University *of Ljubljana*
Faculty *of Mathematics and Physics*

Department of Physics

Stochastic Population Models

9. Task for Model Analysis I, 2023/24

Author: Marko Urbanč
Professor: Prof. Dr. Simon Širca
Advisor: doc. dr. Miha Mihovilovič

Ljubljana, August 2024

Contents

1 Introduction

Today's task is another in the series of population model methods. We will be taking a look at stochastic population models, which are another way of modeling populations. Perhaps in a more realistic way, as they take into account the randomness of the environment and the population itself. The curves produced by these models are not smooth, but rather jagged, as they represent the random fluctuations in the population, which is actually more realistic than the smooth curves produced by deterministic models.

To illustrate the workings of stochastic population models let us consider a simple exponential death-birth model. The model is defined by the following system of differential equations:

$$\frac{\mathrm{d}N}{\mathrm{d}t} = \lambda N - \mu N \,, \tag{1}$$

$$N(0) = N_0 \,, \tag{2}$$

$$N(t) = N_0 e^{(\lambda-\mu)t} \,. \tag{3}$$

thus if we expand the solution in a Taylor series for the exponents we get:

$$\Delta N = \lambda N \Delta t - \mu N \Delta t \,. \tag{4}$$

This is the deterministic model, which we can transform into a stochastic model by sampling the two terms in ΔN from a Poisson distribution with the mean values $\lambda N \Delta t$ and $\mu N \Delta t$. Thus the stochastic model as such:

$$\Delta N = \mathcal{P}(\lambda N \Delta t) - \mathcal{P}(\mu N \Delta t) \,, \tag{5}$$

where $\mathcal{P}(\lambda N \Delta t)$ is a Poisson distribution with the mean value $\lambda N \Delta t$. This is the essence of stochastic population models. We will be applying this method to a few different models and analyzing the results.

Worth mentioning is also the fact that for models as above we can evolve the population in time by using a **Transfer matrix**. This matrix is defined by the transition probabilities between the states of the population. In general it is the block-Toeplitz matrix of a two-scale system which characterizes a class of functions known as *refinable functions* (see [1]). We can formulate the transfer matrix as follows:

$$\mathrm{M} = \begin{pmatrix} 1-\lambda\Delta t & \lambda\Delta t & 0 & \cdots & 0 \\ \mu\Delta t & 1-(\lambda+\mu)\Delta t & \lambda\Delta t & \cdots & 0 \\ 0 & \mu\Delta t & 1-(\lambda+\mu)\Delta t & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} . \tag{6}$$

With that we can evolve the population in time by simply multiplying the population vector with the transfer matrix. The population vector can be thought as the probability distribution of the population in different states. So the n-th element of the vector represents the probability of the population being in the n-th state. Evolving the population in time is then simply done as:

$$\boldsymbol{N}(t+\Delta t) = \mathrm{M}\boldsymbol{N}(t) . \tag{7}$$

From this formulation we can extract some interesting formulae for the population. The derivation of these formulas is explanied in the professors book [2]. The first interesting formula gives us the *extinction probability* of the population which is exactly as it sounds. The formula is:

$$p_0(t) = \begin{cases} \left[\frac{\mu\left(e^{(\lambda-\mu)t}-1\right)}{\lambda e^{(\lambda-\mu)t}-\mu}\right]^N & \text{if } \lambda \neq \mu , \\ \left[\frac{\lambda t}{1+\lambda t}\right]^N & \text{if } \lambda = \mu . \end{cases} \tag{8}$$

We can also derive the *mean population* formula which gives us the average population in time. We obtain the formula by calculating the first moment of the population distribution. The general moment formula is:

$$\langle \mu_i \rangle = \sum_{n=0}^{N} n^i p_n(t) . \tag{9}$$

The mean population formula is then:

$$\langle N(t) \rangle = N_0 e^{(\lambda-\mu)t} . \tag{10}$$

Likewise we can derive the *variance of the population* by calculating the second moment of the population distribution. The general formula for the variance is:

$$\sigma^2 = \langle \mu_2 \rangle - \langle \mu_1 \rangle^2 , \tag{11}$$

from which it follows that the variance of the population is:

$$\sigma^2(t) = \sigma^2(0)e^{2(\lambda-\mu)t} + N_0 e^{(\lambda-\mu)t}\left[e^{(\lambda-\mu)t}-1\right]\frac{\lambda+\mu}{\lambda-\mu} . \tag{12}$$

This formula is useful in the case of a general distribution of the vector $\boldsymbol{N}(t)$. If we know the exact number of individuals in the population at the beginning then $\sigma^2(0) = 0$ and $\langle N(0) \rangle = N_0$, thus the formula simplifies to:

$$\sigma^2(t) = \begin{cases} N e^{(\lambda-\mu)t}\left[e^{(\lambda-\mu)t}-1\right]\frac{\lambda+\mu}{\lambda-\mu} & \text{if } \lambda \neq \mu , \\ 2N\mu t & \text{if } \lambda = \mu . \end{cases} \tag{13}$$

We will use these formulas to analyze the results of a Lotka-Volterra model later on.

2 Task

This task contains a whole 4 subtasks which is quite a lot. Let's not loose any time and get right to it.

2.1 Statistics of Death Time for a simple Death and Death-Birth model

The instructions demand we explore the statistics of the death time of a population in the case of a simple Death model and a Death-Birth model. Since our model is stochastic we expect the death time to be a random variable. The death only model is defined by the following stochastic model:

$$\Delta N = -\mathcal{P}(\beta N \Delta t) \,, \tag{14}$$

where β is the death rate. The death-birth model is defined by the following stochastic model:

$$\Delta N = \mathcal{P}(\beta_b N \Delta t) - \mathcal{P}(\beta_d N \Delta t) \,, \tag{15}$$

where β_b is the birth rate and β_d is the death rate. By default I've set them to be $\beta_b = 4\beta$ and $\beta_d = 5\beta$ as stated in the instructions. β here is just a constant which scales the rates. We've got to take a look at the distribution of the death time for an initial population of $N_0 = 25$ and 250 individuals.

2.2 Transfer matrix for Death-Birth model

We must calculate the transfer matrix for the Death-Birth model and use the formulas derived in the introduction to calculate the extinction probability, mean population and variance of the population. It would be cool to compare these deterministic values to the results of the stochastic model.

2.3 Fluctuation Death: Lotka-Volterra model

The instructions want us to find the average lifetime of a system in the Lotka-Volterra model if we start in a stationary state. In a stationary state the population would never go extinct in the deterministic model, however due to the randomness of our stochastic model the population can go extinct, which is an effect called *fluctuation death*. The system is defined by the following stochastic model:

$$\Delta R = \mathcal{P}(5\alpha R \Delta t) - \mathcal{P}(4\alpha R \Delta t) - \mathcal{P}(\frac{\alpha}{F_0} R F \Delta t) \,, \tag{16}$$

$$\Delta F = \mathcal{P}(4\beta F \Delta t) - \mathcal{P}(5\beta F \Delta t) + \mathcal{P}(\frac{\beta}{R_0} R F \Delta t) \,, \tag{17}$$

where α and β are the birth rates of the rabbits and foxes, respectively. F_0 and R_0 are the initial populations of the foxes and rabbits, again respectively.

