

Project 4: Modeling the Spread of an Infectious Disease

Jane Kim

Physics 480: Computational Physics

(Dated: May 7, 2018)

A Monte Carlo simulation of the spread of an infectious disease in an isolated population was constructed from the SIRS model. Four populations (A,B,C, and D) with different rates of recovery were studied and compared to the predictions from a deterministic approximation. One hundred samples were generated for each population, which was enough to produce relatively smooth averages but varying standard deviations. Larger uncertainties were observed in populations B and C due to the enhanced nonlinearity of the system.

I. INTRODUCTION

The goal of this project was to develop a Monte Carlo simulation of the spread of an infectious disease. The classical SIRS model of epidemiology was studied and used to develop the necessary transition probabilities for the simulation. Interpreting the simulation requires a thorough understanding of the SIRS model, so a deterministic approach was used in conjunction with the Monte Carlo methods.

The main purpose of creating such a simulation is to investigate how a disease spreads throughout a given population over time. Then we can make predictions about whether or not a certain disease has the capacity to establish itself within the population, i.e. a fraction of the population remains infected after the system reaches equilibrium.

In the following section, the SIRS model is discussed in-depth. Then in Section III, the SIRS model is adapted for the Monte Carlo simulation and four test cases are described. The implementation of the SIRS model using both deterministic and stochastic approaches is presented in Section IV, the results of which are shown in Section V. Finally, possible improvements upon the model are described in Section VI and the conclusion is given in Section VII.

II. THEORY

The SIRS model considers an isolated population of N people which are divided into three separate groups:

- Susceptible (S): those without immunity to the disease,
- Infected (I): those who are currently infected with the disease,
- Recovered (R): those who have been infected in the past and have developed an immunity to the disease.

A person can move from one group to another only in the cyclic order suggested by the name of the model $S \rightarrow I \rightarrow R \rightarrow S$. The rate of transmission a , the rate of recovery b and the rate of immunity loss c help describe the flow of people moving between the three groups. The population is assumed to mix homogeneously and total population is assumed to remain constant so that

$$N = S(t) + I(t) + R(t). \quad (1)$$

For the simple case, we will assume that the dynamics of the epidemic occur during a time scale much smaller than the average person's lifetime. Hence the effect of the birth and death rate of the population is ignored.

From these assumptions, a set of coupled differential equations can be constructed to form the classical SIRS model:

$$\begin{aligned} S' &= cR - \frac{aSI}{N} \\ I' &= \frac{aSI}{N} - bI \\ R' &= bI - cR \end{aligned} \quad (2)$$

Note that for small populations, one may choose to write aSI instead of $\frac{aSI}{N}$ since the number of susceptibles which become infected depend more on the absolute number of infected people rather than the infected fraction of the population.

Though this set does not have analytic solutions like the closely-related SIR model, the equilibrium solutions are simple to obtain. The constraint in (1) reduces this three dimensional system into a two dimensional one so that the equation for R' can just be omitted:

$$\begin{aligned} S' &= c(N - S - I) - \frac{aSI}{N} \\ I' &= \frac{aSI}{N} - bI \end{aligned} \quad (3)$$

The steady state is found by setting both equations in (3) equal to zero. Let s , i , and r denote the fractions of people in S , I , and R , respectively. Then the fractions

of people in each group at equilibrium are:

$$\begin{aligned} s^* &= \frac{b}{a}, \\ i^* &= \frac{1 - \frac{b}{a}}{1 + \frac{b}{c}}, \\ r^* &= \frac{b}{c} \frac{1 - \frac{b}{a}}{1 + \frac{b}{c}}. \end{aligned} \quad (4)$$

Here, the asterisk (*) signifies that these fractions are at equilibrium. Notice that each fraction must be a number between 0 and 1, and that the three fractions must add up to 1. Thus the equations in (4) suggest that the rate of recovery b must be less than the rate of transmission a for the number of infected people at equilibrium to be greater than zero. In other words, the disease establishes itself in the population only if $b < a$.

III. METHOD

Four different populations were created to investigate the effect of increasing the rate of recovery b and crossing the "threshold". Each population consisted of 400 people, 100 of which were initially infected and 300 of which were initially susceptible. The rate of transmission a and the rate of immunity loss c were consistent between populations, but the rate of recovery b was varied because it is the one parameter which can be reasonably controlled by the actions of human society (access to health care, development of new medicines, etc). Note that the rates have units of inverse time.

