Problem set 3: Stochastic dynamics in large but finite populations

Erik Norlin & Mattias Wiklund

20 June 2023

A)

We begin by simplifying the dynamics from 2D to 1D by setting $S = N - I$

$$\Rightarrow \begin{cases} \dfrac{dI}{dt} = \dfrac{\alpha}{(N-I)+I}(N-I)I - \beta I & (1) \\ \dfrac{d(N-I)}{dt} = -\dfrac{\alpha}{(N-I)+I}(N-I)I + \beta I & (2) \end{cases}$$

$$\Rightarrow \begin{cases} \dfrac{dI}{dt} = \dfrac{\alpha}{N}(N-I)I - \beta I & (3) \\ -\dfrac{dI}{dt} = -\dfrac{\alpha}{N}(N-I)I + \beta I & (4) \end{cases}$$

We find that eq. 3 and 4 are the same equation. To find the steady states of the dynamics we set $dI/dt = 0$ and solve for I^*.

$$\Rightarrow \frac{dI}{dt} = \alpha I\left(1 - \frac{I}{N}\right) - \beta I = 0 \tag{5}$$

$$\Rightarrow I_1^* = 0,\ I_2^* = N\left(1 - \frac{\beta}{\alpha}\right) \tag{6}$$

From the task at hand we have that α and β are both positive constants and since the number of infectives I cannot be negative we must have that $\alpha > \beta > 0$. We perform linear stability analysis around the fixed points to predict the stability of the steady states.

$$\frac{d}{dI}\frac{dI}{dt}\bigg|_{I=I^*} = \alpha - \frac{2\alpha I}{N} - \beta\bigg|_{I=I^*} \Rightarrow I_1^* = \alpha - \beta,\ I_2^* = \beta - \alpha \tag{7}$$

Since $\alpha > \beta > 0$ we have that $I_1^* > 0$ (unstable) and $I_2^* < 0$ (stable). Thus, for the infection to sustain ad infinitum the parameter values must be $\alpha > \beta > 0$ because this makes I_2^* stable and positive, i.e. I_2^* number of infectives will be sustained as $t \to \infty$.

B)

To derive the master's equation we first label the probability to observe n number of infectives at time t as $\rho_n(t)$. The probability to have n number of infectives at time $t+\delta t$ becomes

$$\begin{aligned}\rho_n(t+\delta t) &= \rho(t)_n + [\text{influx}] - [\text{outflux}] \\ &= \rho(t)_n + [b_{n-1}\rho_{n-1}(t)\delta t + d_{n+1}\rho_{n+1}(t)\delta t] - [(b_n + d_n)\rho_n(t)\delta t]\end{aligned} \tag{8}$$

Where $b_n = \alpha n(1 - n/N)$ is the rate of new infections and $d_n = \beta n$ is the rate of recovery. As $\delta t \to 0$ we get

$$\frac{d\rho_n(t)}{dt} \approx \frac{\rho_n(t+\delta t) - \rho_n(t)}{\delta t} = b_{n-1}\rho_{n-1}(t) + d_{n+1}\rho_{n+1}(t) - (b_n + d_n)\rho_n(t) \tag{9}$$

We make the assumption that in the limit of $N \to \infty$ we get the expected value of I due to the law of large numbers as following

$$I = \langle n \rangle|_{N\to\infty} = \sum_{n=0}^{\infty} n\rho_n(t) \tag{10}$$

$$\Rightarrow \frac{dI}{dt} = \frac{d}{dt}\sum_{n=0}^{\infty} n\rho_n(t) = \sum_{n=0}^{\infty}[nb_{n-1}\rho_{n-1}(t) + nd_{n+1}\rho_{n+1}(t) - n(b_n + d_n)\rho_n(t)] \tag{11}$$

It becomes problematic to have $n-1$ and $n+1$ in the equation. We relabel n for only the two first terms. For the first term we get

$$\sum_{n=0}^{\infty} nb_{n-1}\rho_{n-1}(t) = \{n' = n-1\} = \sum_{n'=-1}^{\infty} (n'+1)b_{n'}\rho_{n'}(t) \tag{12}$$

The probability to have a negative number of infectives is 0. We relabel $n' = n$ and get the first term as

$$\sum_{n=0}^{\infty}(n+1)b_n\rho_n(t) \tag{13}$$

We relabel the second term

$$\sum_{n=0}^{\infty} nd_{n+1}\rho_{n+1}(t) = \{n' = n+1\} = \sum_{n'=1}^{\infty}(n'-1)d_{n'}\rho_{n'}(t) = \sum_{n'=0}^{\infty}(n'-1)d_{n'}\rho_{n'}(t) + d_0\rho_0(t) \tag{14}$$