2.4 Disease Eradication Time in an SIR model

For the last subtask we see a revival of `mod104` where we are again tasked with simulating an epidemic. This time however we must add a term that allows for recovered individuals to become susceptible again. The system is defined by the following stochastic model:

$$\Delta S = -\mathcal{P}(\beta S I \Delta t) + \mathcal{P}(\gamma I \Delta t) \,, \tag{18}$$

$$\Delta I = \mathcal{P}(\beta S I \Delta t) - \mathcal{P}(\gamma I \Delta t) \,, \tag{19}$$

$$\Delta R = \mathcal{P}(\gamma I \Delta t) - \mathcal{P}(\delta R \Delta t) \,, \tag{20}$$

3 Solution Overview

3.1 Statistics of Death Time

To solve the first subtask I've implemented the stochastic models (14) and (15) as classes in python. The classes are called `DeathModel` and `DeathBirthModel`. As I always say the class-based approach is the best way to go for executing parameter scans. Like demonstrated in `mod108` I also used the `ProcessPoolExecutor` from the `concurrent.futures` module to parallelize the execution of the parameter scans. The results are then saved to a HDF5 ♡ file and plotted separately. This time I really do not have any noteworthy intermediate results to show. The final results are in the next section.

3.2 Transfer matrix for Death-Birth model

I had some trouble with setting up the transfer matrix for it to work properly. I think I managed in the end however. The code for this subsection essentially expands on the code from the previous subsection. Again I have little intermediate results to show. I tried to focus on the most informative plots as time is not on my side. The final results are in the next section.

3.3 Fluctuation Death: Lotka-Volterra model

This subtask was quite a pain to solve since I steered a bit back to my old habits and decided to use distributed computing to gather more data. This was due to the `ProcessPoolExecutor` locking up if it got a very large order of tasks to execute. As is standard I created a class for the Lotka-Volterra model which I used for parameter scans. Since I wanted decent statistics I wanted to get a lot of data points on quite a fine grid of parameters. For this I used the `ray` library which has been nothing short of amazing. It is implemented at a very high level which makes it very very easy to use. This is the case since I'm essentially spawning multiple instances of the same class and running them in parallel. What I used to focus excessively on before was the actual parallelization of the core algorithm which is for the most part considarably harder and entirely unnecessary. The cluster whipped up multiple gigabytes of resultant data which meant I had to abandon my beloved HDF5 files and switch to an even more efficient format from the `zarr` library. My greatest feat was compressing data with a $1 : 300$ compression ratio, which equates to squeezing 15 GB of data into 50 MB. Read/Write times are also approximately $3\times$ faster than with HDF5. I didn't include any of my benchmarks here as I don't think that they are of any interest in terms of the task. As before the final results are in the next section.

3.4 Disease Eradication Time in an SIR model

Analogously to the previous subtask I used the `ray` library to distribute the computation of the SIR model. I also used the `zarr` library for storing the data. I had to unfortunately somewhat limit myself in the number of repeat runs as managing the data was becoming quite the hassle. The whole project folder for `mod109` already exceeds 30 GB and that is only counting the data I decided to keep. I am like a parrot: Results at the end of the next section.

4 Results

I've spent too many days on this task as I've been having some personal issues. Let's get to the results. So I can move on to the next task.

4.1 Statistics of Death Time

The instructions of the first subtask wanted us to explore the relationship between the death time of a population and the algorithm step size ΔT. The results for the Death-only model are shown in Figures 1 and 2 for $N_0 = 25$ and 250, respectively. The rate parameter β was set to 1 for both cases.

Take note that plotted on the axes are the inverse time steps. So the actual value of the time step is $\Delta T = 1/x$. This was done due to the way the instructions provided information. The way I've set up the model is that one can request the number of iterations to simulate for, this means that the only *temporal* data I have is the number of iterations. This is shown on the left-most plot in the figures. The middle plot then shows what the histogram looks like if we convert the number of iterations to time by multiplying it by Δt. Since our aim was also to figure out if there is any relationship between the death time and the algorithm step size Δt I've plotted residuals of the data points in the parameter space on the right-most plot. Fitting a linear model to approximate the skew of the residuals we can see that the death times are not very strongly dependent on the step size. The slight correlation might be an artifact of statistically small sample set. We can see in both cases that very large step sizes (remember, the x-axis values are inverted) lead to the largest residuals. It could be worth investigating this further, however I did not do so. The larger initial population has a larger average death time but the spread seems to remain about the same.

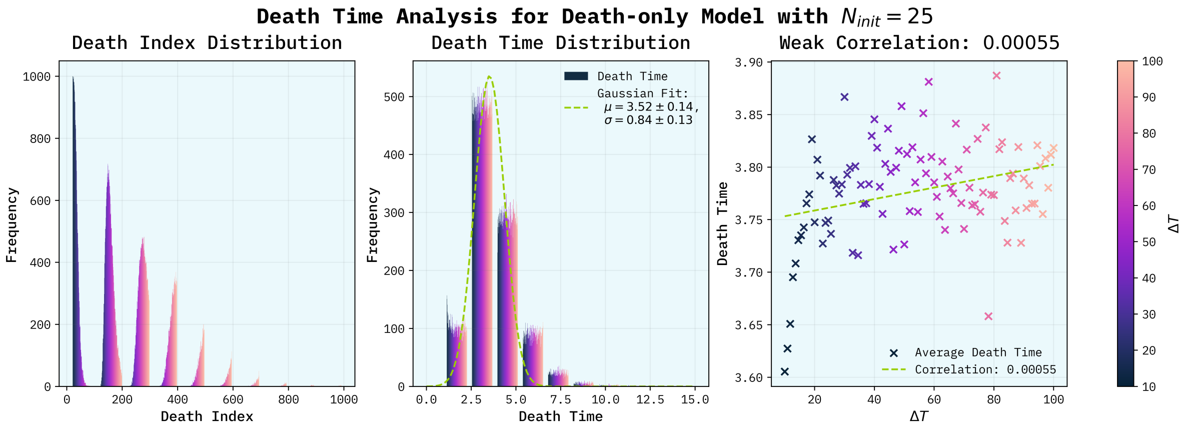

Figure 1: Distribution of the death time for the Death-only model with $N_0 = 25$.

