# Comparison between non-deterministic model and deterministic model for Malaria and Sickle-Cell Hemoglobin gene

Eleven (Shiyi) Chen sc7825@nyu.edu

April 9, 2021

## Contents



## 1 Introduction

Malaria is a contagious disease that is mainly spread on the Mosquitoes[1], while the Sickle-Cell Homeoglobin gene is a recessive gene located at the chromosome which may cause the red blood cell to look like a sickle[2]. The sickle-shaped blood cell cannot transport the oxygen properly thus causing the condition. As we know, Human chromosomes come in pairs, so the Homeoglobin gene. Let us denote the mutated gene(Sickle-Cell Homeoglobin gene) as 1 while the normal gene is 0. Thus, there are three kinds of genotype:

$$11, 01(\text{ same as } 10), 00$$

The 11 genotype causes expression of the recessive gene, thus the blood cell will become sickle-like and the person will die because of lack of oxygen. The 01 and 00 genotypes cause the expression of the normal blood cell gene, thus no Sickle-Cell disease. **However,** 01 **genotype can prevent people from getting the Maria while** 00 **does not[1].**

In this project, we will introduce a model with its numerical implementation to quantify the evolution of two diseases. Moreover, two variances - the deterministic version and the random version will support each other but also show the difference when the initial number of 01 genotype people is very small.

| Notation | Meaning | Default value in the program |
|---|---|---|
| $S$ | Number of susceptible | 5e4 |
| $I$ | Number of infected | 2e4 |
| $R$ | Number of resistant | 3e4 |
| $N$ | Total number of people | 1e5 |
| $B_S$ | Susceptible new born | N/A |
| $B_R$ | Resistant new born | N/A |
| $\delta$ | Death rate for normal people | 0.01 |
| $\delta'$ | Death rate for malaria patient | 0.02 |
| $\beta$ | Birth rate for normal people | 0.02 |
| $\beta'$ | Birth rate for malaria patient | 0.01 |
| $a$ | Infectious rate for malaria | 0.5 |
| $\lambda$ | Growth rate of the whole population | N/A |

Table 1: Notation used in this report

# 2 Math model for the disease

This section focus on the mathematical description of the model, the notation used in this section can be found in Table 1. In the following sections, the default values are used if not specified in the report. The actual evolution of two diseases depends on many factors, for example, the intervention of medical cures, the number of the mosquito which is affected by the climate. To simplify the problem, we made those assumptions:

1. There is no recovery in the model, What $R$ means here is the resistant population. **Don't confuse the $R$ with that in the SIR model.**

2. For the Sickle-Cell gene, there are three kinds of genotype: $01, 11, 00$, the 11 genotype causes the death of a person immediately when he/she is born. We call 01 genotype people resistant, 00 type of people susceptible. Notice that, our nomenclatures are malaria-oriented.

3. No mutation is considered in our model.

4. We only consider the female population in our model, and the gene frequency of the Sickle-Cell gene is the same among males and females.

5. No interventions for both diseases which means that there are no cures or medicines.

6. The malaria is spread by mosquitoes in the real world. To simplified the model, the effect of the mosquitoes is quantified by the infection rate which describes how many new cases a patient can infect during one period.

7. $\beta, \delta$ denote the birth rate and death rate respectively for the people who do not have malaria while $\beta', \delta'$ denotes the birth rate and death rate for the people who have malaria. Those parameters follow the relation
$$\beta > \beta', \delta < \delta', \beta > \delta, \beta' < \delta'.$$
The fist two inequalities can be interpreted as the people with malaria has a lower birth rate and a higher death rate compared to normal people. As for the last two equations, the birth rate is greater than the death rate for the case when there is no Malaria while the death rate is smaller than the birth rate when everyone get Malaria

Based on those assumptions, we can build a model as shown below:

$$\begin{aligned} \frac{dS}{dt} &= B_{\mathrm{S}} - \delta S - a(I/N)S \\ \frac{dI}{dt} &= \quad -\delta' I + a(I/N)S \\ \frac{dR}{dt} &= B_{\mathrm{R}} - \delta R \\ N &= S + I + R. \end{aligned} \tag{1}$$

The $B_S, B_R$ are the new born of the susceptible and resistant respectively. To calculate $B_S, B_R$, we need to know the how many people are going to born and what type of baby they will born. Thus, we treat 01 and 00 type female differently. For 00 type female(either susceptible or infected), the probability to born 01 and 00 type of baby are denoted by $P_{00,01}, P_{00,00}$ respectively. Similarly, we have $P_{01,00}, P_{01,01}$ to denote 01 type female give birth to $00, 01$ type of baby respectively.