$d_0\rho_0(t) = 0$ and relabelling $n' = n$ the second term becomes

$$\sum_{n=0}^{\infty}(n-1)d_n\rho_n(t) \tag{15}$$

The third term persists the same. Replacing the first and second term in the master eq. 11 with 13, 15, and then simplifying we get

$$\frac{dI}{dt} = \sum_{n=0}^{\infty} \rho_n(t)(b_n - d_n) = \sum_{n=0}^{\infty} \rho_n(t) \left[\alpha n \left(1 - \frac{n}{N}\right) - \beta n\right] =$$

$$= \alpha \sum_{n=0}^{\infty} n\rho_n(t) - \frac{\alpha}{N} \sum_{n=0}^{\infty} n^2 \rho_n(t) - \beta \sum_{n=0}^{\infty} n\rho_n(t) = \alpha I - \frac{\alpha}{N} I^2 - \beta I =$$

$$= \frac{\alpha I(N - I)}{N} - \beta I = \frac{\alpha SI}{S + I} - \beta I. \tag{16}$$

Here we use the assumption from eq. 10 and substitute $\sum_{n=0}^{\infty} n\rho_n(t)$ with I. In addition to this we make the assumption that $\langle n^2 \rangle = \sum_{n=0}^{\infty} n^2 \rho_n(t) = \langle n \rangle^2 = I^2$. This is valid if $\mathrm{Var}(n) = 0$, which is true in the limit of $N \to \infty$. As we can see, we obtain the deterministic dynamics eq. 5. The way the stochastic model differ from the deterministic model is that the infection in the stochastic model will eventually go from the stable steady state to extinct due to fluctuations in the dynamics, even though the steady state is "stable". That is why this steady state in these stochastic dynamics is referred to a *quasi-steady state* because it is sort-of-stable. For the deterministic model however, this is not the case because there are no fluctuations in this model, hence the name deterministic. In the deterministic model, the infection will sustain ad infinitum.

C)

The probability distributions $P(t_b)$ and $P(t_d)$ of the times until the next infection (t_b) and recovery (t_d) were generated by iterating over time, incrementing with δt, and sampling the time (t_b) and (t_d) when a random drawn number $r \in U[0, 1]$ fulfilled $r < b_n \delta t$ and $r < d_n \delta t$ for respective distribution, where b_n and d_n are the rates of new infection and recovery respectively. $P(t_b)$ and $P(t_d)$ were generated for three sets of b_n and d_n where they took the values $\{b_n = 0.1, d_n = 0.2\}$, $\{b_n = 1, d_n = 2\}$, and $\{b_n = 10, d_n = 5\}$ that can be seen in the six figures below.

For each rate b_n and d_n in a given set, when done sampling, 100 evenly distributed values of the probability density function were obtained by making a histogram with 100 bins of the sampled points and setting the density property of the histogram to true. These values were then fitted to a regression line and, to compare, plotted against the analytical exponential probability density function $\lambda e^{-\lambda t}$ where λ is the given rate. The end result was that the sampled distributions $P(t_b)$ and $P(t_d)$ matched the analytical expression for every set of rates, showing that the distributions exponentially decays with time scaling to $\sim -\lambda t$.

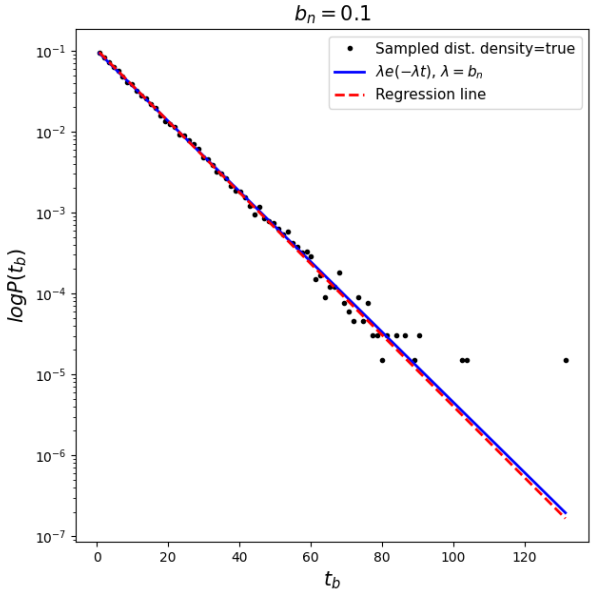


Figure 1: $\log P(t_b)$ against t_b when $b_n = 0.1$. Regression line of sampled $P(t_b) \approx \lambda e^{-\lambda t}$.