Rate	A	B	C	D
a	4	4	4	4
b	1	2	3	4
c	0.5	0.5	0.5	0.5

Table 1: Rates of transmission a , recovery b , and immunity loss c for populations A, B, C, and D.

The set of differential equations in (2) inherently assume that S , I , and R are continuous variables, when they are discrete in reality. We can use the idea of randomness to remedy this issue by defining a set of transition probabilities for the possible moves a person can take from one state to another: $S \rightarrow I$, $I \rightarrow R$, and $R \rightarrow S$. To obtain the specific values for each probability, we first notice from (2) that in a small time step Δt , the number of people moving from S to I is approximately $\frac{aSI}{N}\Delta t$. Likewise, about $bI\Delta t$ move from I to R and $cR\Delta t$ move from R to S .

Now we let Δt be small enough so that *at most* one person moves from a given group to another. Since

$$\begin{aligned} \max \left\{ \frac{aSI}{N} \Delta t \right\} &= \frac{a}{N} \left(\frac{N}{2} \right)^2 \Delta t = \frac{aN}{4} \Delta t, \\ \max \left\{ bI \Delta t \right\} &= bN \Delta t, \\ \max \left\{ cR \Delta t \right\} &= cN \Delta t, \end{aligned} \quad (5)$$

the time step is then given by

$$\Delta t = \min \left\{ \frac{4}{aN}, \frac{1}{bN}, \frac{1}{cN} \right\}. \quad (6)$$

This construction allows us to reinterpret the values $\frac{aSI}{N}\Delta t$, $bI\Delta t$, and $cR\Delta t$ as transition probabilities:

$$\begin{aligned} P(S \rightarrow I) &= \frac{aSI}{N} \Delta t, \\ P(I \rightarrow R) &= bI \Delta t, \\ P(R \rightarrow S) &= cR \Delta t. \end{aligned} \quad (7)$$

For each possible move, a random number between 0 and 1 is generated. If the number is less than the probability for the move, the move is taken.

IV. IMPLEMENTATION

A class called `CInfectedPopulation` was created to carry the parameters N , a , b , and c for each population. Then its member functions `deterministic_SIRS` and `montecarlo_SIRS` solve the system with a deterministic and Monte Carlo approach, respectively. Both functions require a file name and initial conditions, and the latter also requires a number of samples `nsamples`.

A. Deterministic Approximation

The function `deterministic_SIRS` solves the system of differential equations in (2) using the second order Runge-Kutta formula. The number of people in each category are then written to file at each time step.

```
void CInfectedPopulation::deterministic_SIRS(string
    filename, double S0, double I0, double tf){

    // open file
    ofstream outfile;
    outfile.open(filename + ".dat");
    cout << "write to ---> " << "' ' <<
        filename+".dat'" << endl;

    // file headings
    outfile << "# N = " << N_ << endl;
    outfile << "# (S0, I0, R0) = (" << S << ", " <<
        I << ", " << R << ")" << endl;
```

```

outfile << "# (a, b, c) = (" << a_ << ", " << b_
    << ", " << c_ << ")" << endl;
outfile << "# time, S, I, R" << endl;

// initial conditions
double S = S0, I = I0, R = N_-S0-I0;
double S_k1, S_k2, I_k1, I_k2, R_k1, R_k2;

for(double t = 0.0; t < tf; t += dt_){

    // print
    outfile << t << "\t" << S << "\t" << I <<
        "\t" << R << endl;

    // use RK2 method to calculate next S, I, R
    S_k1 = dt_*(c_*R-a_*S*I/N_);
    I_k1 = dt_*(a_*S*I/N_-b_*I);
    R_k1 = dt_*(b_*I-c_*R);

    S_k2 = dt_*(c_*(R+0.5*R_k1)
        -a_*(S+0.5*S_k1)*(I+0.5*I_k1)/N_);
    I_k2 = dt_*(a_*(S+0.5*S_k1)*(I+0.5*I_k1)/N_
        -b_*(I+0.5*I_k1));
    R_k2 = dt_*(b_*(I+0.5*I_k1) -c_*(R+0.5*R_k1));

    S += S_k2;
    I += I_k2;
    R += R_k2;
}
// close file
outfile.close();
}

```

A phase portrait was generated for each population by running the above function with various initial conditions and plotting S vs. I as the system evolves over time.