Since we have the assumption that the gene frequencies for males and females are the same. For a 00 type female, it has a $(S+I)/N$ chance to marry with a 00 type male, and a $R/N$ chance to meet a 01 type male. 00 type female with 01 type male has $1/2$ chances to get a 00 type baby and $1/2$ chance to get a 01 type baby due to meiosis. For another scenario, if this 00 female meets a 00 type male, they can only bore 00 type baby. Thus,

$$P_{00,00} = \frac{S+I}{N}(1) + \frac{R}{N}\left(\frac{1}{2}\right).$$

Similarly,

$$\begin{aligned} \mathrm{P}_{00,01} &= \frac{R}{2N} \\ \mathrm{P}_{01,00} &= \frac{S+I}{2N} + \frac{R}{4N} \\ \mathrm{P}_{01,01} &= \frac{S+I}{2N} + \frac{R}{2N} \\ \mathrm{P}_{01,11} &= \frac{R}{4N} \end{aligned}$$

Finally, we can calculate the new born $B_S, B_R$ with those probabilities.

$$B_{\mathrm{S}} = (\beta S + \beta' I)\, P_{00,00} + \beta R P_{01,00} \tag{2}$$

and

$$B_{\mathrm{R}} = (\beta S + \beta' I)\, P_{00,01} + \beta R P_{01,01} \tag{3}$$

with $(\beta S + \beta' I)$ denotes the number of 00 female that going to born new baby while $\beta R$ denotes the number of 01 female that are going to born new baby.

# 3 Numerical method and implementation in the code

Both the deterministic and non-deterministic versions of simulation apply Euler's method. They both calculate the changes during $dt$, then add the changes to the original $S, I, R$. The $dt$ is set to be one year for simplicity. Finally, recalculate $N$ by adding $S, I, R$ together. For the deterministic version of the simulation, the right-hand sides of Equation 1 are directly calculated by Equation 2 and Equation 3. Thus, the deterministic model allows the existence of the fraction for the population which is not the case in real scenarios. However, when the population is reasonably large, it makes sense to ignore this peculiarity.

For the non-deterministic version of the simulation. The steps come as follow:

First, calculate probabilities $P_{00,01}, P_{00,00}, P_{01,00}, P_{01,01}$ the same way as the deterministic version of the simulation. Then calculate the number of female who are going to give birth to babies by generating random

array with element from 0 to 1 then comparing to the birth rate $\beta, \beta'$ for $00, 01$ group separately. Using $P_{00,01}, P_{00,00}, P_{01,00}, P_{01,01}$, the number of new born $B_S, B_R$ can also be calculated. For the detailed implementation, the code is listed in the Appendix. The non deterministic version of simulation only allow the population to be positive integers thus it may be more accurate to simulate the situation when the population is quite small.

The argument validation in Matlab is a new feature since R2019b which can be quite handy because one can run the program without specifying any inputs. Also, parameters can come with any order, any combination, for example:

```
determ('a',0.6,'T',2000);
determ('T',1000,'d',0.7,'a',2);
```

In Matlab R2021a, the name value pairs also support:

```
determ(a = 0.6, T = 2000);
```

Another feature of the code is the way to generate random numbers. Matlab rand function generates value with the class double by default. However, we do not need this precision and it could be faster by just generate the single class by:

```
rand(1, n, 'single').
```

# 4 Result and discussion

## 4.1 Compare with different infectious rate-$a$ using deterministic and non deterministic simulation

With the deterministic and non-deterministic versions, simulations were made by changing parameter $a$. Figure 1 shows the long-term behavior of the Susceptible, Infected, Resistant population over time. For the long term, $SIR$ all settle down into a steady growth rate as suggested by the linearity in exponential axis this is also supported by the right side of the Figure 2 which shows the growth rate as time goes by. When $a = 0.01$, the resistant goes extinct. Also, the resistant has been diluted as time goes by(the tangent of the blue curve is larger than the yellow). This is due to the reason that the resistant has a $1/4$ chance to born a dead baby and thus the natural selection is against the resistant and tends to the susceptible. This effect is also suggested by the left side of Figure 2 which plots the ratio of $S, I, R$ when the population settles down into the steady growth rate state.