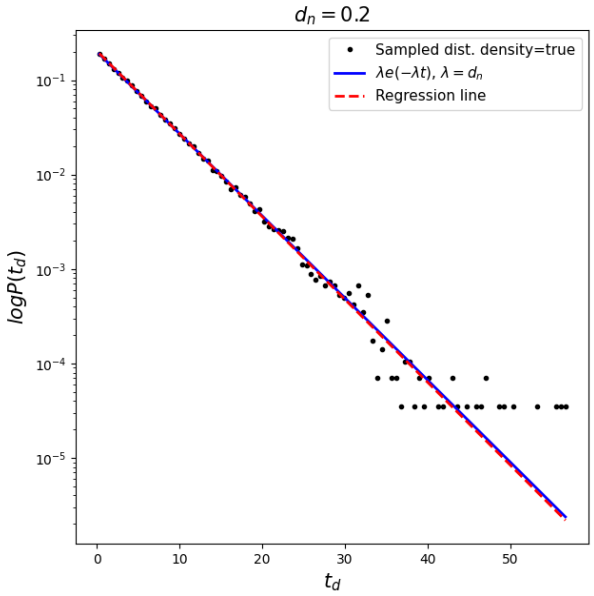


Figure 2: $\log P(t_d)$ against t_d when $d_n = 0.2$. Regression line of sampled $P(t_d) \approx \lambda e^{-\lambda t}$.

In the set of $\{b_n = 0.1, d_n = 0.2\}$, we can see in figure 1 and 2 that the regression lines of the sampled distributions matches the analytical expression of $\lambda e^{-\lambda t}$. The sampled points at large t deviate further from the distribution lines because these points are more difficult to sample accurately. To make them more accurate one would have to sample more points over more runs. Since both lines are linear and the y-axis is scaled to log, we can conclude that the sampled distributions decays exponentially over time scaling to $\sim -\lambda t$.

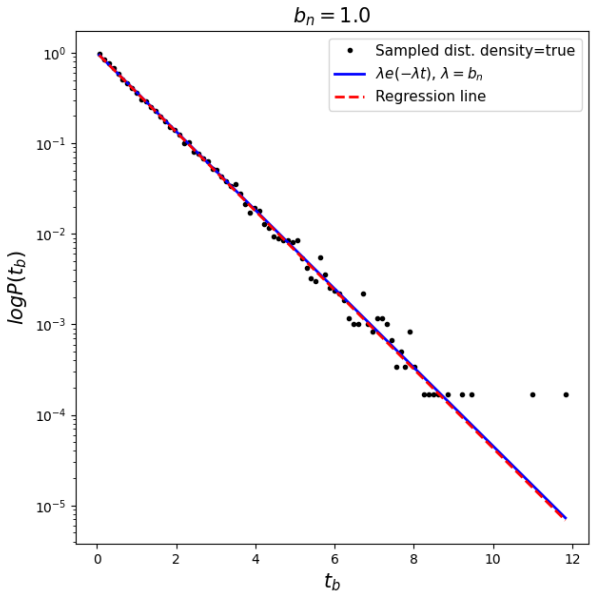


Figure 3: $\log P(t_b)$ against t_b when $b_n = 1$. Regression line of sampled $P(t_b) \approx \lambda e^{-\lambda t}$.

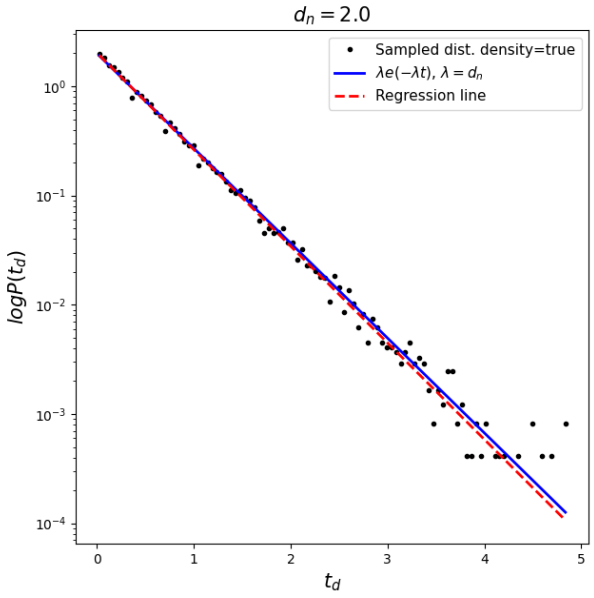


Figure 4: $\log P(t_d)$ against t_d when $d_n = 2$. Regression line of sampled $P(t_d) \approx \lambda e^{-\lambda t}$.

