiotics, as well as hygienic conditions, have improved significantly within the last century, we are still at risk to an outbreak of an epidemic. Not only can there be significant economic damage, but millions of people may lose their life. Antibiotic resistance and increasing vaccine fatigue are in addition counter-productive. In view of this, mathematical models and their computer simulations serve as important tools to describe epidemic courses or to forecast them.
In this report, we model the spread of an infectious disease. The simulation is based on the SIRS model, which uses coupled differential equations to simulate the number of susceptibles, infected and recovered. Ordinary differential equations appear in all types of physical problems and by discretizing the continuous equations, we can solve them numerically.
After the basic SIRS model is presented in the method section, the model is extended by vital dynamics. It takes into account death and birth rates and thus allows to simulate the spread of diseases that occur over longer stretches of time. Additionally, a seasonally varying disease and possibility of getting vaccinated improve the model as well. Next, both a deterministic approach, using the fourth-order Runge Kutta method, and the Monte Carlo method are introduced to solve the differential equations. Then the methods are applied to the model. We shall for instance look at four population groups, that have different recovery rates. The obtained results are thereafter discussed and the findings are concluded.

2 Methods

2.1 SIRS model

All following explanations and implementations are based on Ref. [4].
The SIRS model simulates the spread of an infectious disease throughout a given population over time. This allows predictions to be made about the risk of infection and whether the disease will eventually be eradicated.
Given an isolated population of N individuals, the SIRS model divides the population into three separate groups. Those who are *susceptible*, (S), are without immunity to the disease. Those *infected*, (I), are currently ill with the disease and those recently infected have *recovered*, (R), and developed an immunity to the disease.
As the letters of the model specify, a person's state of health can only evolve cyclically from $S \to I \to R \to S$. The number of people moving from one status to another is given by the rate of *transmission*, a, the rate of *recovery*, b, and the rate of *immunity loss*, r. The rates have units of inverse time. As the population is assumed to remain constant and mix homogeneously, the population size at any time, t, reads

$$N = S(t) + I(t) + R(t). \tag{1}$$

In other words, it is assumed that the epidemic prevails for a time scale which is much smaller than the average person's lifetime. Therefore, variations in populations size due to a birth and death rate are ignored.

2.1.1 Set of coupled ODE - simple model

The SIRS model is based on a set of coupled differential equations. The change for each of the three statuses is given by

$$S' = cR - \frac{aSI}{N}, \tag{2}$$

$$I' = \frac{aSI}{N} - bI, \tag{3}$$

$$R' = bI - cR. \tag{4}$$

The fraction, $\frac{aSI}{N}$, in Equation (2) may be rewritten to aSI for small populations. This is due to the number of susceptibles becoming infected depend more on the absolute number of infected people rather than the infected fraction of the population.
Since the population is considered constant, see Equation (1), the three dimensional system (Equation (2) to (4)) can be reduced to a two dimensional one by replacing R with $N - S - I$. We get

$$S' = c(N - S - I) - \frac{aSI}{N} \tag{5}$$

$$I' = \frac{aSI}{N} - bI. \tag{6}$$

Unlike the closely related SIR model, the SIRS model cannot be solved analytically. However, the steady state can be found by setting Equation (5) and Equation (6) equal to zero. At equilibrium, the fractions of people in each status are

$$s^* = \frac{b}{a}, \tag{7}$$

$$i^* = \frac{1 - \frac{b}{a}}{1 + \frac{b}{c}}, \tag{8}$$

$$r^* = \frac{b}{c}\frac{1 - \frac{b}{a}}{1 + \frac{b}{c}}. \tag{9}$$

Note that each fraction must be a number between 0 and 1 with their sum adding up to 1. Apparent from the above equations, the rate of transmission, a, must be larger than the rate of recovery, b, in order for the number of infected people at equilibrium to be large than zero. Thus, the disease only becomes entrenched in the population if $b < a$.

2.1.2 Initial conditions

Table 1 lists a set of possible parameters for four different populations and Table 2 lists the respective expected equilibrium values calculated with Equations (7) to (9).

Table 1: Possible set of parameters for four different populations. a, b, c represent the transition rates (transmission, recovery and immunity loss) of the SIRS model for population A, B, C, D.

Rate $[\frac{1}{\text{time}}]$	Group A	Group B	Group C	Group D
a	4	4	4	4
b	1	2	3	4
c	0.5	0.5	0.5	0.5

Table 2: Expected equilibrium values for the number of suseptibles (s^*), infected (i^*) and recovered (r^*) for the four different population-groups defined in Table 1

Rate $[\frac{1}{\text{time}}]$	Group A	Group B	Group C	Group D
s^*	1/4	1/2	3/4	1
i^*	1/4	1/10	1/28	0
r^*	1/2	2/5	3/14	0

2.1.3 Vital dynamics

Until now, it was assumed that the population was constant in time. In order to extend the model, vital dynamics can be added, which allows to describe the spread of diseases that occur over longer stretches of time. Let e and d represent the birth and death rate respectively and let d_I stand for the death rate of infected people due to the disease. Equations (2) to (4) can be rewritten to

$$S' = cR - \frac{aSI}{N} - dS + eN \tag{10}$$

$$I' = \frac{aSI}{N} - bI - dI - d_I I \tag{11}$$

$$R' = bI - cR - dR. \tag{12}$$

Note that all newborn babies are assumed to be initially susceptible.

2.1.4 Seasonal variation

Some diseases, like the influenza, are more common during the colder months where individuals keep less social distance. Thus, the rate of transmission is dependent on the time of year. Its oscillation can be modeled by

$$a(t) = Acos(\omega t) + a_0, \tag{13}$$

where a_0 is the average transmission rate, A is the maximum deviation from a_0, and ω is the frequency of oscillation.

2.1.5 Vaccination

If a disease can be prevented with vaccination, susceptible individuals can directly move from S to R. Thus, the cyclic structure of the SIRS model is broken. It is assumed that the rate of vaccination, f,

may depend on time. These variations may be due to increased awareness and medical research or simply due to the time of year. Again, Equations (2) to (4) can be rewritten to

$$S' = cR - \frac{aSI}{N} - f \tag{14}$$

$$I' = \frac{aSI}{N} - bI \tag{15}$$

$$R' = bI - cR + f. \tag{16}$$

We look at the impact of vaccination with a constant value and varying value for f. When f varies with time, we model it the same way as we did with the rate of transmission a, thus

$$f = F\cos(\omega t) + f_0, \tag{17}$$

where f_0 is the average vaccination rate, F is the maximum deviation from f_0, and ω is the frequency of oscillation. For this study, we will also investigate the impact of vaccinating heavily before an infection wave, instead of during. This is done by adding a phase shift $\phi = \pi$ to either a or f, such that one parameters is at a maximum when the other is at a minimum.