As $a = 0.1$, the portion of susceptible people (blue curve) goes down. We can also see the prevalence of the Sickle Cell gene because there is more malaria. As $a = 0.5$, a little fluctuation in susceptible for the non-deterministic version is observed because the number of susceptible is small enough for the randomness showing. Compared with $a = 0.1$, the growth rate is lower due to the prevalence of Malaria and the Sickle-Cell gene. With $a = 3$, the fluctuation of $S$ is more obvious.

## 4.2 A close look at the growth rate $\lambda$ with increasing $a$

The growth rate $\lambda$ at time $t$ is calculated using the total population at time $t$ and population at time $t - dt$, i.e.

$$\lambda(t) = \frac{N(t) - N(t - dt)}{N(t)}.$$

By carefully looking at the right side of the Figure 2, we noticed that the larger the $a$ is, the smaller the $\lambda$ will stabilize in. Also, the larger the $a$, the closer the two lines will stabilize. To show this feature clearly, I plot the relationship between $a$ and $\lambda$ as shown in Figure 3. The yellow curve represents the result from the

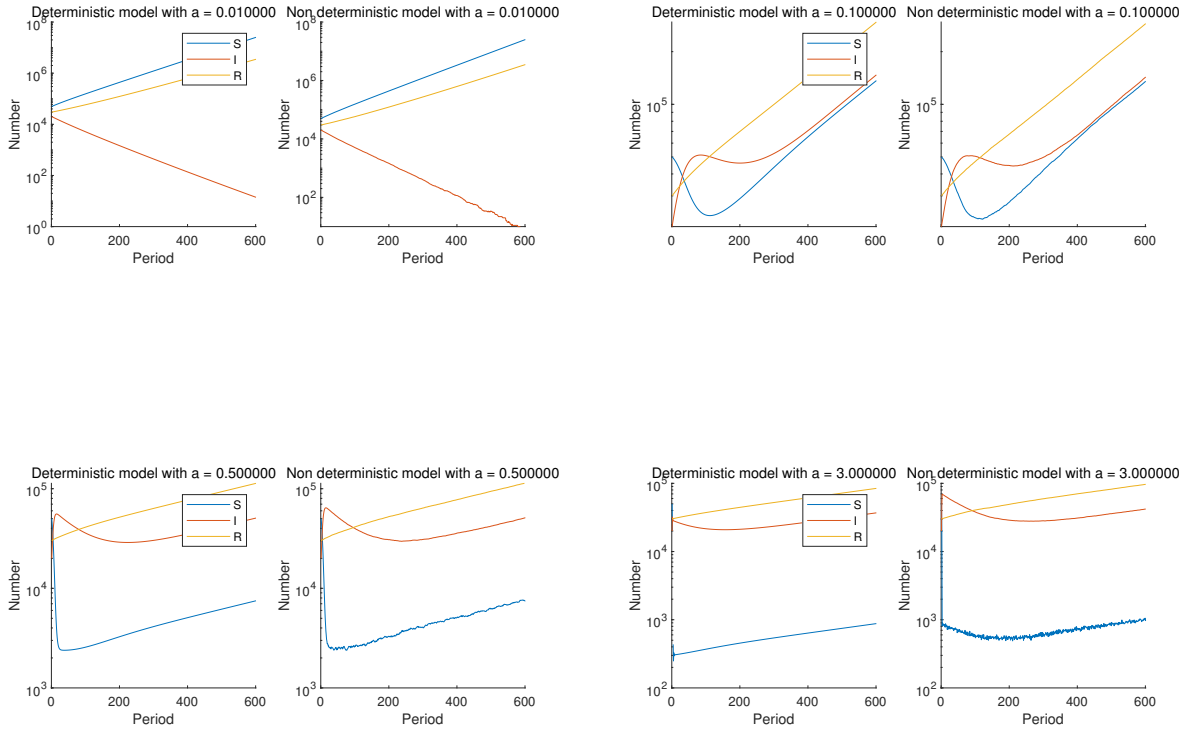


Figure 1: $S, I, R$ v.s. Time comparing with the two models, year is used for the unit of the time