In the set of $\{b_n = 1, d_n = 2\}$, we can observe from figure 3 and 4 similar behaviour as before; that the sampled points at large t deviate further from the distribution lines, and that the regression lines of the sampled distributions decays exponentially over time scaling to $\sim -\lambda t$.

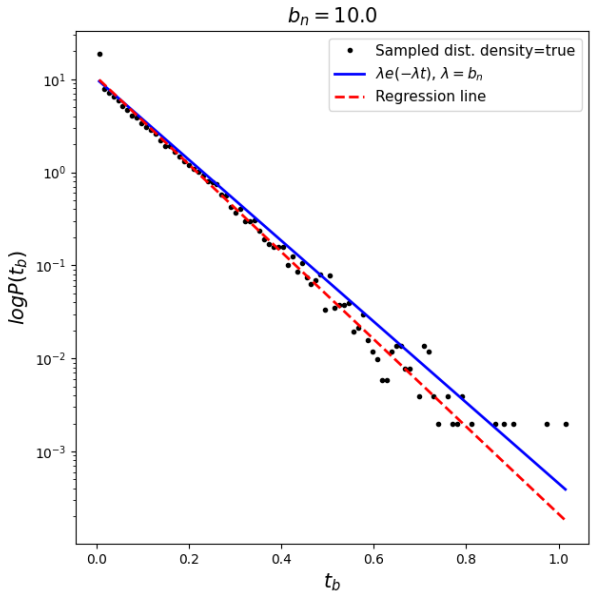


Figure 5: $\log P(t_b)$ against t_b when $b_n = 10$. Regression line of sampled $P(t_b) \approx \lambda e^{-\lambda t}$.

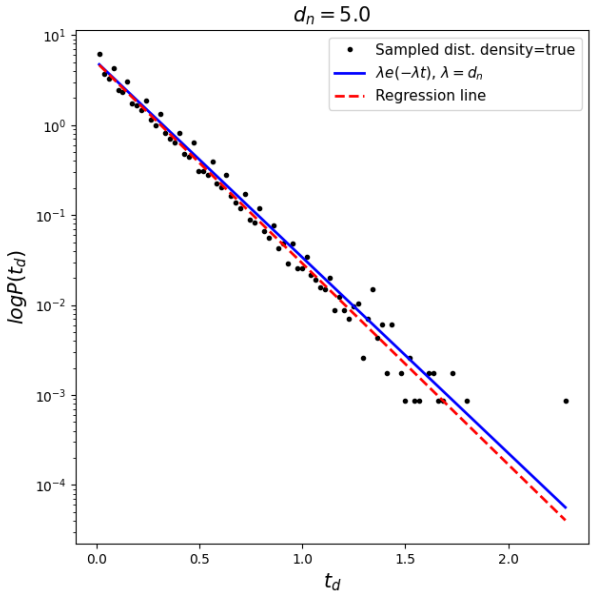


Figure 6: $\log P(t_d)$ against t_d when $d_n = 5$. Regression line of sampled $P(t_d) \approx \lambda e^{-\lambda t}$.

In the set of $\{b_n = 10, d_n = 5\}$, we can see in figure 5 and 6 that we have similar behaviour once again. This time however, the sampled points deviates much further from the distribution lines than from the first set we investigated. This is because d_n and b_n are much larger here and the size of these rates will greatly affect how the sampled points are chosen (as mentioned before, sample t if $r < b_n \delta t$ and $r < d_n \delta t$ for respective distribution, where $r \in U[0, 1]$). If the rates are large then there will be a higher probability of sampling t, meaning that large t-values will be sampled less and will therefore give a less accurate

sampled distribution. This can be observed by looking at the regression lines, even though they do approximate the analytical expression $\lambda e^{-\lambda t}$, they do not approximate $\lambda e^{-\lambda t}$ as well as with smaller rates. That being said, for this set the regression lines of the sampled distributions still decays exponentially over time scaling to $\sim -\lambda t$.

Since we have confirmed that $P(t_b)$ and $P(t_d)$ matches $\lambda e^{-\lambda t}$ we can say that they are exponential distributions, where in this case, as previously mentioned, λ is the rate of new infection or new recovery. The analytical expression for the average of an exponential distribution is $1/\lambda$. In table 1 and 2 we can see the analytical and the numerical average times until a new infection and recovery respectively for the each of previously investigated sets for b_n and d_n. The results show that the numerical and the analytical averages are approximately the same.

Table 1: Average time [s] until new infection for the investigated infection rates b_n.

Infection rate, b_n	Analytical, $1/b_n$	Numerical
0.1	10	9.93
1	1	0.99
10	0.1	0.09

Table 2: Average time [s] until new recovery for the investigated recovery rates d_n.