2.2 Solving the SIRS model

In the following, the fourth-order Runge Kutta method and the Monte Carlo method for solving the simple SIRS model are presented.

2.2.1 Runge Kutta methods

Information obtained from Ref. [3].
The Runge Kutta (RK) methods are a family of iterative methods that are based on Taylor expansion fromulae. More precise, the solution to the future time step is extrapolated by judiciously using the information on the "slope" at more than one point, but the methods are still one-step methods. The classical, fourth order Runge Kutta method, refferred to as RK4, is the most popular which shall also be used for our analysis. The key concept of the methods is to compute an intermediate step between the function value y_i and y_{i+1}.
Consider a function, f, and its integrand, y, defined as

$$\frac{dy}{dt} = f(t,y), \qquad y(t_0) = y_0, \tag{18}$$

$$y(t) = \int f(t,y)dt, \tag{19}$$

where t represents the time. We like to solve the differential equation for a given time period: t_0 to t_{end}. Setting $t_0 = 0$, the step size, h, is then given by

$$h = \frac{t_{end} - t_0}{L} = \frac{t_{end}}{L}, \tag{20}$$

where L is the number of integration points. Thus, the time, t, is discretized by

$$t_i = i\Delta h, \qquad i = 0, 1, \ldots, L. \tag{21}$$

Making use of Equations (18), (19) and the step size h, $y_{i+h} = y_{i+1}$ can be obtained by the next value of the function y

$$y_{i+1} = y_{t_{i+1}} = y_i + \int_{t_i}^{t_{i+1}} f(t, y)dt. \tag{22}$$

The order, p, of the repsective RK method stems from the Taylor series of the search solution function, that is y, which is truncated after the term with h^p. We remind that the Taylor approximation for y_{i+1} is given by

$$y_{i+1} = y(t_i + h) = \sum_{\nu=0}^{p} \frac{h^\nu}{\nu!} \left[\frac{d^\nu}{dt^\nu} y_i(t)\right], \tag{23}$$

with the truncation error

$$R_{i+1} = \frac{h^{p+1}}{(p+1)!} \left[\frac{d^{p+1}}{dt^{p+1}} y_i(t)\right]. \tag{24}$$

Second-order Runge Kutta For demonstration purposes, we will, to begin with, look at the second-order RK method. By Taylor expanding $f(t, y)$ around the center of the integration interval $t \in [t_i, t_{i+1}]$, that is at $t_i + h/2$, we get the first approximation. Then, the midpoint formula is applied to the integrand at $y(t_i + h/2) = y(t_{i+1/2})$

$$\int_{t_i}^{t_{i+1}} f(t, y)dt \approx hf(t_{i+1/2}, y_{i+1/2}) + O(h^3). \tag{25}$$

Hence, the next function value y_{i+1} is given by

$$y_{i+1} = y_i + hf(t_{i+1/2}, y_{i+1/2}) + O(h^3). \tag{26}$$

Since $y_{i+1/2}$ is not known, the value is approximated using Euler's method (see Sec. 6.1.1)

$$y_{i+1/2} = y_i + \frac{h}{2}\frac{dy}{dt} = y(t_i) + \frac{h}{2} f(t_i, y_i). \tag{27}$$

According to the second-order RK method, the sought value, y_{i+1}, is given by

$$k_1 = hf(t_i, y_i), \tag{28}$$

$$k_2 = hf(t_{i+1/2}, y_i + k_1/2), \tag{29}$$

$$y_{i+1} \approx y_i + k_2 + O(h^3), \tag{30}$$

where k_1 is the increment based on the slope at the beginning o the interval, using y. Similarly, k_2 is the increment based on the slope at the midpoint of the interval, using $y + hk_1/2$.
Whereas the Euler method has a global error of $O(h)$, the second-order RK method has one of $O(h^2)$. This is due to the additional evaluated intermediate step at $t_i + h/2$. Thus, the cost for a better stability is more calculations.

Fourth-order Runge Kutta The fourth-order RK method has an global error of $O(h^4)$. This is done by first calculating k_1 and k_2 as in Equations (28) and (29). Additionally, k_3 is again the increment based on the slope at the midpoint, using $hk_2/2$ and k_4 is the increment at the end of the interval, using $y + hk_3$. We get

$$k_1 = hf(t_i, y_i), \tag{31}$$

$$k_2 = hf(t_{i+1/2}, y_i + k_1/2), \tag{32}$$

$$k_3 = hf(t_{i+1/2}, y_i + k_2/2), \tag{33}$$

$$k_4 = hf(t_{i+1}, y_i + k_3), \tag{34}$$

$$y_{i+1} \approx y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)O(h^5), \tag{35}$$

Note that the RK methods are explicit techniques. For our case, f represents S' and I' given in Equations (5) and (6) respectively. Thus, y stands for S, I and R. As the number of susceptibles and infected do not explicitly depend on time, f is reduced to only a function of y.

Code implementation Below, an implementation for the basic SIRS model using the fourth-order Runge Kutta method is given. The complete code can be found here Github repository.

Algorithm 1: Fourth-order Runge Kutta

Result: Population's health status

// Find value of y at a given time t and defined transition rates (a, b, c) using step size h. y contains $S = y[0]$, $I = y[1]$ and $R = y[2]$

initialization:

  Define $y_0 = [S_0, I_0, R_0]$, $N = S_0 + I_0 + R_0$, t_{end}, h and set up dy containing $dy[0] = S'$ and $dy[1] = I'$

Function `f(`yt`):`

  // Returns the derivatives, dy, for each time step

  $dy[0] = c * yt[2] - a * yt[1] * yt[0]/N$ // computing S'

  $dy[1] = a * yt[0] * yt[1]/N - b * yt[1]$ // computing I'

  return dy

for $i = h, 2h, \ldots, t_{end}$ **do**

  // Apply Runge Kutta fromulas to find next value of y

  $k_1 = hf(y_i)$

  $k_2 = hf(y_i + k_1/2)$

  $k_3 = hf(y_i + k_2/2)$

  $k_4 = hf(y_i + k_3)$

  $y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ // Update next value of y

end

2.2.2 Monte Carlo method

Ref. [2] and Ref.[5] served as supporting literature.
The Monte Carlo method belongs to the statistical simulation methods, since random numbers are used to perform the simulations. There are four terms characterizing the method. *Random variable* and *probability distribution functions (PDF)* are important for a random walk simulation. The expectation values comprising the *mean value* and *variance* are needed for studying the result.