B. Monte Carlo Simulation

The vast majority of `montecarlo_SIRS` is devoted to calculating the averages and standard deviations of the `nsamples` samples for each time step. Only three simple lines within a loop is required to actually run the simulation. Hence it is quite quick and easy to obtain the results.

```

void CInfectedPopulation::montecarlo_SIRS(string
    filename, int nsamples, int S0, int I0, double
    tf){

    // RNG
    mt19937 generator;
    uniform_real_distribution<double> rand01(0.0,
        1.0);

    // time vector
    vector<double> time;
    for(double t = 0.0; t < tf; t += dt_)
        time.push_back(t);
    int ntimes = time.size();

```

```

// averages for each point in time
vector<double> avgS(ntimes,0.0),
    avgI(ntimes,0.0), avgR(ntimes,0.0);

// create random samples
for(int n = 0; n < nsamples; ++n){

    // open file
    ofstream outfile;
    outfile.open(filename+to_string(n)+".dat");
    cout << "write to ---> " << " " <<
        filename+to_string(n)+".dat" << endl;

    // file headings
    outfile << "# N = " << N_ << endl;
    outfile << "# (S0, I0, R0) = (" << S << ", "
        << I << ", " << R << ")" << endl;
    outfile << "# (a, b, c) = (" << a_ << ", " <<
        b_ << ", " << c_ << ")" << endl;
    outfile << "# time, S, I, R" << endl;

    // initial conditions
    int S = S0, I = I0, R = N_-S0-I0;

    // run simulation
    for(int i = 0; i < ntimes; ++i){

        // write to file
        outfile << time[i] << "\t" << S << "\t" <<
            I << "\t" << R << endl;

        // calculate averages
        avgS[i] += S/(double) nsamples;
        avgI[i] += I/(double) nsamples;
        avgR[i] += R/(double) nsamples;

        // keep-or-reject
        if(rand01(generator) < a_*S*I*dt_/N_){ I
            += 1; S -= 1; }
        if(rand01(generator) < b_*I*dt_){ R += 1;
            I -= 1; }
        if(rand01(generator) < c_*R*dt_){ S += 1;
            R -= 1; }
    }
    outfile.close();
}

double t, S, I, R;
double avg_sigS = 0.0, avg_sigI = 0.0, avg_sigR
    = 0.0;
string dummy;
vector<double> varS(ntimes,0.0);
vector<double> varI(ntimes,0.0);
vector<double> varR(ntimes,0.0);
int count = 0;

// calculate variances
for(int n = 0; n < nsamples; ++n){

    ifstream infile;
    infile.open(filename + to_string(n) + ".dat");

```

```

// ignore headings
for(int i = 0; i < 4; ++i) getline(infile,
    dummy);

// calculate variance and average standard
deviation
for(int i = 0; i < ntimes; ++i){

    infile >> t >> S >> I >> R;

    varS[i] +=
        (avgS[i]-S)*(avgS[i]-S)/(nsamples-1);
    varI[i] +=
        (avgI[i]-I)*(avgI[i]-I)/(nsamples-1);
    varR[i] +=
        (avgR[i]-R)*(avgR[i]-R)/(nsamples-1);

    avg_sigS +=
        sqrt(varS[i])/(nsamples*ntimes);
    avg_sigI +=
        sqrt(varI[i])/(nsamples*ntimes);
    avg_sigR +=
        sqrt(varR[i])/(nsamples*ntimes);
}

infile.close();
}

// write averages and standard deviations to file
ofstream outfile;
outfile.open(filename+to_string(nsamples)
    +"_stats.dat");
cout << "write to ---> " << "' ' <<
    filename+to_string(nsamples) + "_stats.dat"
    << endl;

outfile << "# N = " << N_ << endl;
outfile << "# nsamples = " << nsamples << endl;
outfile << "# (S0, I0, R0) = (" << S0 << ", " <<
    I0 << ", " << N_-S0-I0 << ")" << endl;
outfile << "# (a, b, c) = (" << a_ << ", " << b_
    << ", " << c_ << ")" << endl;
outfile << "# time avg std dev (<sigma S>,
    <sigma I>, <sigma R>) = (" << avg_sigS << ",
    " << avg_sigI << ", " << avg_sigR << ")" <<
    endl;
outfile << "# time, avg S, sigma S, avg I, sigma
    I, avg R, sigma R" << endl;

for(int i = 0; i < ntimes; ++i){
    outfile << time[i] << "\t" << avgS[i] << "\t"
        << sqrt(varS[i]);
    outfile << "\t" << avgI[i] << "\t" <<
        sqrt(varI[i]);
    outfile << "\t" << avgR[i] << "\t" <<
        sqrt(varR[i]) << endl;
}

outfile.close();
}