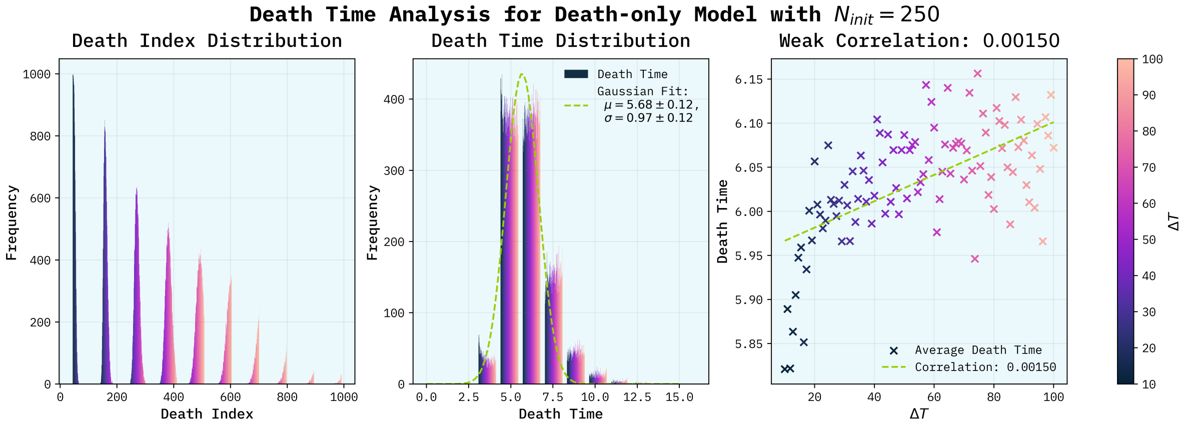

Figure 2: Distribution of the death time for the Death-only model with $N_0 = 250$.

Next I added a birth term to the model as described in the instructions. The results are shown in Figures 3 and 4 for $N_0 = 25$ and 250, respectively. The birth to death rates have been set to a $4:5$ ratio. Here we see an interesting effect, where the average time to death is shorter when adding a birth term to the model. I'm not exactly sure how to interpret this however it seems to be consistent with the fact that even populations with a larger birth rate will eventually go extinct. As observed in the death-only model the spread of the death times is about the same for both initial populations, but the larger population seems to have a slightly larger average death time.

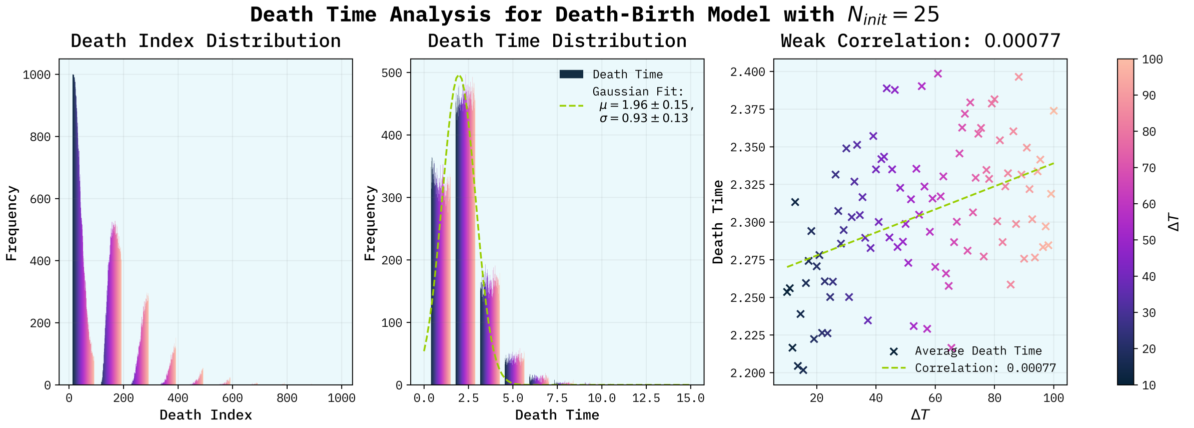

Figure 3: Distribution of the death time for the Death-Birth model with $N_0 = 25$.

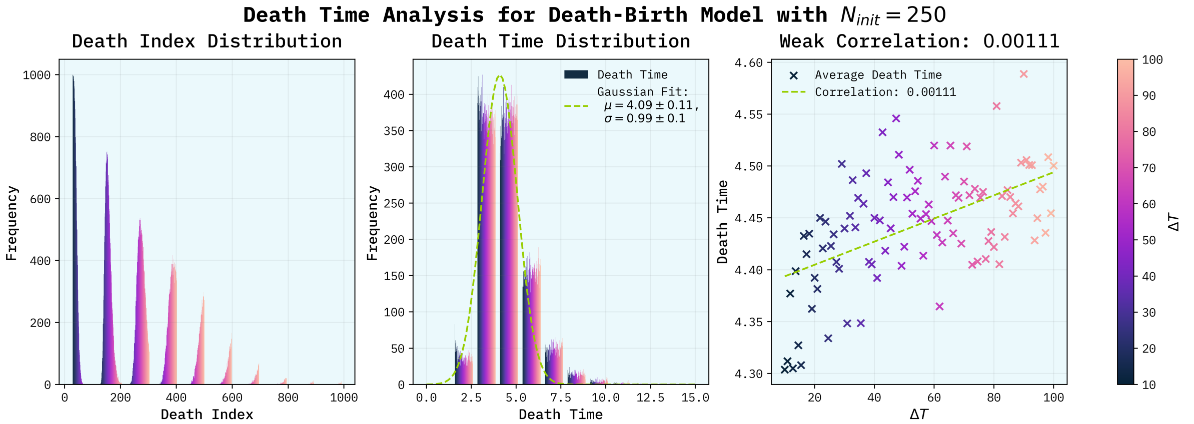

Figure 4: Distribution of the death time for the Death-Birth model with $N_0 = 250$.

For the sake of completeness I also plotted two different values of the birth rate multiplier $\beta = 3$ and $\beta = 5$ for different initial population sizes. Plotted in Figures 5, 6, 7 and 8 are the results for $N_0 = 25$ and 250 for both values of β in the Death-only model. We see that higher values of β lead to shorter average death times. The relative spread also seems to be larger. For larger values of β and larger initial populations the correlation between the death time and the step size seems to be stronger.

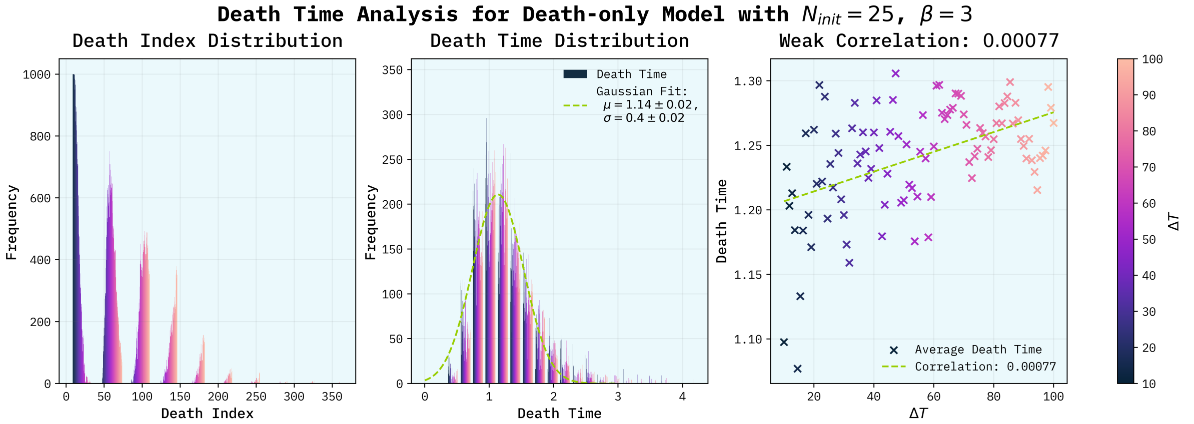

Figure 5: Distribution of the death time for the Death-Birth model with $N_0 = 25$ and $\beta = 3$.