Random variable As the outcome cannot be presumed, the random variable represents a stochastic variable. Taking two dice as an example. Their possible domain of numbers is $\{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$ with the corresponding probabilities $\{1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1\}\frac{1}{36}$. Even though we do not know the outcome, by using the probability, predictions about the range can be made. If the dice is thrown only once, there is no guarantee that the results will be 7, despite it being most likely. However, by repeating the experiment, the thrown numbers would eventually reflect the probabilities above.
For the SIRS model, the outcome will be based on the random variable, $r \in [0, 1]$, and the transition-probabilities for a person to move from one group to another. If the transition-probability is larger than r, the transition is accepted. Here, the random samples, that is the outcome, are drawn sequentially for each time-step. In addition, the distribution of the samples drawn depends on the last value, the health status of the population at that time. Thus, a *Markov chain* is created where each sample depends only on its direct predecessor. For instance, the more people are infected, the higher the probability for a susceptible person to become infected.

Probability distribution functions It should be mentioned that the set of differential equations in (5) and (6) naturally assume that S, I and R are continuous variables, but they are actually discrete. This problem can be solved by defining a set of transition probabilities.
Let $P_{i,j}$ describe the probability of a transition from state i to state j. Here, the possible states are S, I and R. After l time steps, t, the transition-probability reads

$$P_{i,j}^{(l)} = Prob(X_t = j | X_{t-1} = i), \tag{36}$$

where X stands for the stochastic variable and $Prob$ represents the transition matrix. Hence $P_{i,j}$ is the probability of a single-step transition solely depending on the previous state.
Apparent from Equations (5) and (6), the number of people moving from S to I can be estimated to be $\frac{aSI}{N}\Delta t$. Similarly, approximately $bI\Delta t$ move from I to R and $cR\Delta t$ from R to S. In order to constrain Δt such as at most one person moves from a given group to another, we first compute

$$\max\left\{\frac{aSI}{N}\Delta t\right\} = \frac{a}{N}\left(\frac{N}{2}\right)^2 \Delta t = \frac{aN}{4}\Delta t, \tag{37}$$

$$\max\left\{bI\Delta t\right\} = bN\Delta t, \tag{38}$$

$$\max\left\{cR\Delta t\right\} = cN\Delta t. \tag{39}$$

It follows that the time step, Δt, reads

$$\Delta t = \min\left\{\frac{4}{aN}, \frac{1}{bN}, \frac{1}{cN}\right\}. \tag{40}$$

Consequently, $\frac{aSI}{N}\Delta t$, $bI\Delta t$, and $cR\Delta t$ can be seen as transition probabilities:

$$p(S \to I) = \frac{aSI}{N}\Delta t, \tag{41}$$

$$p(I \to R) = bI\Delta t, \tag{42}$$

$$p(R \to S) = cR\Delta t. \tag{43}$$

This means, if each of the above probabilities is greater than a random number, $r \in [0, 1]$ a person moves from one group to another.
For the vital dynamics simulation, see Sec. 2.1.3, the population changes with time. Strictly speaking, Δt, given by Equation (40), must change with time accordingly. However, Δt is set constant to the value given by the initial condition.

Expectation values Let one experiment denote one Monte Carlo cycle, where for each time step, the number of people moving from one group to another are added to and subtracted from the previous values. We remind that for each time-step, three independent random numbers are drawn accepting or rejecting the move based on the three transition probabilities.
In order to calculate the functional average of S, I and R, these values are summed for each Monte Carlo cycle, MC. In other words, the average number of for example susceptibles at a given time, t, is found. Since we are interested in the data after the system has reached equilibrium, the average of the given averaged values after equilibration time is calculated. For the expected number of susceptibles at equilibrium, we get

$$\langle S\rangle = \frac{1}{n}\sum_{t=t_i}^{t_n}\frac{1}{MC}\sum_{k=1}^{MC} S_k(t), \tag{44}$$

where t_i marks the equilibration time and n represents the number of time steps between t_n and t_i. Furhter, $S_k(t)$ is the value after the k^{th} cycle at the time t. As the number of Monte Carlo cycles increases, the average approaches a more realistic value. The corresponding variance (standard deviation squared) is further defined as

$$\sigma_S^2 = \frac{\sum_{t=t_i}^{t_n}(S_t - \langle S\rangle)^2}{n}, \tag{45}$$

where again t_i marks the equilibration time and $S_t = \frac{1}{MC}\sum_{k=1}^{MC} S_k(t)$ represents the averaged number of susceptibles at a given time. $\langle S\rangle$ is given by Equation (44) and n is the number of time steps the values are summed over.
The average equilibration values $\langle I\rangle$, $\langle R\rangle$ and their variance can be calculated in a similar way.

Code implementation Below, an implementation for the basic SIRS model using the Monte Carlo simulation is given. The complete code can be found here Github repository.

Algorithm 2: Monte Carlo simulation

Result: Population's health status

// Find value of y at a given time t, and defined transition rates (a, b, c) using step size h. y contains $S = y[0]$, $I = y[1]$ and $R = y[2]$

initialization:

  Define $y_0 = [S_0, I_0, R_0]$, $N = S_0 + I_0 + R_0$, t_{end} and $h = \min\left\{\frac{4}{aN}, \frac{1}{bN}, \frac{1}{cN}\right\}$

for $j = 1, \ldots, MCcycles$ **do**

  // Run experiment multiple times

  for $i = h, 2h, \ldots, t_{end}$ **do**

    // Compute health status for each time step

    $pR_S = (cy[2])h$ // probability for a person to move from group R to S

    $pS_I = (ay[0] * y[1)/N)h$ // probability for a person to move from group S to I

    $pI_R = (by[1])h$ // probability for a person to move from group I to R

    Generate three uniform random numbers r_1, r_2, r_3

    reset Scount = 0, Icount = 0, Rcount = 0 // Allows to keep track of number of people moving within the three groups. Start sampling:

    if $r_1 < pR_S$ **then**

      Scount $+= 1$

    end

    if $r_2 < pS_I$ **then**

      Icount $+= 1$

    end

    if $r_3 < pI_R$ **then**

      Rcount $+= 1$

    end

    $y(0)+=$Rcount - Scount // Update number of susceptibles

    $y(1)+=$Scount - Icount

    $y(2)+=$Icount - Rcount

  end

end

// Average values obtained by summing values for each cycle and dividing by the total number of cycles

2.2.3 Relative error

Given the expected equilibrium values in Table 2, the relative error between the numerical results and the expected values can serve as a good means of comparison. The relative error, ϵ, reads

$$\epsilon = \left| \frac{x_e - x_n}{x_e} \right|, \tag{46}$$

where x_e stands for the expected and x_n for the numerical value. For the numerical values computed by the Monte Carlo method, x_n will be given by Equation (44). Note that for group D, the relative error for the number of infected and recovered is not defined.