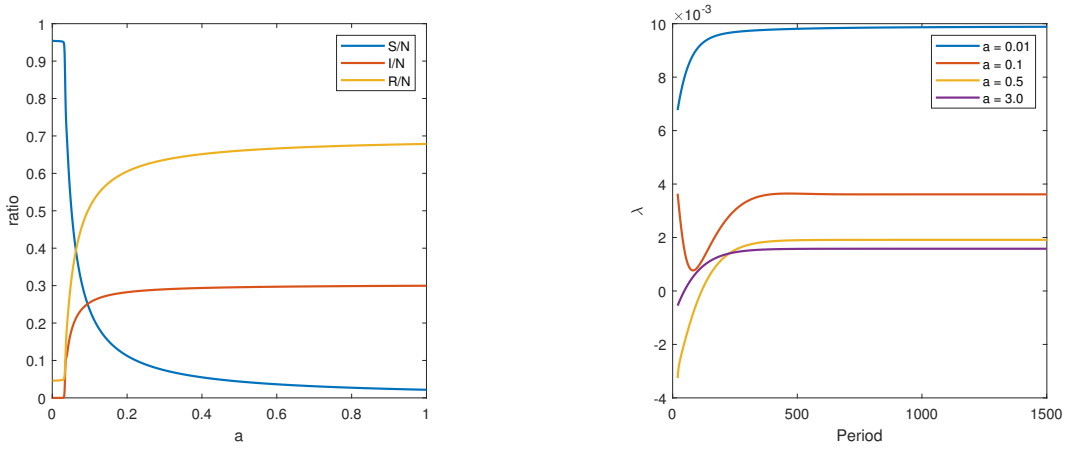


Figure 2: $a$ v.s. the ratio of $S, I, R$ population when the simulation arrives at the stable growth rate state(left), the time v.s. the growth rate of the populate with $a = 0.01, 0.1, 0.5, 3.0$ (right)

deterministic model while the blue dots represent the result from the non-deterministic model. On the one hand, we can see that the deterministic yellow curve agrees very well with the blue dots. Moreover, we see that as $a$ gets larger, the growth rate $\lambda$ decreases and finally set down to a fixed value between 2 and 1.5.

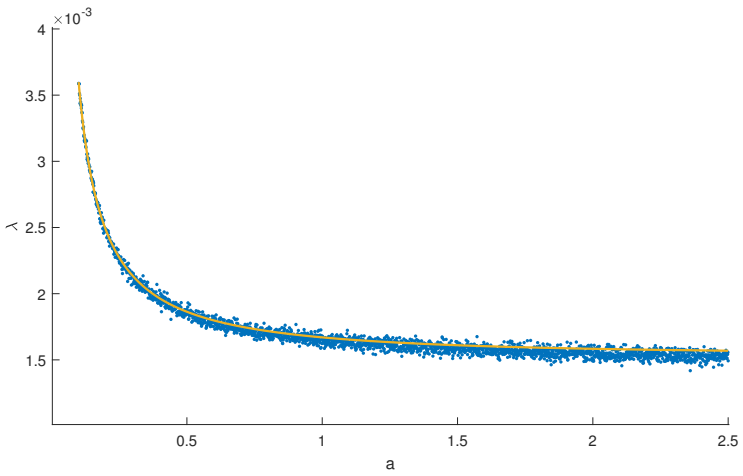


Figure 3: a v.s. the stabilized growth rate $\lambda$, the blue dots represent the result from the non-deterministic model while the yellow curve represents the result from the deterministic model

## 4.3 When initial $R(t = 0)$ is very small

Everything discussed above based on the initial population of $S, I, R$ is the same as the default value as shown in Table 1. It will be very interesting to see what will happen if there are only several resistant cases while keeping the ratio of the susceptible and resistant to be the same.