```

V. RESULTS

Based on (4) and our choices of parameters in Table 1, the fractions of people at equilibrium in each category is as follows:

Fraction	A	B	C	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

Table 2: The expected fractions of people in each category at equilibrium for populations A , B , C , and D .

These fractions provide a useful test for both approaches to solving the SIRS model.

A. Deterministic Approximation

The number of people in each category is plotted in Figure 1 for all four populations. As expected, the number of infected people drops as the rate of recovery increases. However, the population becomes vulnerable to an outbreak when the rate of recovery equals the rate of transmission (population D) because the entire population eventually becomes susceptible.

The phase portraits for each population are shown in Figure 2. Since the total population is assumed to be constant, the area above the dashed lines are inaccessible. In general, the solutions spiral toward a stable equilibrium point located somewhere within the boundaries of the triangle or at $(s,i)=(1,0)$. Notice that for populations B and C , the spirals are confined to a smaller region in phase space, leading to large changes in one variable and small changes in the other.

B. Monte Carlo Simulation

One hundred samples were generated for each population and plotted in Figure 3. This was enough to produce smooth averages, but the average for population C did not agree with the deterministic approximation. For example, the fraction of people in the susceptible category appeared to tend to 1 rather than 3/4.

The standard deviations are relatively small for populations A and D , but large for B and C . This may be due to the exaggerated nonlinearity found in the phase portraits for these populations. The time-averaged standard deviations for each population are:

	A	B	C	D
$\langle\sigma_S\rangle$	7.095	16.56	20.98	3.492
$\langle\sigma_I\rangle$	6.716	7.561	4.075	0.651
$\langle\sigma_R\rangle$	5.892	12.73	17.89	3.221

Compared to the population size of $N = 400$, the differences between populations are significant.

VI. IMPROVEMENTS

The same principles used in this simple model can be extended to include more details about the population and disease.

A. Vital Dynamics

For instance, vital dynamics can be easily added to the system so that the model can describe the spread of diseases which occur over longer stretches of time. If b is the birth rate, d is the death rate, and d_I is the death rate of infected people due to the disease, then the modified differential equations are given by:

$$\begin{aligned} S' &= cR - \frac{aSI}{N} - dS + eN \\ I' &= \frac{aSI}{N} - bI - dI - d_I I \\ R' &= bI - cR - dR \end{aligned} \quad (8)$$

Here, we have assumed that all the babies born into the population are initially susceptible.

B. Seasonal Variation

For diseases such as influenza, the rate of transmission depends largely on the time of year. During the colder months, individuals are more likely to spend time in closer proximity to one another, resulting in a rate of transmission which oscillates. So we can let a be given by

$$a(t) = A \cos(\omega t) + a_0, \quad (9)$$

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

C. Vaccination

Diseases with available vaccinations allow people to move directly from S to R , breaking the cyclic structure of the SIRS model. We assume that a susceptible individual's choice to become vaccinated does not depend

on how many other susceptibles are vaccinated. We may, however, assume that the rate of vaccination f can depend on the time, since this rate may oscillate during the course of a year and/or increase as awareness and medical research increases. Then the system of differential equations become

$$\begin{aligned} S' &= cR - \frac{aSI}{N} - f \\ I' &= \frac{aSI}{N} - bI \\ R' &= bI - cR + f \end{aligned} \quad (10)$$

D. Inhomogeneity

A major flaw of the SIRS model is that the population is assumed to mix homogeneously. This is not at all realistic, since we all know from experience that it is more likely for people to infect others that come into contact with them, either directly or indirectly. So we can instead develop a spatial model for the spread of a disease using the same compartmental design of the SIRS model.