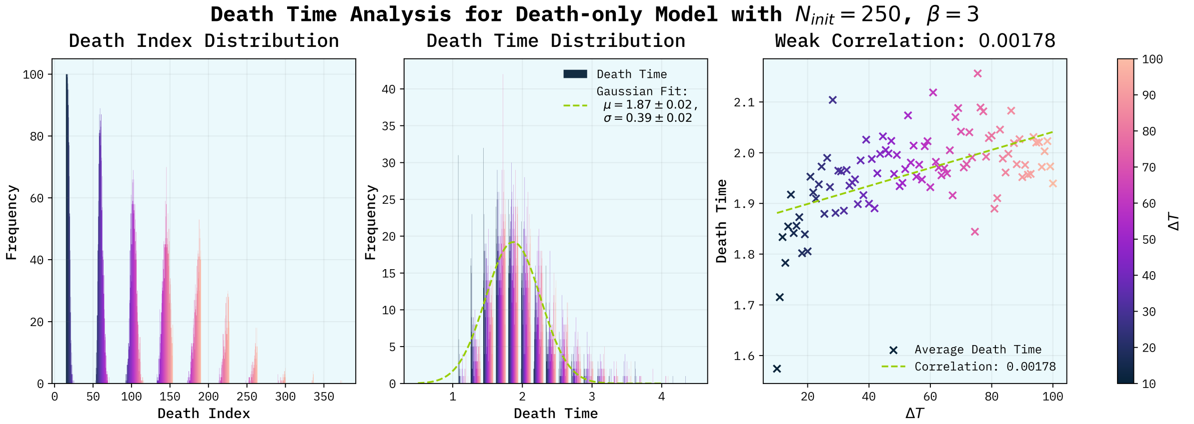

Figure 6: Distribution of the death time for the Death-Birth model with $N_0 = 250$ and $\beta = 3$.

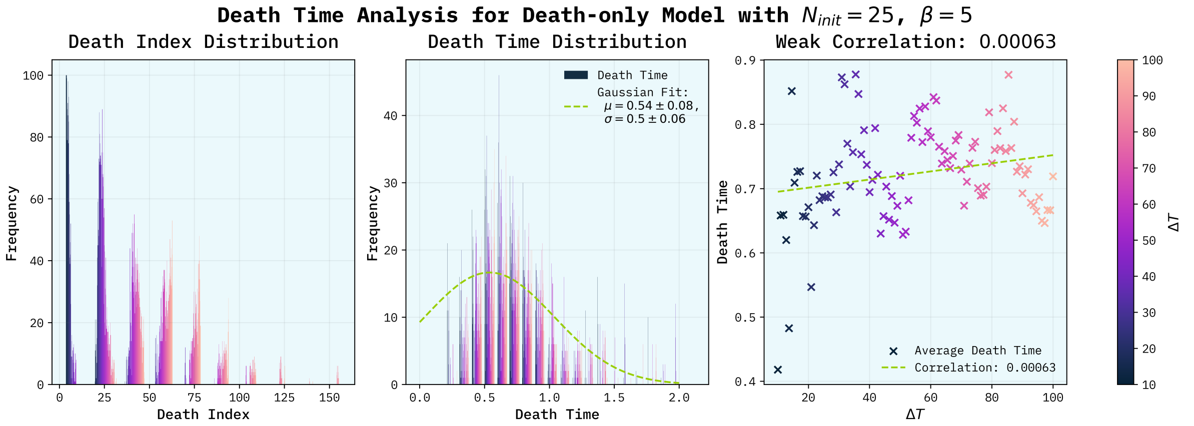

Figure 7: Distribution of the death time for the Death-Birth model with $N_0 = 25$ and $\beta = 5$.

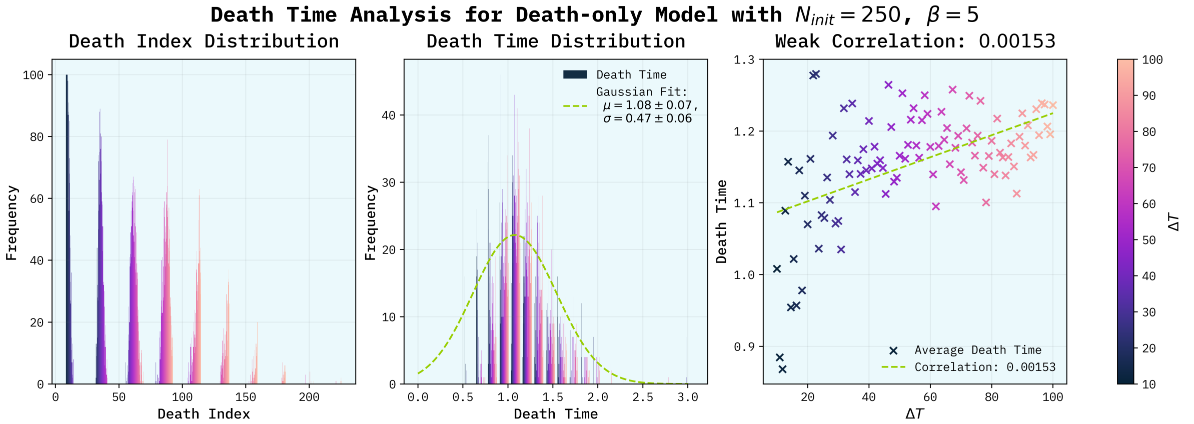

Figure 8: Distribution of the death time for the Death-Birth model with $N_0 = 250$ and $\beta = 5$.

Likewise I have also plotted the results for the Death-Birth model for the same values of β and for $N_0 = 25$, 250 and 2500 as seen in Figures 9, 10 and 11 for $\beta = 3$ and in Figures 12, 13 and 14 for $\beta = 5$. The results are as expected. Death times are even shorter as expected. The spread of the death times seems to be smaller for larger populations and as before the correlation between the death time and the step size seems to be stronger for larger populations.