3 Results

The code used to obtain the following results can be found here Github repository. The different rates that have unit inverse time are not specified further.

3.1 Simple SIRS model

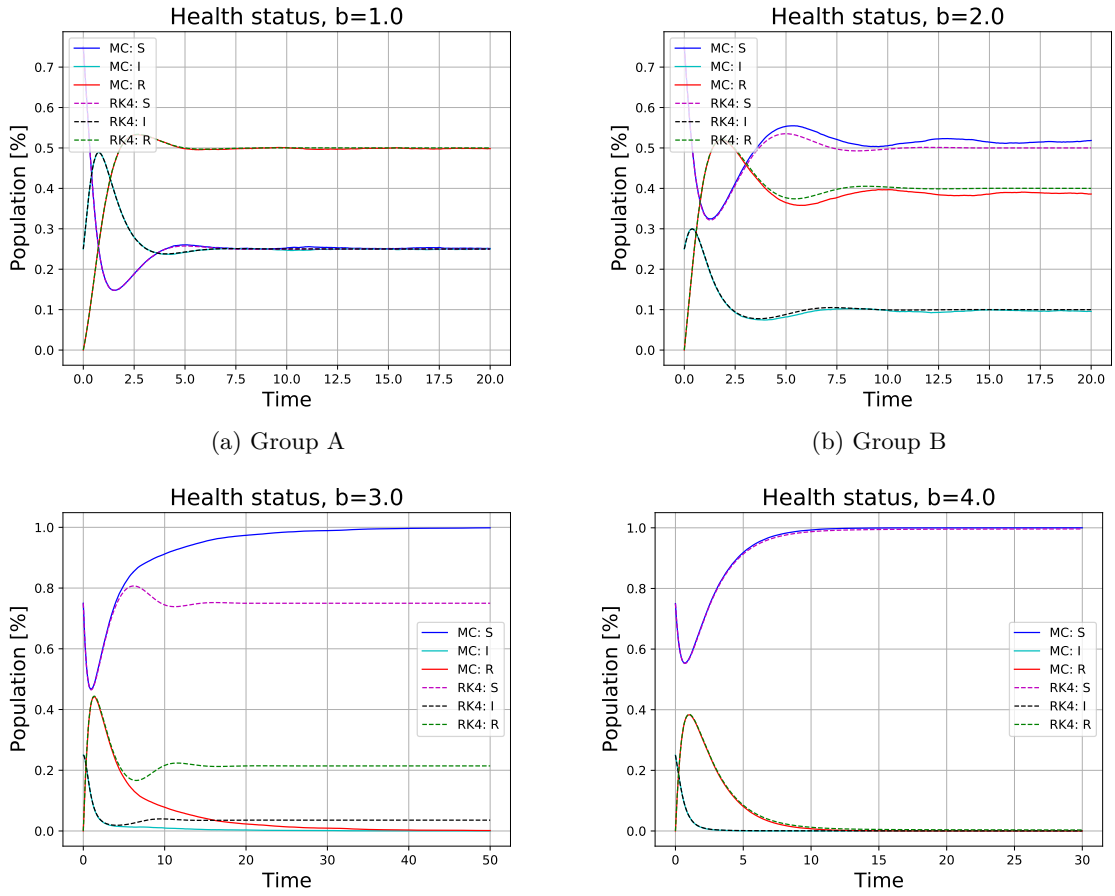
First, the simple SIRS model, see Equations (5) and (6), is simulated by using both the RK4, with a step-size of $\Delta t = 0.0025$, and the Monte Carlo method, with 1000 cycles. Further initial conditions are given in Table 1. Figure. 1a to Figure. 1d show the results for Group A to Group D. Table 3 lists the standard deviation (Equation (45)) resulting from the Monte Carlo method after equilibration time. The equilibration time for the different groups from A to D is $10, 10, 40$ and 15, respectively. In addition, the relative error between the numerical and expected equilibrium values, for both methods, are given in Table 4. As the expected number of infected and recovered is zero for Group D, the relative error is not defined in this case.

Table 3: Standard deviation after equilibration time, as defined in Equation (45), resulting from Monte Carlo method using 1000 cycles for different population-groups with initial conditions listed in Table 1.

	Standard deviation			
	Group A	Group B	Group C	Group D
S	$1.24 \cdot 10^{-3}$	$4.03 \cdot 10^{-3}$	$5.12 \cdot 10^{-4}$	$1.21 \cdot 10^{-4}$
I	$9.54 \cdot 10^{-4}$	$1.73 \cdot 10^{-3}$	$8.13 \cdot 10^{-5}$	0.00
R	$9.50 \cdot 10^{-4}$	$3.56 \cdot 10^{-3}$	$4.45 \cdot 10^{-4}$	$1.21 \cdot 10^{-4}$

Table 4: Relative error, ϵ, between the numerical and expected equilibrium values for different population groups defined in Table 1. Numerical values gained from both Runge Kutta 4 (RK4), with $\Delta t = 0.0025$, and Monte Carlo (MC) method, with 1000 cycles.