We can start by creating a square lattice with N lattice sites. We randomly place I_0 infected people and R_0 recovered people on the lattice, and filling the remaining lattice sites with susceptibles. Then at each time step, a random infected site is chosen from a list and recovered with probability β . Next, one of its nearest neighbors (with periodic boundary conditions) is randomly chosen and infected with probability α . Finally, a recovered site is randomly chosen from another list and becomes susceptible with probability γ .

These transition probabilities α , β , and γ are different from (7) since equations (2) no longer apply. A simple alternative would just depend on the rates, such as:

$$\begin{aligned} \alpha &= \frac{a}{a + b + c} = P(S \rightarrow I) \\ \beta &= \frac{b}{a + b + c} = P(I \rightarrow R) \\ \gamma &= \frac{c}{a + b + c} = P(R \rightarrow S) \end{aligned} \quad (11)$$

VII. CONCLUSION

Overall, the results of the Monte Carlo simulation closely followed the deterministic approximations when the critical point of the phase portrait (Figure 2) was either far from the boundary or on the boundary itself. When it was near the boundary, the heightened nonlinearity caused the spread of the samples to significantly rise.

The SIRS model is an excellent starting point for modeling the spread of an infectious disease. As shown in the

previous section, additional details can be added later on to match the specifics of a given disease and/or population. A major drawback of the model, however, is that the population is assumed to mix homogeneously. Therefore, the development of a spatial model is recommended.

¹ M. Hjorth-Jensen. "Computation Physics, Lecture Notes Fall 2015". University of Oslo. August 2015.

² <http://leonidzhukov.net/hse/2014/socialnetworks/papers/2000SiamRev.pdf>

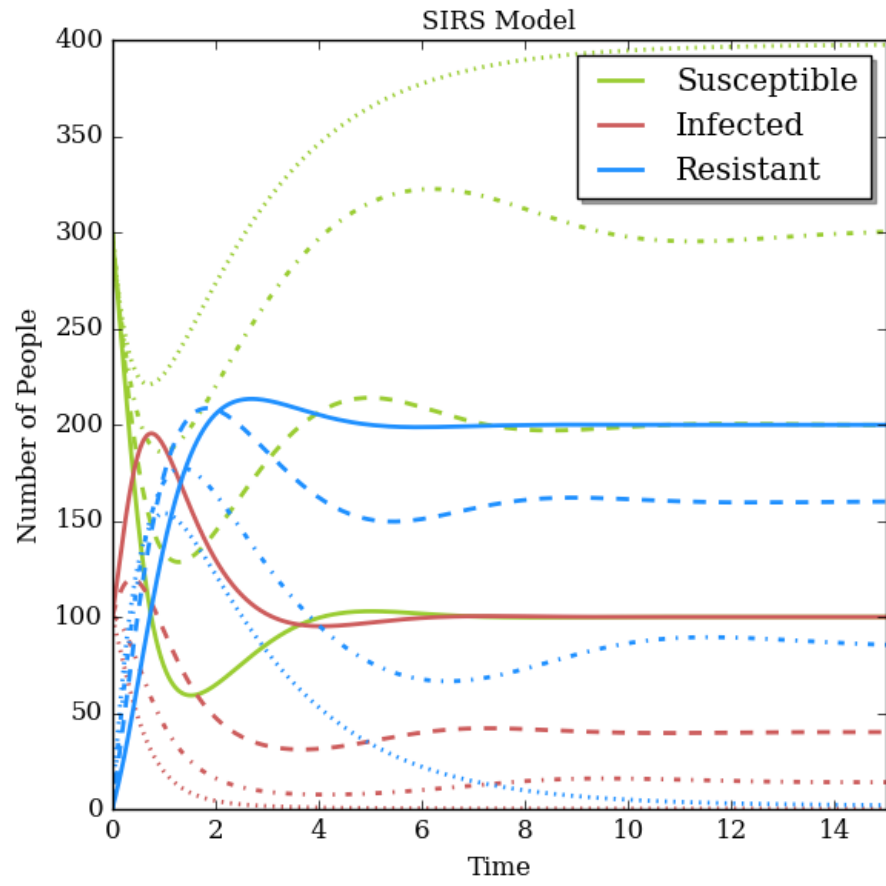


FIG. 1: The number of people in each category for all populations A (solid), B (dashed), C (dot-dashed), and D (dotted).