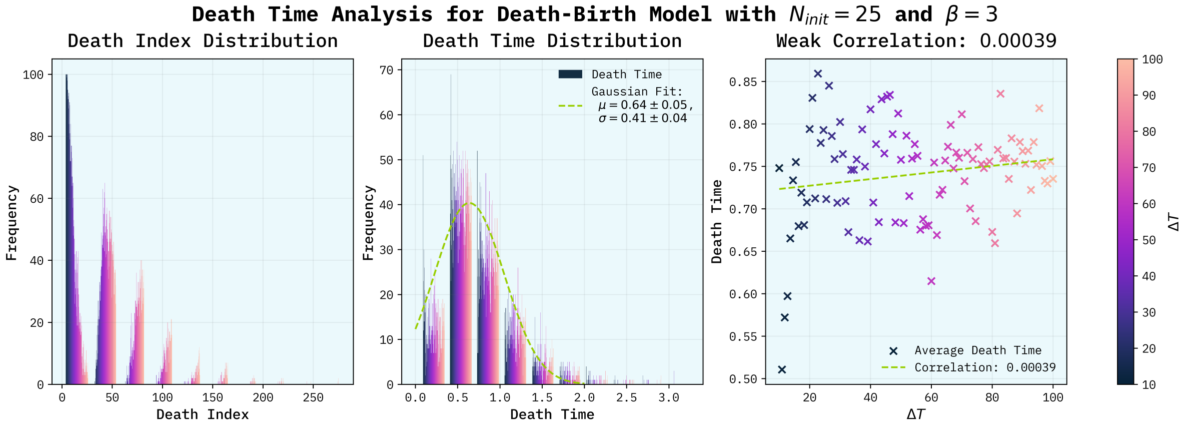

Figure 9: Distribution of the death time for the Death-Birth model with $N_0 = 25$ and $\beta = 3$.

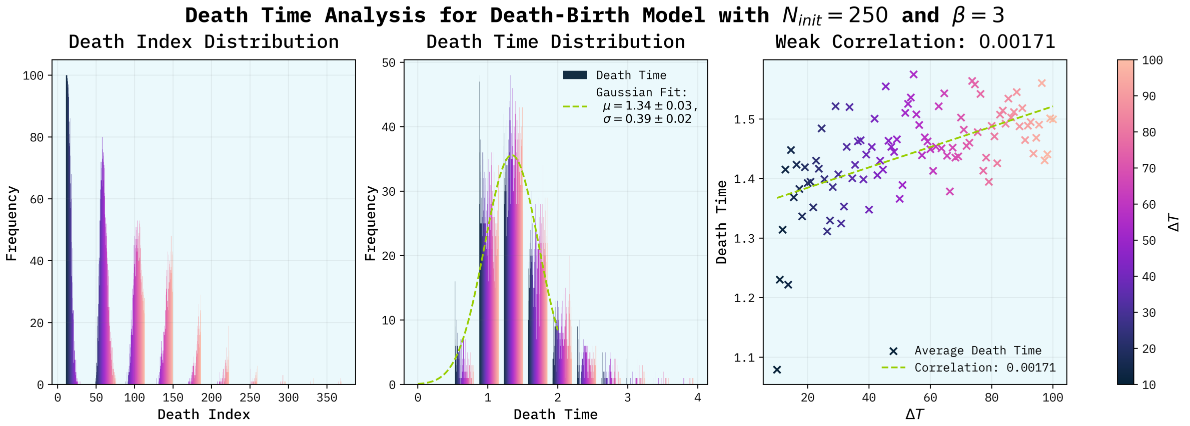

Figure 10: Distribution of the death time for the Death-Birth model with $N_0 = 250$ and $\beta = 3$.

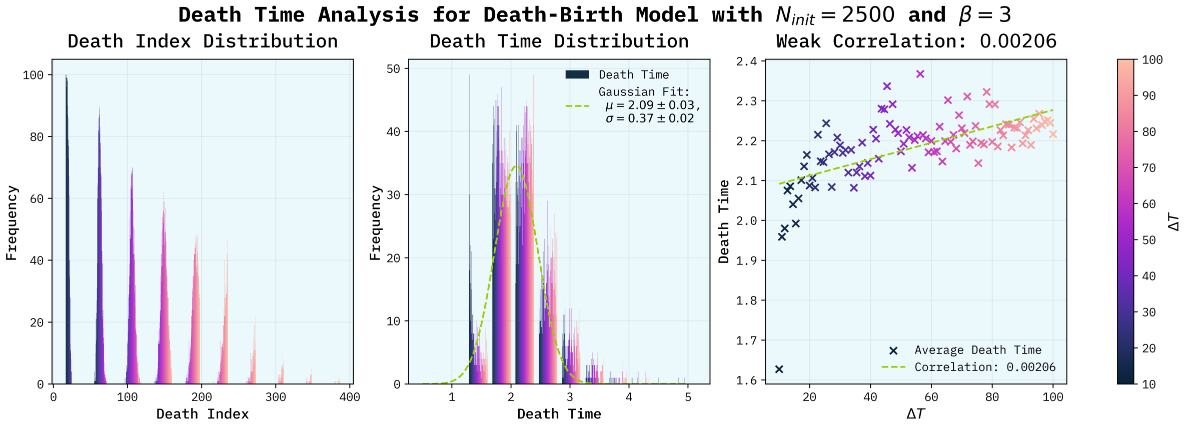

Figure 11: Distribution of the death time for the Death-Birth model with $N_0 = 2500$ and $\beta = 3$.

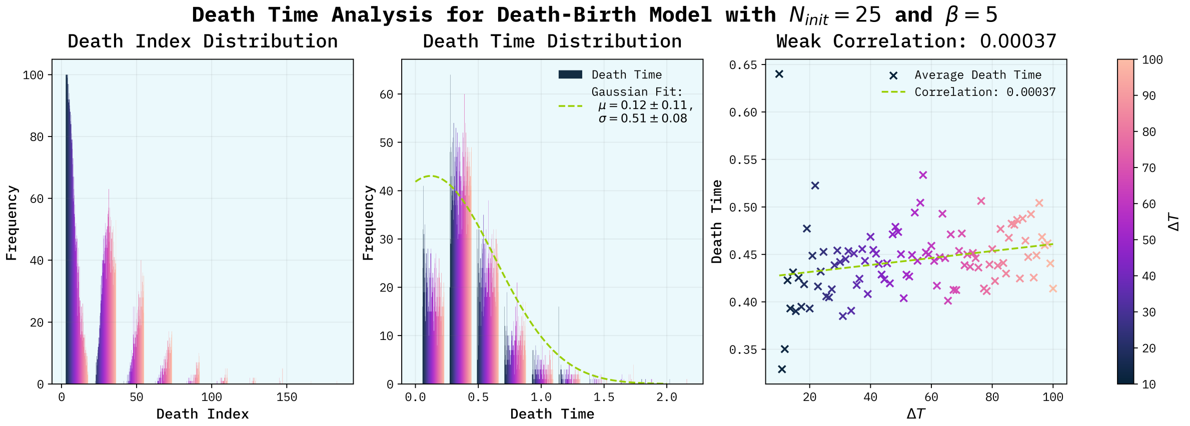

Figure 12: Distribution of the death time for the Death-Birth model with $N_0 = 25$ and $\beta = 5$.