Recovery rate, d_n	Analytical, $1/d_n$	Numerical
0.2	5	4.99
2	0.5	0.49
5	0.2	0.19

D)

For high population sizes N, we can approximate the lifetime of the quasi-steady state as

$$T_{\text{ext}} \approx k e^{N(\log r_0 - (1-1/r_0))} \tag{17}$$

where k is some prefactor. Analysing this closer, we fixed the reproductive rate $r_0 = \alpha/\beta$ to $r_0 = 1.3$ ($\alpha = 1.3$, $\beta = 1$) and looked at the average T_{ext} (over 2500 runs) for $N = 100, 110, 120, 130, 140$. The model was simulated using the Gillespie algorithm with time step $dt = 0.01$ and initial condition $n_0 = I^* = N(1 - \beta/\alpha)$.

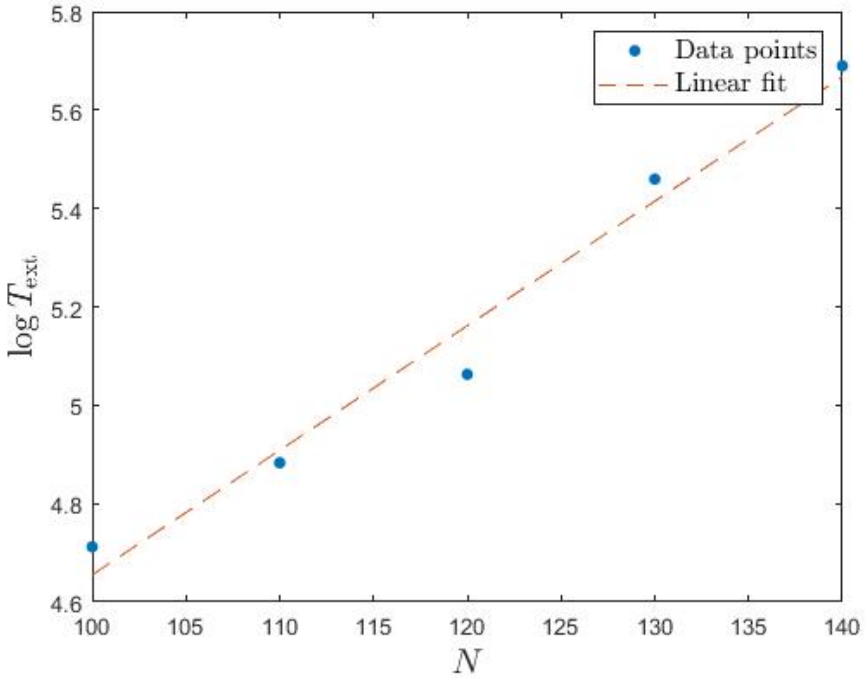


Figure 7: Logarithm of T_{ext} as a function of N for $N = 100, 110, 120, 130, 140$. Data points from simulation are shown in the blue and a linear fit is shown in red.

When performing a linear regression for $\log T_{\text{ext}}$, we obtain

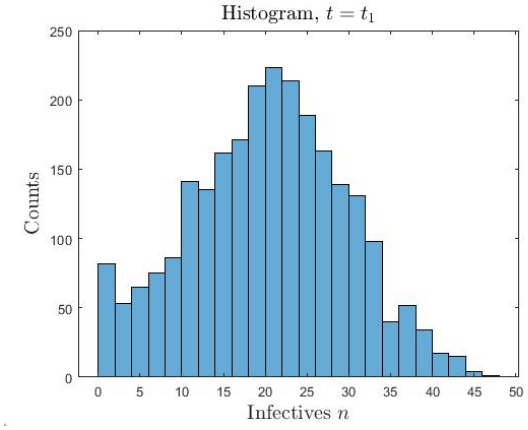
$$\log T_{\text{ext}} \approx 0.03N + 2.12, \tag{18}$$

resulting in

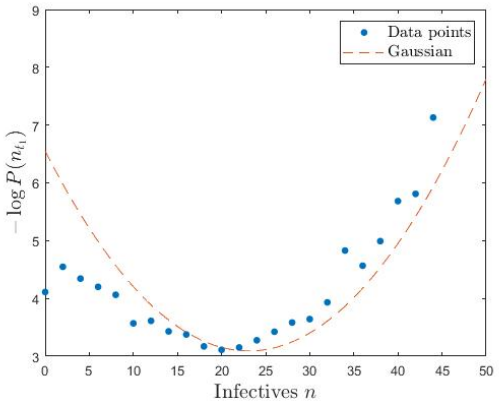
$$T_{\text{ext}} \approx 8.35e^{0.03N}. \tag{19}$$

Comparing to Equation (17), we note that $\log r_0 - (1 - 1/r_0) = \log 1.3 - (1 - 1/1.3) \approx 0.03$. This indicates that the approximation in Equation (17) works rather well, with $k \approx 8.35$ in this case. An even better linear fit could be obtained by analysing more, and higher, N values.