### 4.3.1 The deterministic model

Figure 4 shows the deterministic version of the simulation. The left side shows the situation for when there is no resistant while the right side shows what happens if there is only one resistant person. It is quite surprising to see that even one person with the Sickle-Cell gene could save the whole population from extinction which is the case on the left side. This "one person saves whole human beings" miracle reveals the process of natural selection and the evolution of the human species. Even with one mutation(01) in the human group, the mutated gene(1) can spread across the species because the (01) gene can prevent Malaria although it will cause 1/4 death.

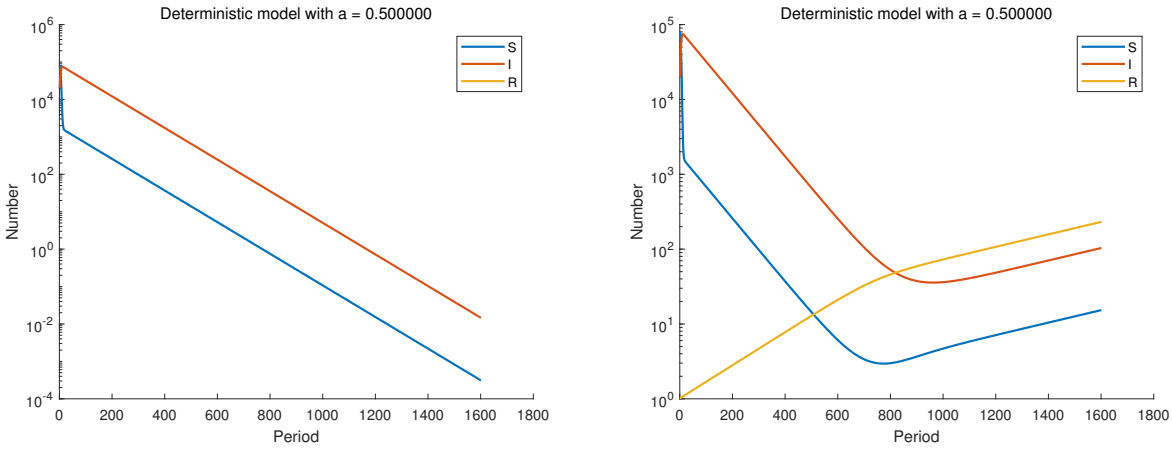


Figure 4: The deterministic simulation when the initial R = 0 (left) and the initial R = 1(right)

### 4.3.2 The non-deterministic model

While we see the "one person saves whole human beings" paradigm in the deterministic model, it is not always the case with the non-deterministic model as we can see from Figure 5. Thus, when $R$ is very small, there are some situations when the Malaria causes human extinction while there are also some other situations when the resistant people are too few to prevent the whole human beings go extinct.

To quantified the finding for the non-deterministic model, the initial value $R(0)$ ranging from 1 to 10 is tested with 100 times each. During each test, we record the final result either resistant dies out-human being extinct or resistant save the whole human beings. The result is shown in Figure 6. For $R(0)$ equal to 1 or 2, we see that the $R$ gene dies out easily. However, as $R(0)$ increases, the chances for the gene to die out is quite low.