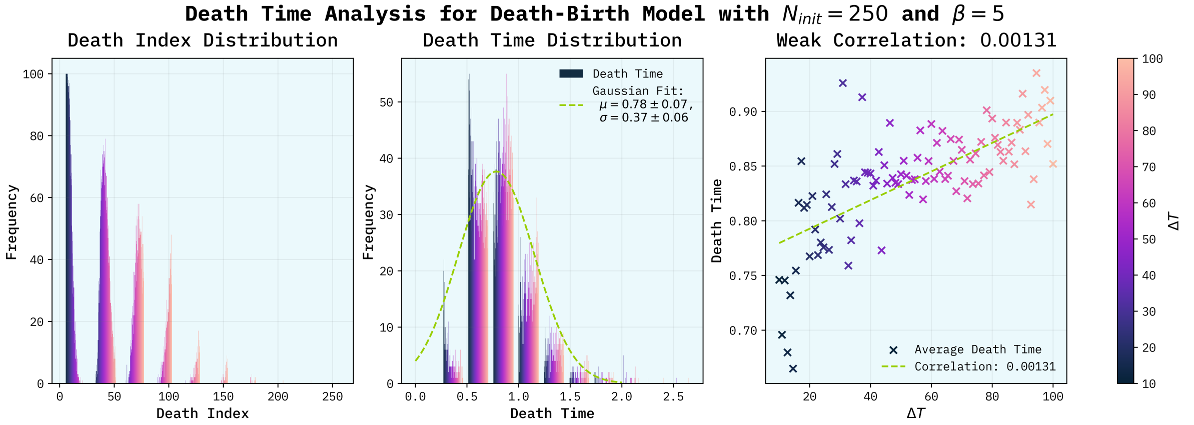

Figure 13: Distribution of the death time for the Death-Birth model with $N_0 = 250$ and $\beta = 5$.

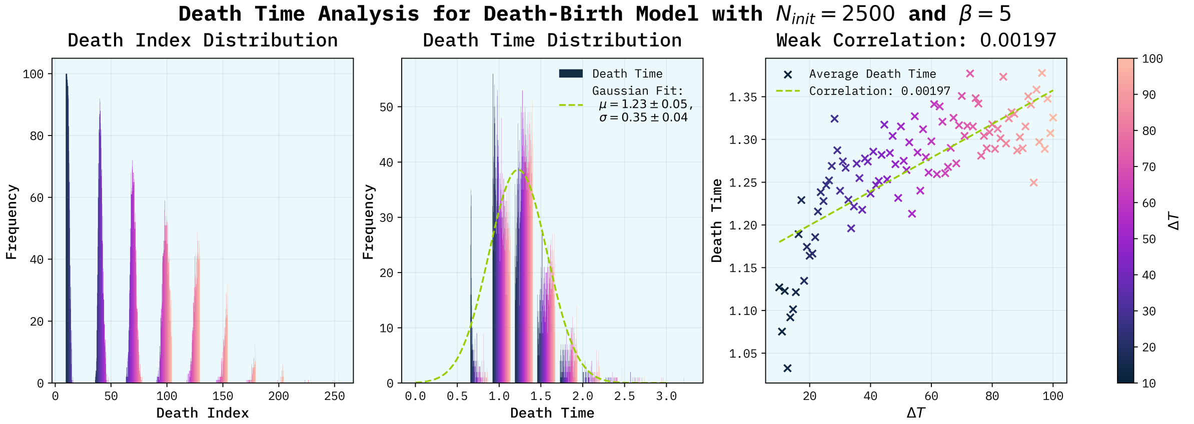

Figure 14: Distribution of the death time for the Death-Birth model with $N_0 = 2500$ and $\beta = 5$.

I think these plots are all pretty menial but for the sake of illustration it might make sense to include them.

4.2 Transfer matrix for Death-Birth model

Since I had a few problems with the transfer matrix I decided to first see if the results of the deterministic model I got using the transfer matrix are consistent with the results of the stochastic model. I used the Death-Birth model from the previous subtask. The results are shown in Figure 15, where we can se that results line up very nicely in both the terms of the average population and the standard deviation of the population. To get a better feeling I also plotted the individual runs of the stochastic model in a very faint blue. There were 100 repeat runs.

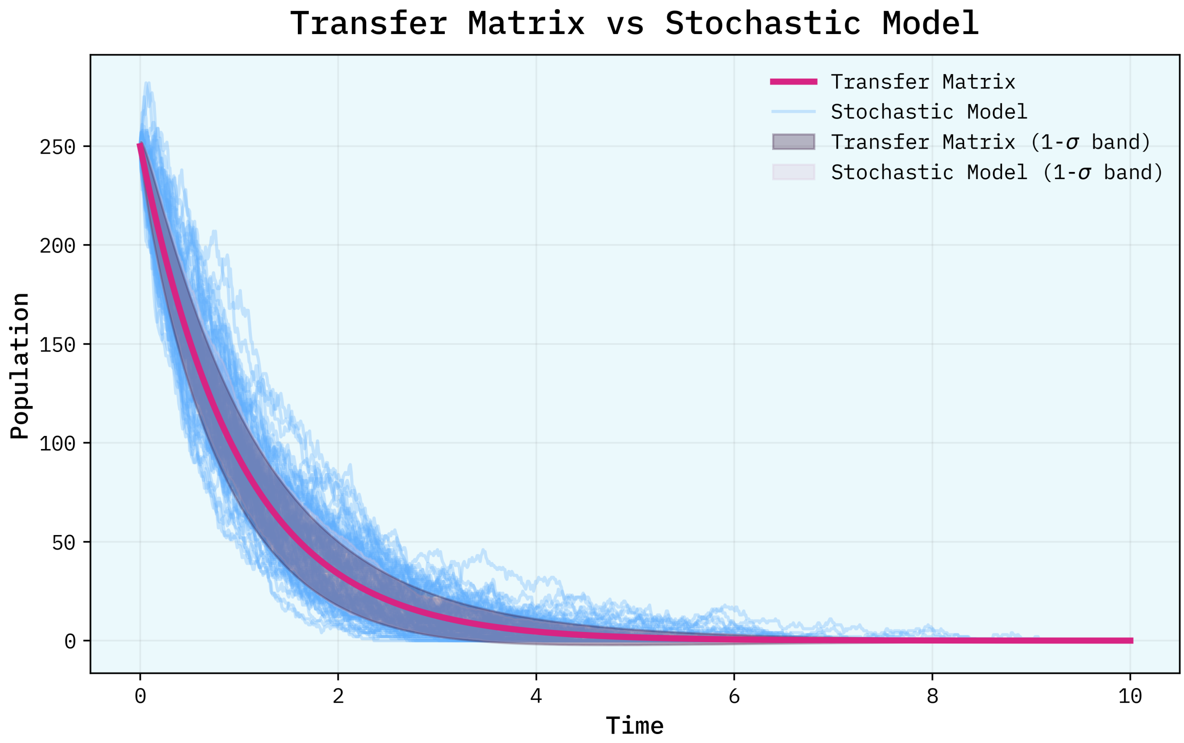

Figure 15: Comparison between stochastic model and deterministic model with the transfer matrix.