	Group A		Group B		Group C		Group D	
	ϵ_{MC}	ϵ_{RK4}	ϵ_{MC}	ϵ_{RK4}	ϵ_{MC}	ϵ_{RK4}	ϵ_{MC}	ϵ_{RK4}
S	$1.01 \cdot 10^{-2}$	0.00	$3.15 \cdot 10^{-2}$	$8.5 \cdot 10^{-5}$	$3.30 \cdot 10^{-1}$	0.00	$6.70 \cdot 10^{-5}$	$3.73 \cdot 10^{-3}$
I	$3.58 \cdot 10^{-3}$	0.00	$3.58 \cdot 10^{-2}$	$1.25 \cdot 10^{-5}$	$9.91 \cdot 10^{-1}$	$1.00 \cdot 10^{-6}$	–	–
R	$3.27 \cdot 10^{-3}$	0.00	$3.04 \cdot 10^{-2}$	$1.00 \cdot 10^{-4}$	$9.89 \cdot 10^{-1}$	$1.67 \cdot 10^{-7}$	–	–



(a) Group A

(b) Group B

(c) Group C

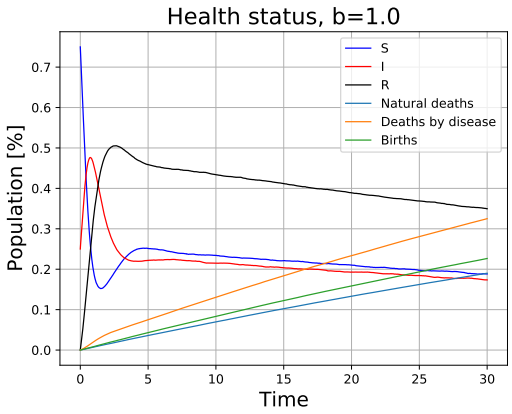
(d) Group D

Figure 1: Simulation of the four different population-groups defined in Table 1. The plots show the results from both the Runge-Kutta 4, $\Delta t = 0.0025$, and Monte Carlo method, 1000 cycles.

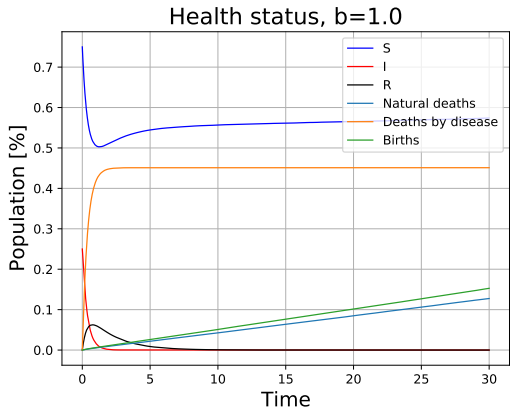
3.2 Vital dynamics

As described in Sec. 2.1.3, vital dynamics allow to more realistically simulate the spread of diseases that occur over longer stretches of time. Figure. 2a to Figure. 2d show the results for Group A and Group D using the Monte Carlo method with 1000 cycles. The natural birth and death rate are set to 0.009 and 0.0075, respectively. The death rate of the infected people due to the disease is varied from 0.05 to 5. Other initial conditions are listed in Table 1.

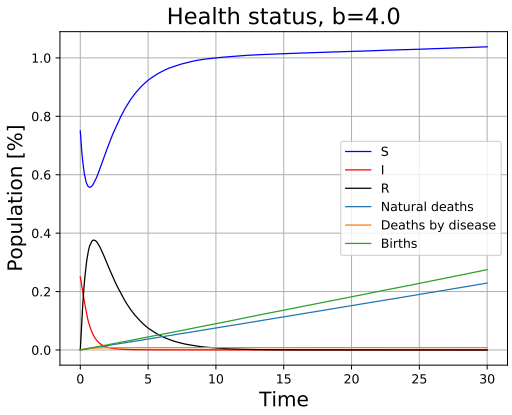
Figure 3 shows the portion of the total population that dies due to the disease, as a function of the disease death parameter d_I. This was simulated with the MC method, using 600 Monte Carlo cycles.



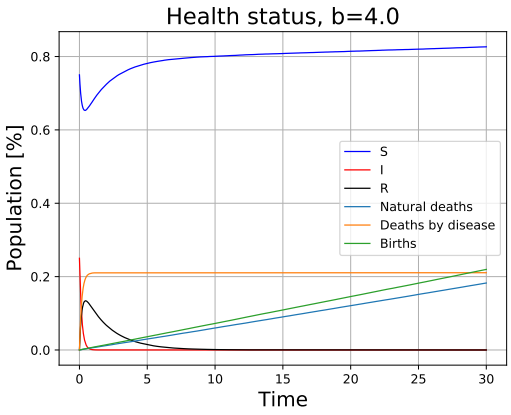
(a) Group A, $d_I = 0.05$



(b) Group A, $d_I = 5$



(c) Group D, $d_I = 0.05$



(d) Group D, $d_I = 5$

Figure 2: Group A and D (defined in Table 1) with two different disease death parameters d_I and natural birth and death rate are set to 0.009 and 0.0075, respectively; simulated with 1000 Monte Carlo cycles

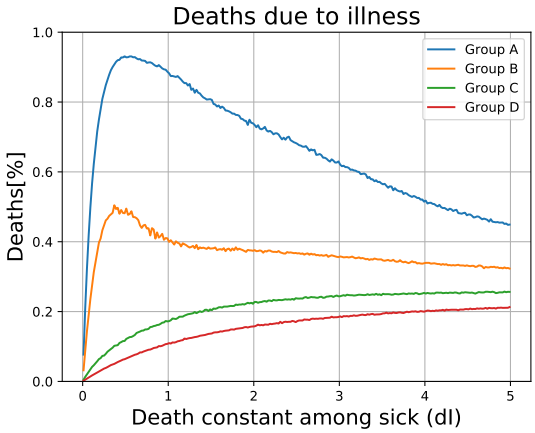
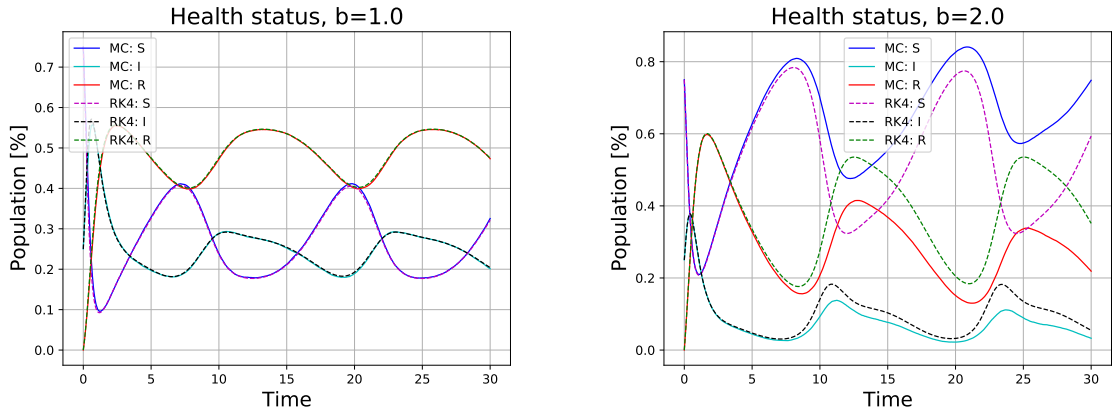


Figure 3: Simulated the effect of the of deadliness of the disease for the four different groups (defined in Table 1), with $\Delta d_I = 0.02$ and natural birth and death rate are set to 0.009 and 0.0075, respectively. All simulations ran with 600 Monte-Carlo cycles.