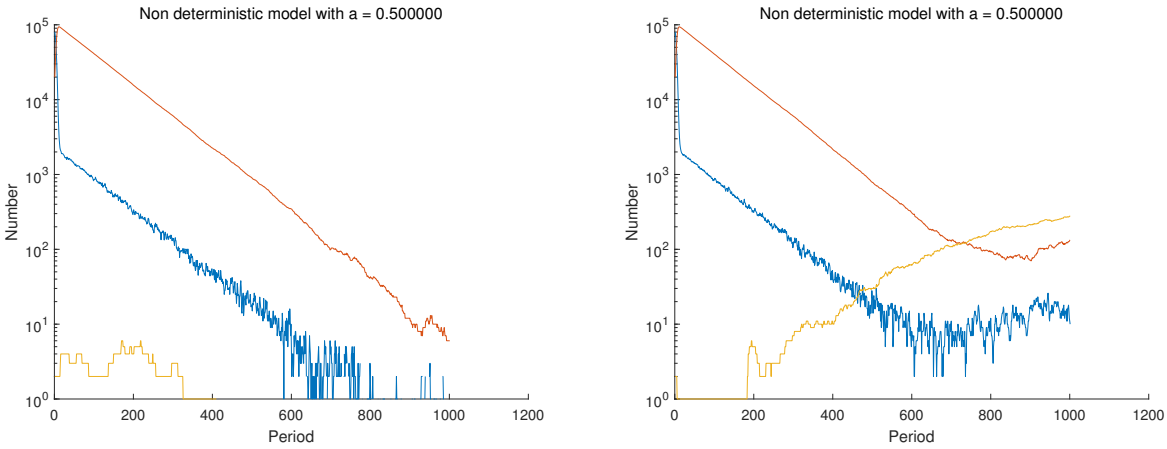


Figure 5: Two rounds of simualtion with initial $R = 2$

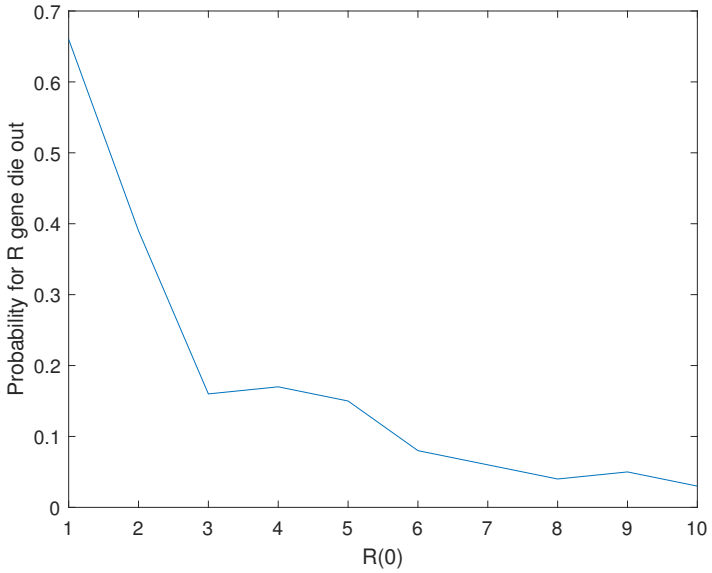


Figure 6: Experiment the different initial resistant people

# 5 Summary and conclusion

In this project, we explore the long-term behavior of the Sickle-Cell gene and Malaria. We first introduce the mathematical model and its implementation in Matlab in Section 2 and Section 3 respectively. The model has two version-the deterministic versions and the non-deterministic version. In section 4.1, we find that with the default value for initial susceptible, infected and resistant, the deterministic model and the non-deterministic model have similar behavior. Moreover, as the infection rate $a$ gets larger, the ratio of Infected and resistant also gets larger as the result of the natural selection.

However, the number of initial resistant is very small. It is no longer the case that the two models have similar behavior. As for the deterministic model, the final fate of the human being - either distinct or survive on the Malaria crisis, only depends on whether the resistant exists or not. If the initial resistant is a positive number, the human being then will survive, if it is zero, then human beings will go extinct. As for the non-deterministic model, we argue that the final fate for human beings depends on the initial number of the resistant. With more resistant at time 0, it is less likely that human beings will go extinct.

# References


[1] CDC. *What is malaria?* 2021. URL: https://www.cdc.gov/malaria/about/faqs.html#:~:text=Malaria%20is%20a%20serious%20and,humans%3A%20Plasmodium%20falciparum%2C%20P. (visited on 04/08/2021).

[2] Lung National Heart and Blood Institute. *Sickle Cell Disease*. 2021. URL: https://www.nhlbi.nih.gov/health-topics/sickle-cell-disease (visited on 04/08/2021).