Now that we've checked that the matrix is working properly we can move on to the extinction probability, which is the main goal of the subtask. The results are shown in Figure 16. Where we can see the extinction probability over time for different initial population sizes at the birth and death rates of $\beta_b = 4\beta$ and $\beta_d = 5\beta$ as supplied in the instructions. The middle plot shows the probability of extinction for different population sizes in the event that the birth and death rates are equal. Both plots contain a dashed green line which presents the limit (meaning certain extinction) of the extinction probability. The way certain extinction is gradually reached is different for the two scenarios. For the set ratio of rates we get a nice S shaped curve. For the equal rates we get what I can think of as a more logarithmic curve on the lower end of the initial population sizes but a practically instantaneous extinction for larger populations. The right-most plot shows the extinction probability for different birth and death rates expressed as integer multiples of $\beta = 0.1$. We can see that some data is missing in the right-most plot, this is due to the fact that the extinction probability never rose to a non-zero value, meaning that those populations never went extinct.

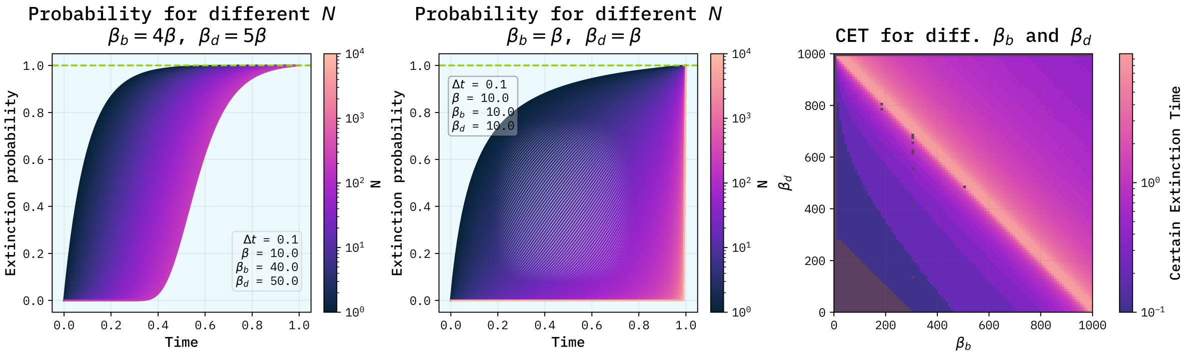

Figure 16: Extinction probability of the population in time.

Since one of the demands the instructions make is to also calculate or show how the mean of the population and the variance of the population evolve in time I've plotted these as well. But before we take a look at the results of that calculation I'd like to show the way the mean and the variance behave for different parameter values. I've tried to create comprehensive plots of how the two moments behave as shown in Figure 17, which may have been *inspired* by a similar plot in the professors book [2]. The results seem logical. A perfectly defined extinction probability leads to a constant mean and a root function like variance.

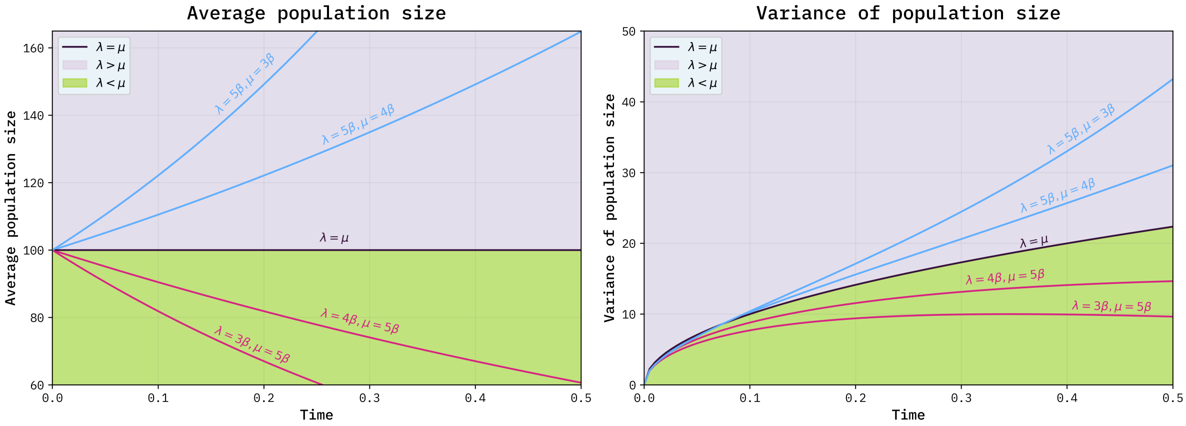

Figure 17: Behaviour of the population average and variance in time.

Now on to two example distributions that are shown in Figure 18 for the Birth-Death model. The upper row contains heatmaps which are made of vertically stacked probability vectors $\boldsymbol{p}(t)$ for two different initial distributions, where as we've said the n-th element of the vector represents the probability of the population being in the n-th state. The lower row contains the calculation of the mean and variance of the population which are calculated according to the formulas provided in the introduction (Eq. 10) and (Eq. 12). The plots seem to match up nicely. I like the method of visualizing the population probability distribution as a heatmap. It gives a nice visual representation of how the distribution moves in time.