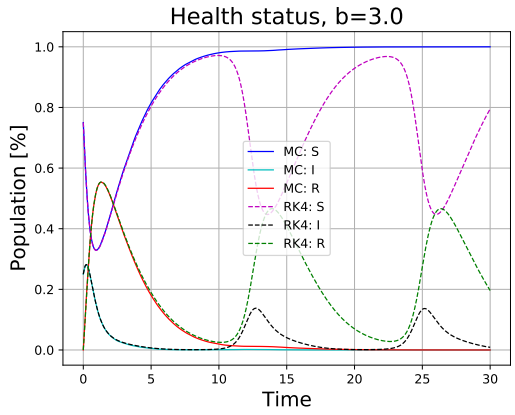
3.3 Seasonal variation

Figure 4 shows the implementation of seasonal variation for the rate of transmittion a, see Equation (13). All the groups are presented, with both the Runge-Kutta 4 and Monte Carlo method in each subplot. The parameters for a are $A = 1.5$, $\omega = 0.08 \cdot 2\pi$, and $a_0 = 4$.

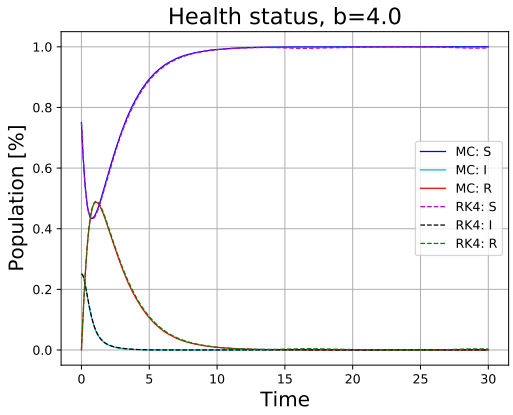


(a) Group A

(b) Group B



(c) Group C



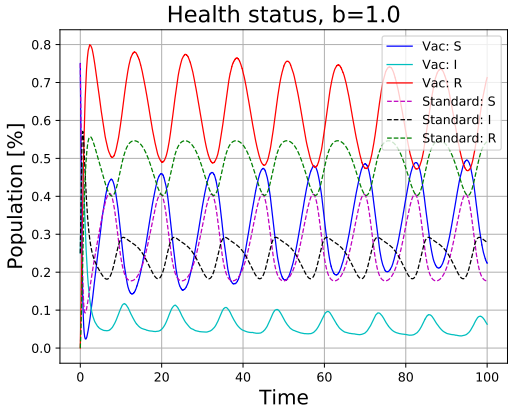
(d) Group D

Figure 4: Simulations of the effects of a seasonally varying disease for different groups defined in Table 1. The rate of transmission however varies with time, as $a = 1.5\cos(0.08 \cdot 2\pi t) + 4$. Both the Runge-Kutta 4, $t = 0.0025$, and Monte Carlo method, 1000 cycles, are presented.

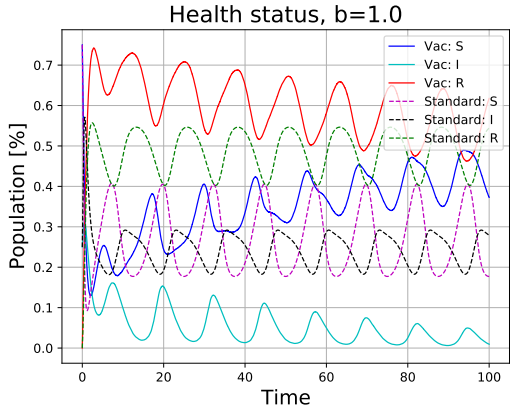
3.4 Vaccination

Figure 5 compares the simple SIRS model with and without vaccination for group A. The parameter a varies as presented in equation (13), with $A = 1.5$, $\omega = 0.08 \cdot 2\pi$ and $a_0 = 4$. When the vaccination was constant, we used $f = 100$, and when it varied (see Equation (17)), we used $F = 70$, $\omega = 0.08 \cdot 2\pi$ and $f_0 = 100$. The simulation was done with 1000 Monte Carlo cycles, but the RK4 method yields the same results.

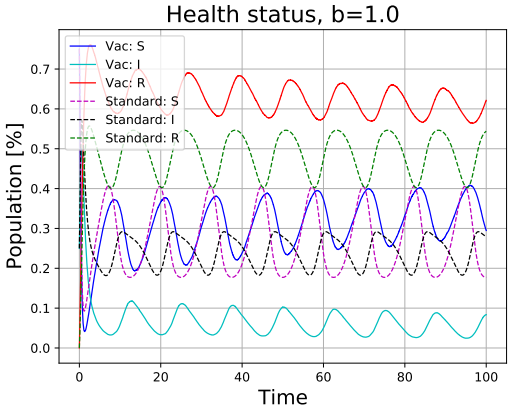
Figure 3 shows the portion of the population that dies as a result of the disease, as a function of the vaccination parameter f. This was only done with the Monte Carlo method, using 500 Monte Carlo cycles. The disease death parameter d_I was set to 1.



(a) f and a in phase



(b) f and a with a 180° phase difference



(c) Constant f

Figure 5: Impact of vaccination using the Monte Carlo method with 1000 cycles. Rate of transmission and vaccination as defined in Equation (13) and Equation (17) with $A = 1.5$, $\omega = 0.08 \cdot 2\pi$, $a_0 = 4$, $F = 70$ and $f_0 = 100$. Other parameters given in Table 1 under group A.