We now choose $N = 100$ and therefore $T_{\text{ext}} \approx 105$ for the actual simulations. We focused on the times $t_1 = 10 < T_{\text{ext}}$, $t_2 = 100 \approx T_{\text{ext}}$ and $t_3 = 200 > T_{\text{ext}}$. The results are displayed below.

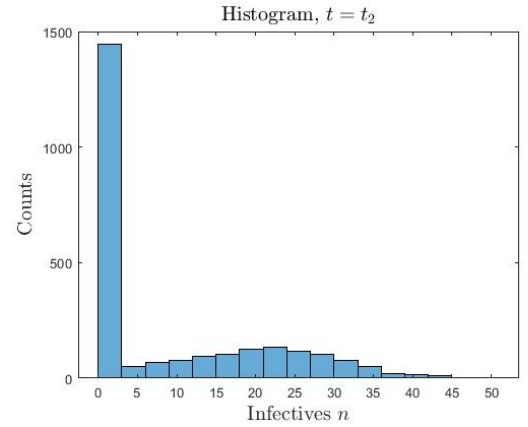


(a) Histogram for the probability distribution $P(n_{t_1})$. Note that the y axis is not normalised, but displays "counts".

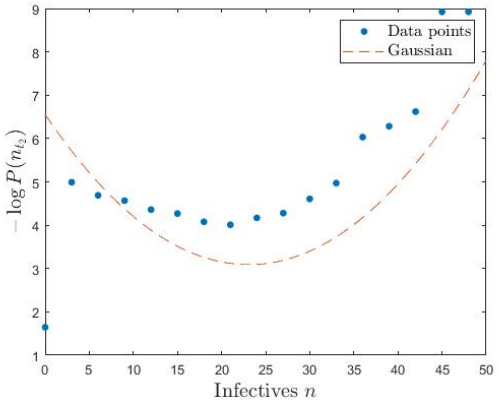


(b) Minus logarithm of the probability distribution and Gaussian approximation ($\mu = n_0$, $\sigma^2 = N/r_0$).

Figure 8: Probability distribution $P(n_t)$ for $t = t_1$.

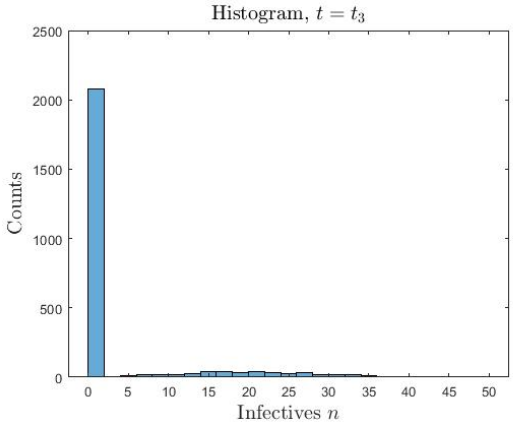


(a) Histogram for the probability distribution $P(n_{t_2})$. Note that the y axis is not normalised, but displays "counts".

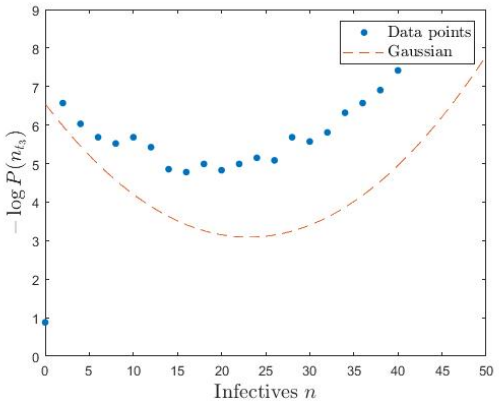


(b) Minus logarithm of the probability distribution and Gaussian approximation ($\mu = n_0$, $\sigma^2 = N/r_0$).

Figure 9: Probability distribution $P(n_t)$ for $t = t_2$.



(a) Histogram for the probability distribution $P(n_{t_3})$. Note that the y axis is not normalised, but displays "counts".



(b) Minus logarithm of the probability distribution and Gaussian approximation ($\mu = n_0$, $\sigma^2 = N/r_0$).

Figure 10: Probability distribution $P(n_t)$ for $t = t_3$.