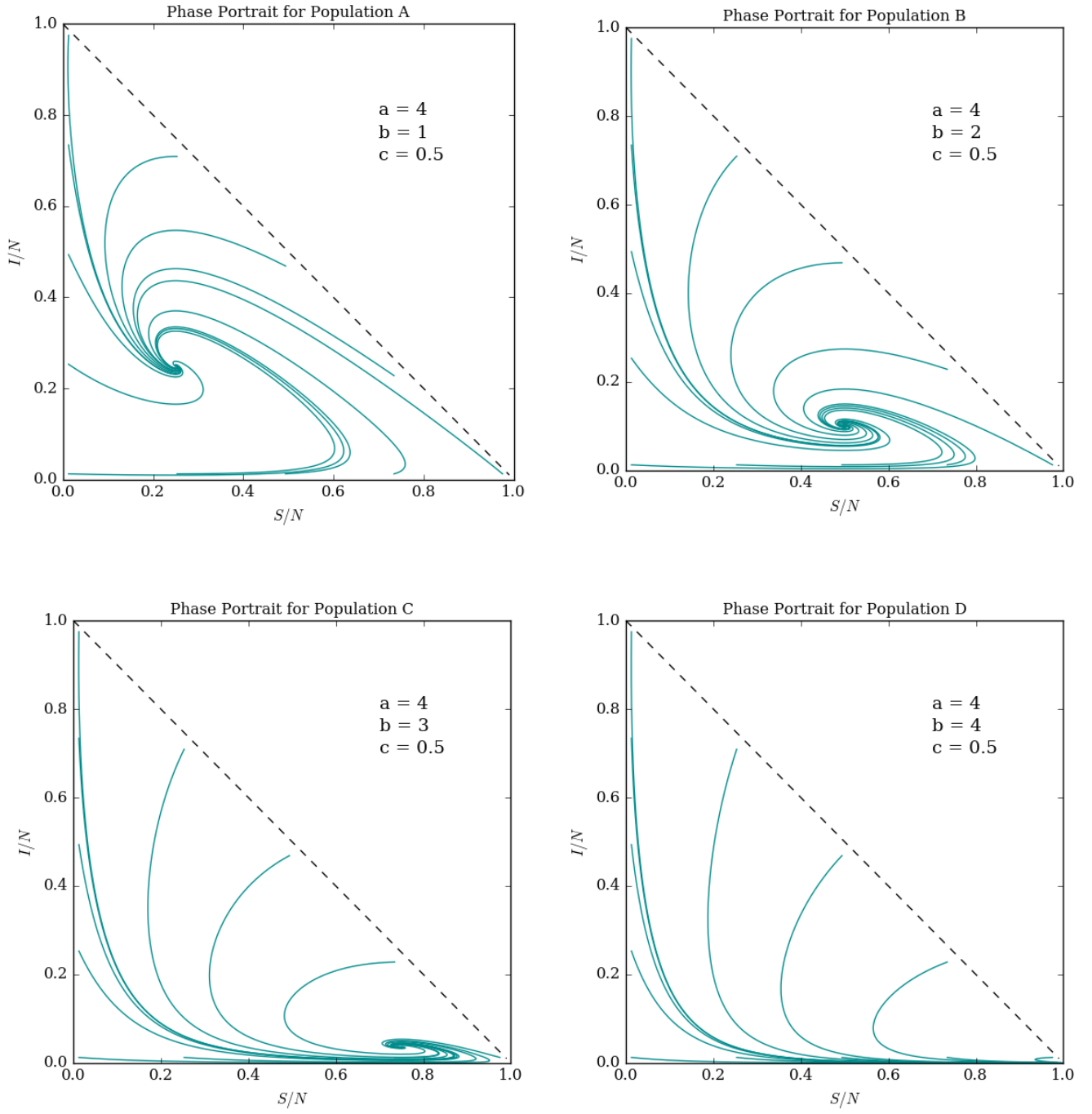


FIG. 2: Phase portraits for all populations A, B, C, and D.