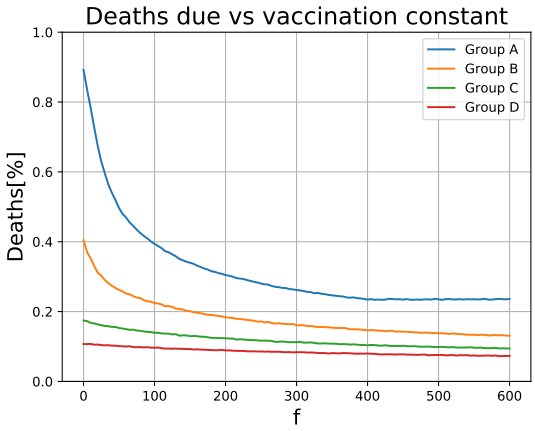


Figure 6: Deaths due to a disease as a function of the vaccination parameter f, $\Delta f = 5$, using 500 Monte Carlo cycles. The natural birth, death and death due to disease rate are set to 0.009, 0.0075 and 1.0, respectively. The simulation was done or group A (defined in Table 1).

4 Discussion

4.1 Simple SIRS model

Apparent from Table 2, which lists the standard deviation of the equilibrium values for the different population groups, resulting from the MC method, the standard deviation is of the order 10^{-3} to 10^{-5}. Thus, the equilibrium expectation values can be considered to be constant in time. The standard deviation is greatest for group B. Increasing the number MC cycles to more than 1000 does not change the standard deviation notably. Simulating for a longer period of time, however, makes a bigger impact, though the effect is small and thus does not justify longer calculations. Noteworthy, for group D and the number of infected, the equilibrium values do not vary at all.

The relative error between the numerical MC values and expected ones, see Table 4, has almost the same magnitude within each population group. Group 5 performs best with a difference of 0.0067%, even though only the error for the number of susceptibles is defined. Group 3 has the largest deviation, where the relative error is close up to 100% for I and R. The values for group B did not seem to stabilize. Therefore, at $t \approx 20$, the deviation is about 30%. Generally, group A and group B have a good accordance with the expected values. As stated for the standard deviation, the choice of the equilibration time influences the results.

In general, RK4 matches better with the expected equilibrium values. Again, the magnitude of the error within each population group is almost constant, varying from 10^{-3} to 10^{-7}. For Group A, there is no difference at all.

Looking at Figure 1a and 1d, there is no significant difference between the two methods. As $b = 1$ for group A, the disease remains in the population with 25% infected in the long run. Shortly after the start of the simulation, the number of infected has a peak of almost 50%. Whereas for group D, where the recovery rate is $b = 4$, the number of infected decreases quickly after the start.

If group B is simulated with the MC method longer than shown in Figure 1b, the number of susceptibles continuously increases while the number of infected decreases accordingly. The reasons for this are

still unclear and must be investigated further. For group C, the MC method states that the disease should vanish as for group D. This is contrary to the expected equilibrium values given by Equations (7) to (9). Hence, the large large relative error mentioned above. RK4 states that the disease should remain, but the number of infected reduces to about 2.5%. Therefore, we see a clear trend. As the rate of recover increases, there are fewer infected.

4.2 Vital dynamics

In Figure 2, it is easy to see that an increased disease death rate d_I results in a higher number of deaths short term. The disease also disappears rather quickly because everyone that is infected dies. In the long term, this is actually preferred. If we compare Figures 2a and 2b, it is easy to see that the number of deaths due to the disease peaks at about 45% of the population when $d_I = 5$, but continues to grow when $d_I = 0.5$. The number of individuals in all the population groups slowly decreases. It seems like this will continue until the entire population is dead, which is why a higher death rate is preferred for this example. When comparing Figures 2c and 2d on the other hand, a lower death rate leads to less deaths. This is because the recovery rate b is high in this group, which means the disease disappears regardless of the death rate. A higher death rate now leads to more deaths in total. In other words, a low death rate is good when the recovery rate is high, but when the recovery rate is low, a high death rate is better in the long run.

4.3 Seasonal variation

When implementing seasonal variation for the rate of transmission, we see similarities in all the groups. It seems they reach a stabilized state at about the same time as the simple SIRS model. In group A, we see that both the RK4 and MC method yields the same results. They reach a stabilized state, and all the population groups oscillates with the rate of transmission. For group B on the other hand, we see a difference between the RK4 and MC method. They both stabilizes in the same amount of time, but the infection and recovered rate is a little lower for the MC method. In group C, the difference between the two methods continues, but this time the difference is rather huge. Before stabilization, they give the same results, but after stabilization, the MC method reaches an equilibrium, while the RK4 method keeps oscillating with the rate of transmission. For group D, both the methods yield the same results again, and they both stabilizes.

We see that the disease disappears from the population when $b = 4$, as expected from equation (8), and as discussed earlier, this is also the case for the MC simulation when $b = 3$. Even though group D stabilizes in about the same time as without the seasonal variation, as we see when comparing 4d and 1d, there is a slight difference. The difference is that the number of infected peaks higher in the beginning when we have the seasonal variation. This is due to the rate of transmission being at a maximum in the beginning, as it is modeled by a cosine wave.

4.4 Vaccination

Figure 5 may look chaotic at first, but there are some interesting observations to be made. We first take a look at 5a. With vaccination, the number of infected is greatly decreased, and the number of recovered (and immune) is greatly increased. Remember that the rate of transmission a and the

vaccination parameter f here has the same period, and they both start at a maximum. If we look at Figure 5b on the other hand, a is now shifted, such that it starts at a minimum. That means a lot of susceptibles are vaccinated in the beginning. When the rate of transmission increases, there is a great number of immune, which means the disease does not spread as much. In other words, vaccinating before the rate of transmission is at a maximum is a great way to prevent the spread of a disease. The last subplot 5c shows a constant vaccination parameter. The result of this is about the same as when the vaccination parameter oscillates with the same phase as the rate of transmission.
In Figure 6, we see that less people die when we increase vaccination. We also see that the number of deaths decrease when the recovery rate b increases. The death rates are about 90% , 40% and 18% for groups A, B and C respectively, when there is no vaccination. When the vaccination parameter $f = 600$, the death rates are now about 24%, 13% and 10% respectively. This shows the difference in number of deaths between the groups is also less significant when vaccination is large.