Note that in the lecture notes, the distribution $P(n_t)$ is compared to a Gaussian distribution with mean n_0/N and variance $1/(r_0 N)$. In our units (we set $I = n$, rather than $I = n/N$), the mean is rescaled according to

$$E(n) = NE\left(\frac{n}{N}\right) = N\frac{n_0}{N} = n_0. \tag{20}$$

The variance is rescaled according to

$$\mathrm{Var}(n) = N^2\mathrm{Var}\left(\frac{n}{N}\right) = N^2\frac{1}{r_0 N} = \frac{N}{r_0}. \tag{21}$$

As we can see in the histograms in Figures 8a, 9a and 10a, the average number of infectives is clearly lower (the bin containing $n = 0$ grow) for $t_2 \approx T_{\text{ext}}$ and $t_3 > T_{\text{ext}}$, compared to $t_1 < T_{\text{ext}}$. This is expected given that the simulation started at $n_0 = I^*$, and we are therefore, with higher probability, in the quasi-steady state for $t < T_{\text{ext}}$.

Analysing the minus logarithm of the normalised probability distribution $P(n_{t_1})$ in Figure 8b, we note that the Gaussian distribution described above (mean n_0 and variance N/r_0) approximates the data relatively well for $t = t_1$. However, it does not capture the heavy left tail of the distribution, which corresponds to a low number of infectives. This is consisted with the theory of this stochastic model, which states the quasi-steady state distribution of the model is in fact non-Gaussian, even though the Gaussian distribution is a relatively useful approximation.

For larger times, $t = t_2$ and $t = t_3$, the approximation does hold as the distribution $P(n_t)$ gets more skewed compared to the Gaussian, which can be seen in Figures 9b and 10b. This is reasonable, given that the approximation was derived for the distribution of the quasi-steady state, whose lifetime is T_{ext}.

A Code

```
import numpy as np
import matplotlib.pyplot as plt
# import matplotlib.backends.backend_qt5agg
import PyQt5
import matplotlib as mpl
# import latex
mpl.use('Qt5Agg')
# plt.rcParams['text.usetex'] = True

location = r'C:\Users\erikn\OneDrive - Chalmers\Computational Biology\CB HW 3'
task = 1

b_list = np.array([0.1, 1, 10])
d_list = np.array([0.2, 2, 5])
dt = 0.01
no_runs = 5*10**4
tb = np.zeros((len(b_list), no_runs))
td = np.zeros_like(tb)

for i in range(len(b_list)):

    b = b_list[i]
    d = d_list[i]

    for j in range(no_runs):

        if (j % 1000 == 0): print(i, j, no_runs)

        # for b
        t = 0
        flag = True

        while flag == True:
            r = np.random.uniform(0,1)
            if r < b*dt:
                tb[i,j] = t
                flag = False
            else:
                t = t + dt

        # for d
        t = 0
        flag = True

        while flag == True:
            r = np.random.uniform(0,1)
```

```
            if r < d*dt:
                td[i,j] = t
                flag = False
            else:
                t = t + dt

# np.save('tb.npy', tb)
# np.save('td.npy', td)
tb = np.load('tb.npy')
td = np.load('td.npy')

task = 0
fontsize = 15
bins = 100
figsize = 7
inf = 60
location = r'C:\Users\erikn\OneDrive - Chalmers\Computational Biology\CB HW 3'


# Plotting tb
plt.figure()
hist1 = plt.hist(tb[task,:], bins=bins, density=True)

tb_shift = (hist1[1][1] - hist1[1][0]) / 2
tb_list = np.zeros_like(hist1[0])

for i in range(len(hist1[1])-1):
    tb_list[i] = hist1[1][i] + tb_shift

bt_list = b_list[task]*np.exp(-tb_list*b_list[task])

coef1 = np.polyfit(tb_list[0:inf], np.log(hist1[0][0:inf]), 1)
poly1 = np.poly1d(coef1)
y1_fitted = np.exp(coef1[1]) * np.exp(coef1[0] * tb_list)

fig1, ax1 = plt.subplots(figsize=(figsize,figsize))
ax1.plot(tb_list, hist1[0], '.', color='black', label='Sampled dist. density=true')
ax1.plot(tb_list, bt_list, '-', color='blue', lw=2, label='$\\lambda e(-\\lambda t)$, $\\lambda =b_n$')
ax1.plot(tb_list, y1_fitted, '--', color="red", lw=2, label='Regression line')
ax1.set_title('$b_n = {}$'.format(b_list[task]), fontsize=fontsize)
ax1.set_yscale('log')
ax1.set_ylabel('$logP(t_b)$', fontsize=fontsize)
```

```
ax1.set_xlabel('$t_b$', fontsize=fontsize)
ax1.set_box_aspect(1)
ax1.legend(loc="upper right", prop={'size': 11})

# plt.legend(loc="upper right", prop={'size': 11})
title = '/3.1b_bn_{}'.format(b_list[task])
plt.savefig(location+title+'.png')

# Plotting td
plt.figure()
hist2 = plt.hist(td[task,:], bins=bins, density=True)

td_shift = (hist2[1][1] - hist2[1][0]) / 2
td_list = np.zeros_like(hist2[0])

for i in range(len(hist2[1])-1):
    td_list[i] = hist2[1][i] + td_shift

dt_list = d_list[task]*np.exp(-td_list*d_list[task])

coef2 = np.polyfit(td_list[0:inf], np.log(hist2[0][0:inf]),
   1)
y2_fitted = np.exp(coef2[1]) * np.exp(coef2[0] * td_list)

fig1, ax2 = plt.subplots(figsize=(figsize,figsize))
ax2.plot(td_list, hist2[0], '.', color='black', label='
   Sampled dist. density=true')
ax2.plot(td_list, dt_list, '-', color='blue', lw=2, label='$
   \\lambda e(-\\lambda t)$, $\\lambda =d_n$')
ax2.plot(td_list, y2_fitted, '--', color="red", lw=2, label='
   Regression line')
ax2.set_title('$d_n = {}$'.format(d_list[task]),  fontsize=
   fontsize)
ax2.set_yscale('log')
ax2.set_ylabel('$logP(t_d)$', fontsize=fontsize)
ax2.set_xlabel('$t_d$', fontsize=fontsize)
ax2.set_box_aspect(1)
ax2.legend(loc="upper right", prop={'size': 11})

# plt.legend(loc="upper right", prop={'size': 11})
title = '/3.1b_dn_{}'.format(d_list[task])
plt.savefig(location+title+'.png')

plt.show()
```

```
clc
clf
clear all