# Appendices

## A determ.m

```
% compatibility: Matlab R 2019b+

function [lambda,S,I,R,N] = determ(ops)

% argument default and validations
arguments
    ops.N = 1e5;
    ops.I = 2e4;
    ops.R = 3e4;
    ops.T = 600;
    ops.d =0.01
    ops.d_ = 0.02
    ops.b  = 0.02
    ops.b_ = 0.01
    ops.a = 0.5;
    ops.Plot = 1;
    ops.saveVariable = 1
end

b = ops.b; d = ops.d; b_ = ops.b_; d_ = ops.d_;
N = ops.N; I = ops.I; R = ops.R; S = N-I-R;
a = ops.a; T = ops.T;

if ~(b>b_)
    warning("b must be greater than b_")
elseif ~(d<d_)
    warning("d must be less than d_")
elseif ~(b>d)
    warning("b must be greater than d")
elseif ~(b_<d_)
    warning("b_ must be less than d_")
end

if ops.saveVariable==1
    Ssave = zeros(1,T+1); Isave = zeros(1,T+1); Rsave = zeros(1,T+1);Nsave = zeros(1,T+1);
    lambdas = zeros(1,T+1);
    Ssave(1) = S; Isave(1) = I; Rsave(1) = R; Nsave(1) = N;
end

% the main loop for the simualtion
for t = 1: T
    pgtb00 = (b*S+b_*I); % the 00 type people going to born (pgtb00)
    pgtb01 = (b*R);          % the 01 type people going to born (pgtb01)
    P00_00 = (S+I)/N+R/(2*N); % probability for 00 type woman give birth to 00 type child
    P00_01 = 1-P00_00 ;% probability for 00 type woman give birth to 01 type child
    P01_00 = (S+I)/(2*N)+R/(4*N); % probability for 01 type woman give birth to 00 type
    child
    P01_01 = (S+I)/(2*N)+R/(2*N); % probability for 01 type woman give birth to 01 type
    child
    BS = pgtb00*P00_00+pgtb01*P01_00;% new suspetible baby(00)
    BR = pgtb00*P00_01+pgtb01*P01_01;% new resistant baby(01)

    % update S I R
    % we are suposing dt=1 so I didn't create a variable called dt
    S = S+BS-d*S-a*I*S/N;
    I = I-d_*I+a*I*S/N;
    R = R+BR-d*R;

    % if any quatity is smaller than 0, make it equal to 0
    if S<0; S=0; end
    if I<0; I=0; end
    if R<0; R = 0; end
```

```
61
62      N = S+I+R;
63
64      % save the variable
65      if ops.saveVariable == 1
66          Ssave(t+1) = S; Isave(t+1) = I; Rsave(t+1) = R;
67
68      end
69      Nsave(t+1) = N;
70      lambdas(t+1) = (N-Nsave(t))/N;
71  end
72
73  lambda = mean(lambdas(floor(ops.T/2):end));
74
75  % plot SIR if speficied
76  if ops.Plot == 1
77      hold on
78      plot(1:T+1,Ssave,'LineWidth',1.5);plot(1:T+1,Isave,'LineWidth',1.5);plot(1:T+1,Rsave,'
        LineWidth',1.5);
79      set(gca,'YScale','log')
80      legend('S','I','R')
81      xlabel('Period')
82      ylabel('Number')
83      title(sprintf('Deterministic model with a = %2f',a));
84      hold off
85  end
86
87  % save the variables if required
88  if ops.saveVariable==1
89      save('SIR','Ssave','Isave','Rsave','Nsave','lambdas')
90  end
91  end
```