4.5 Code optimization

There are many things that can be done in order to optimize the code we made for this project. One of the things we considered multiple times during the project, was that these simulations did not require a lot of CPU power. Thus, we made a lot of choices for easier implementation in the code, that does not necessarily give the fastest run time. However, if we were to simulate a much bigger population, for a longer period of time, there are many changes that can be made in order to make the code run faster. For instance, we made three different derivative functions for the RK4 method, and we used an if-statements that checks which derivative function to use. These if-statements were checked every time-step, which is not ideal. Alternatively, we could have used the address in memory for the correct derivative function as an argument when calling the RK4 function.
One thing our code does not allow, is simulation the use of vital dynamics and vaccines at the same time. This would not be hard to implement, but we felt it wasn't necessary, as we already have a lot of different interesting results. When initializing the class objects, we have to specify which implementations to use, and so on. The way we did this turned out to be messy and unwieldy. There are currently 8 functions just for initializing the class objects. This could probably have been done a much better way, although we do not have a specific suggestion.

5 Conclusion

The rates stated have the unit inverse time.
First, we looked at four population groups that have have different recovery rates b. For $b = 1$ and $b = 4$, the Monte Carlo method (1000 cycles) and the fourth-order Runge Kutta method ($\Delta t = 0.0025$) give the same results. The standard deviation after equilibration time resulting from the Monte Carlo method was of the magnitude of 10^{-3} to 10^{-5}. The relative error between the expected equilibrium values and the Monte Carlo results for group A is of the order of 10^{-2} and 10^{-3}. The RK4 results are exactly as the expected ones. Generally, as b increases, the fewer are infected. For $b = 4$ using RK4, the disease does not remain in the population. Using the Monte Carlo method, this already happens for $b = 3$. The reason for this discordance is not yet known and must be investigated further.
By extending the model with vital dynamics, it was shown that an increased disease death rate results

in a higher number of deaths short term. Moreover, the disease also vanishes more quickly, as larger number of infected die and thus cannot infect anyone. Further, a low death rate is positive when the recovery rate is high, but when the recovery rate is low, a high death rate saves more lives in the long run.
Computations of a seasonally varying disease showed that all groups react similarly and stabilized themselves at about the same time as the simple SIRS model.
Finally, adding the possibility of vaccination, we showed that fewer people die when they get vaccinated. Moreover, vaccinating before the rate of transmission is at a maximum is a great way to prevent the spread of a disease.
The models used are simplified. It was assumed that the population mixes humorously and factors such as contaminated surfaces that may pose a risk of infection were neglected. Additionally, the population was isolated, which no longer applies in a global world. In general, real data must be used to examine the reality of the two methods. The rates have unit inverse time, but were not specified further. The unit is dependent on the effects of the disease, as well as on the data collected by the healthcare system. A reasonable unit might be days. In addition, it could be investigated whether there are different diseases that are more suitable for the models. Since the Monte Carlo method is based on random sampling, the realistic factor might be higher than for an ordinary differential solver.
In conclusion, an accurate prediction of the spread of an infectious disease is an important tool to introduce appropriate safety measures.

6 Appendix

6.1 Solving the SIRS model

6.1.1 Euler method

The following explanation is taken from Ref. [1].
The Euler method is a simple first-order numerical method for solving ordinary differential equations with a given initial value. Defining the first derivative of y_i as

$$y'(t_i) = f(t_i, y_i). \tag{47}$$

Assuming that the function is rather well-behaved in the observed interval $[t_0, t_{end}]$, h is held constant. In other words, we use a fixed-step method.
In order to find an expression for y_{i+1}, we Taylor expand our function y

$$\begin{aligned} y_{i+1} &= y(t = t_i + h) \\ &= y(t_i) + h(y'(t_i) + \cdots + y^{(p)}(t_i)\frac{h^{p-1}}{p!}) + O(h^{p+1}), \end{aligned} \tag{48}$$

with $O(h^{p+1})$ representing the truncation error. Abbreviating the derivatives in the parenthesis with

$$(y'(t_i) + \cdots + y^{(p)(t_i)}) = \Delta(t_i, y_i(t_i)), \tag{49}$$

we can rewrite Equation (48) to

$$\begin{aligned} y_{i+1} &= y(t = t_i + h) \\ &= y(t_i) + h\Delta(t_i, y_i(t_i)) + O(h^{p+1}). \end{aligned} \tag{50}$$

Truncating the Taylor expansion at the first derivative, we get

$$y_{i+1} = y(t_i) + hf(t_i, y_i) + O(h^2) \tag{51}$$

$$= y_{i+h}. \tag{52}$$

where $O(h^2)$ represent the truncation error. In other words, for every step we make a round off error of the order of $O(h^2)$. Thus, the total error is the sum over all the steps L

$$LO(h^2) = \frac{t_{end}}{h} O(h^2) \approx O(h). \tag{53}$$

Euler's method is not recommended for precision calculation due to round of errors.

Round of errors It is seems plausible that by decreasing the step size h, Euler's method should increase in accuracy. However, the risk of round-off errors also increases. Using the two-step formula to numerically compute the derivative f

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h), \tag{54}$$

round-off errors can occur if $f(x + h) - f(x) \approx 0$. This is the reason why the Euler method is not recommended for precision calculation.

References

[1] Bendik Nyheim Elias Tidemand Ruud and Mira Mors. *Project 3: Modelling the Solar System Using Ordinary Differential Equations; Computing Newton's second law according to ordinary Euler and Velocity Verlet*. Nov. 2020. URL: https://github.com/EliasTRuud/FYS3150/blob/master/Project3/Report/FYS3150ReportProject3.pdf. (accessed: 03.12.2020).

[2] Yong Chung Han. *Mathematics of Viral Infections: A Review of Modeling Approaches and A Case-Study for Dengue Dynamics*. Aug. 2018. URL: https://munin.uit.no/bitstream/handle/10037/14104/thesis.pdf?sequence=2&isAllowed=y. (accessed: 05.12.2020).

[3] Morten Hjorth-Jensen. *Computational Physics Lectures: Ordinary differential equations*. Oct. 2017. URL: http://compphysics.github.io/ComputationalPhysics/doc/pub/ode/pdf/ode-print.pdf. (accessed: 03.12.2020).

[4] Morten Hjorth-Jensen. *Project 5: Disease Modeling*. Sept. 2020. URL: http://compphysics.github.io/ComputationalPhysics/doc/Projects/2020/Project5/DiseaseModeling/pdf/DiseaseModeling.pdf. (accessed: 27.11.2020).

[5] Torbjørn Paschen Seland. *Mathematical Analysis of Epidemic Systems; Comparison of Different Models*. Dec. 2014. URL: https://munin.uit.no/bitstream/handle/10037/14104/thesis.pdf?sequence=2&isAllowed=y. (accessed: 05.12.2020).