```

```
alpha = 1.3;
beta = 1;
r0 = alpha/beta;
N = 100;
n0 = N*(1-beta/alpha);
dt = 0.01;
Tmax = 1000;
nRuns = 2500;
n = zeros(nRuns,round(Tmax/dt));
n(:,1) = n0;
Text=exp(N*(log(r0)-1+1/r0))

for m=1:nRuns
    if mod(m,100) == 0
        disp(m)
    end
    index_old = 1;
    t = 0;
    while t < Tmax
        if n(m,index_old)<1
            break
        else
            bn = alpha*n(m,index_old)*(1-n(m,index_old)/N);
            dn = beta*n(m,index_old);
            tb = exprnd(1/bn);
            td = exprnd(1/dn);
            t = t + min(tb,td);
            index = floor(t/dt) + 1;
            n(m,index_old + 1:index-1) = n(m,index_old) + 1;
            if tb < td
                n(m,index) = n(m,index_old) + 1;
            else
                n(m,index) = n(m,index_old) - 1;
            end
            index_old = index;
        end
    end
end
%%
clc
ext = [];
for i=1:nRuns
    for j=1:round(Tmax/dt)
        if n(i,j) == 0
            ext = [ext j];
            break
        end
    end
```

```
end
histogram(ext)
mean(ext)*dt
length(ext)/size(n,1)
%%
clc
clf
times = [10 100 200];

n_test = n(:,round(times(3)/dt));

histogram(n_test)
xlabel('Infectives $n$','Interpreter','latex', 'FontSize', 15)
ylabel('Counts','Interpreter','latex', 'FontSize', 15)
title('Histogram, $t=t_3$','Interpreter','latex', 'FontSize', 15)

[N_hist,edges] = histcounts(n_test,'Normalization','pdf');
figure(2)
x = edges(1:end-1);
y = -log(N_hist);
plot(x,y,'.','MarkerSize',15)
hold on
X = linspace(0,50);
mu = n0;
sigma = sqrt(N/r0);
gauss = 1/(sigma*sqrt(2*pi))*exp(-1/2*(X-n0).^2/sigma^2);
loggaus = -log(gauss);
plot(X,loggaus,'--')
xlabel('Infectives $n$','Interpreter','latex', 'FontSize', 15)
ylabel('$-\log{P(n_{t_3})}$','Interpreter','latex', 'FontSize', 15)
legend('Data points','Gaussian','Interpreter','latex', 'FontSize', 12)
```

```
x = [100 110 120 130 140];
y = [111.29 132.02 158.01 235.01 295.86];
y = log(y);
%cftool
plot(x,y,'.','MarkerSize',15)
hold on
p = polyfit(x,y,1)
plot(x,p(1)*x+p(2),'--')

xlabel('$N$','Interpreter','latex', 'FontSize', 15)
ylabel('$\log T_\mathrm{ext}$','Interpreter','latex', '
```

```
    FontSize', 15)
12 legend('Data points','Linear fit','Interpreter','latex', '
    FontSize', 12)
```