# B nondeterm.m

```
% compatibility: Matlab R 2019b+

function [lambda,S,I,R,N] = nondeterm(ops)

% argument default and validations

arguments
    ops.N = 1e5;
    ops.I = 2e4;
    ops.R = 3e4;
    ops.T = 600;
    ops.d=0.01
    ops.d_ = 0.02
    ops.b  = 0.02
    ops.b_ = 0.01
    ops.a = 0.5;
    ops.Plot = 1;
    ops.saveVariable = 1
end

b = ops.b; d = ops.d; b_ = ops.b_; d_ = ops.d_;
N = ops.N; I = ops.I; R = ops.R; S = N-I-R;
a = ops.a; T = ops.T;

if ~(b>b_)
    warning("b must be greater than b_")
elseif ~(d<d_)
    warning("d must be less than d_")
elseif ~(b>d)
    warning("b must be greater than d")
elseif ~(b_<d_)
    warning("b_ must be less than d_")
end

if ops.saveVariable==1
    Ssave = zeros(1,T+1); Isave = zeros(1,T+1); Rsave = zeros(1,T+1);Nsave = zeros(1,T+1);
    lambdas = zeros(1,T+1);
    Ssave(1) = S; Isave(1) = I; Rsave(1) = R; Nsave(1) = N;
end

% the main loop for the simualtion
for t = 1: T
     pgtb00 = (b*S+b_*I); % the 00 type woman that are going to born
    pgtb01 = (b*R);          % the 01 type woman that are going to born
    P00_00 = (S+I)/N+R/(2*N); % probability for 00 type woman give birth to 00 type child
    P00_01 = 1-P00_00 ;% probability for 00 type woman give birth to 01 type child
    P01_00 = (S+I)/(2*N)+R/(4*N); % probability for 01 type woman give birth to 00 type
    child
    P01_01 = (S+I)/(2*N)+R/(2*N); % probability for 01 type woman give birth to 01 type
    child
    BS = pgtb00*P00_00+pgtb01*P01_00;% new suspetible baby(00)
    BR = pgtb00*P00_01+pgtb01*P01_01;% new resistant baby(01)

    % dealing with infected people
    % number of Infected new borns
    Ib = sum(rand([1 I],'single')<b_);%number of Infected people get borned

    % Resistant baby born from Infected mom
    I2R =  sum(rand([1 Ib], 'single')<P00_01);

    % Suspectible baby born from Infected mom
    I2S = sum(rand([1 Ib], 'single')<P00_00);

    % dealing with suspectible people
    Sb = sum(rand([1 S], 'single')<b);
```

```
64
65      % Resistant baby born from suspectible mom
66      S2R =  sum(rand([1 Sb], 'single')<P00_01);
67
68      % Suspectible baby born from suspectible mom
69      S2S =  sum(rand([1 Sb], 'single')<P00_00);
70
71      % dealing with Resistant people
72      Rb = sum(rand([1 R],'single')<b);
73
74      % Resistant baby born from Resistant mom
75      R2R =  sum(rand([1 Rb], 'single')<P01_01);
76
77      % Suspectible baby born from resistant mom
78      R2S =  sum(rand([1 Rb], 'single')<P01_00);
79
80      % consoder the spread of the desease
81      S2I = sum(rand([1 S], 'single')<a*I/N);
82
83      % consider some people dies
84      Id = sum(rand([1 I], 'single')<d_);
85      Sd = sum(rand([1 S], 'single')<d);
86      Rd = sum(rand([1 R], 'single')<d);
87
88      S = S + I2S + R2S + S2S - Sd-S2I;
89      I = I+S2I-Id;
90      R = R +I2R + R2R +S2R - Rd;
91
92      % if any quatity is smaller than 0, make it equal to 0
93      if S<0; lambda = 0;return; end
94      if I<0; lambda = 0;return; end
95      if R<0; lambda = 0;return; end
96
97      N = S+I+R;
98
99      % save the variable
100     if ops.saveVariable == 1
101         Ssave(t+1) = S; Isave(t+1) = I; Rsave(t+1) = R;
102
103     end
104     Nsave(t+1) = N;
105     lambdas(t+1) = (N-Nsave(t))/N;
106
107 end
108 lambda = mean(lambdas(floor(ops.T/2):end));
109
110 % plot SIR if speficied
111 if ops.Plot == 1
112     hold on
113     plot(1:T+1,Ssave);plot(1:T+1,Isave);plot(1:T+1,Rsave);
114     set(gca,'YScale','log')
115     xlabel('Period')
116     ylabel('Number')
117     title(sprintf('Non deterministic model with a = %2f',a));
118     hold off
119 end
120
121 % save the variables if required
122 if ops.saveVariable==1
123     save('SIR','Ssave','Isave','Rsave','Nsave','lambdas')
124 end
```