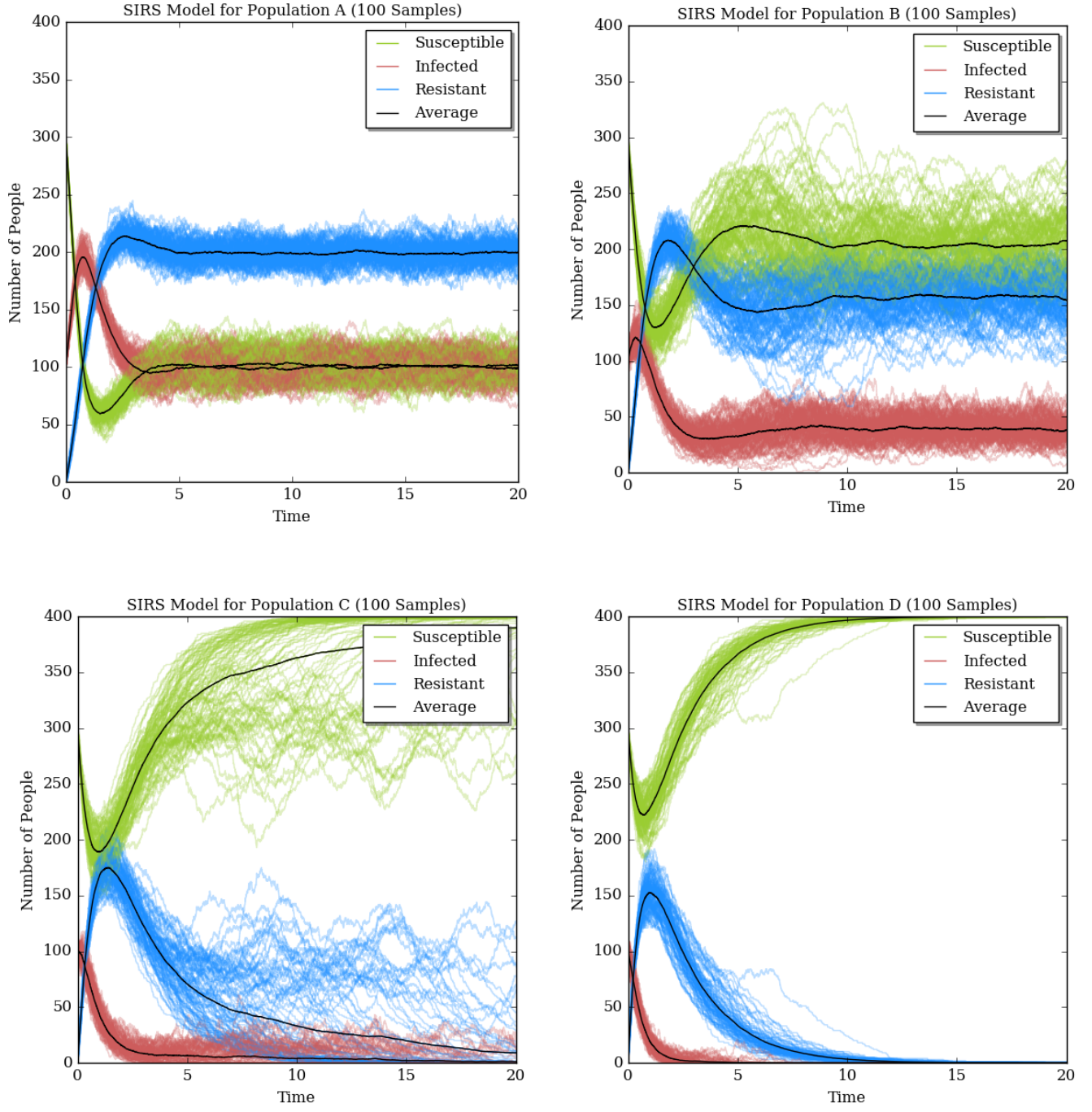


FIG. 3: Monte Carlo simulation with 100 samples for all populations A, B, C, and D.