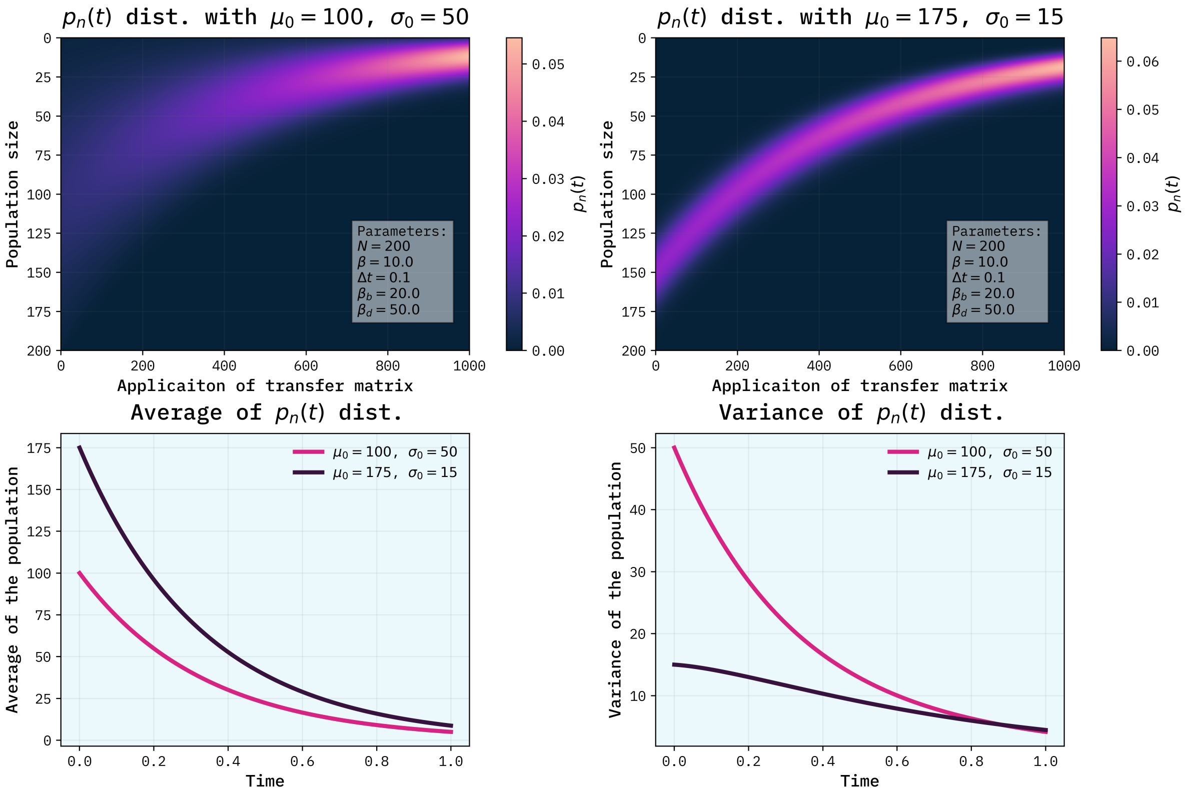

Figure 18: Effective move of the probability distribution $\boldsymbol{p}(t)$ in time.

4.3 Fluctuation Death: Lotka-Volterra model

On to the study of fluctuation deaths in the Lotka-Volterra model for rabbits and foxes. I wanted to get a feel for how the model behaves so I found a pair of parameters for the scaling of rabbit rates and scaling of fox rates that would lead to a population that did not go extinct. Since I used the supplied initial population sizes of $R_0 = 200$ and $F_0 = 50$, which represent a fixed point, the population should in theory never die but due to the stochastic nature of the model it can. In light gray I've plotted some deterministic solutions which were obtained by solving the system of differential equations. The results are shown in Figure 19. The population does not go extinct in this case.

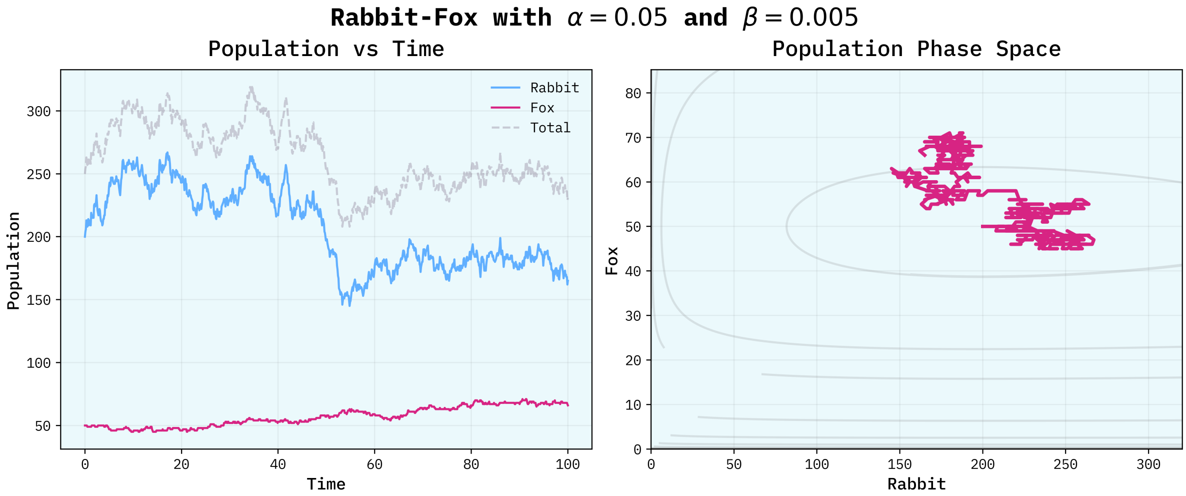

Figure 19: Lotka-Volterra model with $\alpha = 0.05$ and $\beta = 0.005$ where the population does not go extinct.

Next we have a pair of parameters where the population does go extinct. The results are shown in Figures 20 and 21. The population goes extinct in both cases. We can see that a larger rate multiplicator for the foxes leads to a faster extinction.

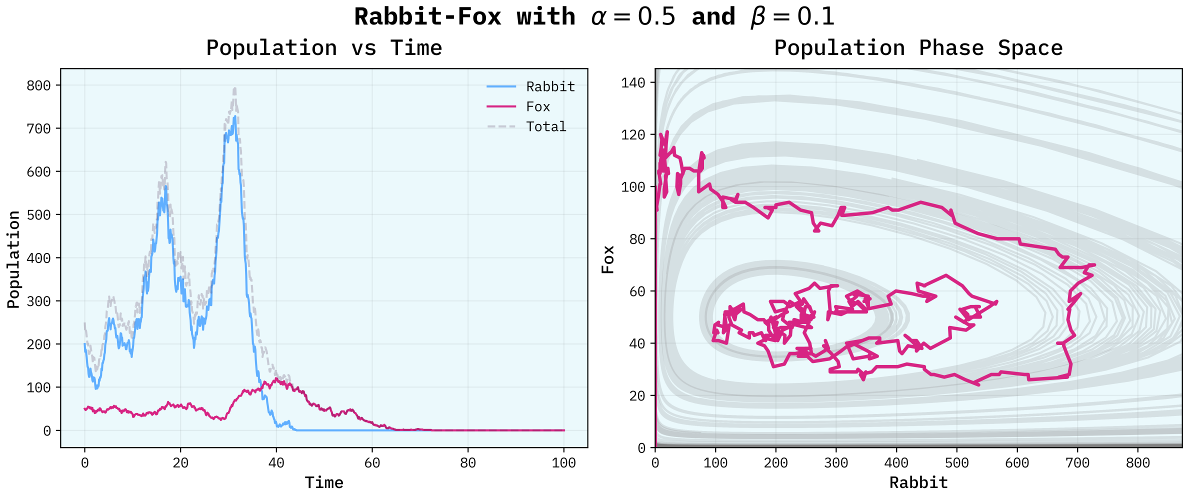

Figure 20: Lotka-Volterra model with $\alpha = 0.5$ and $\beta = 0.1$ where the population goes extinct.

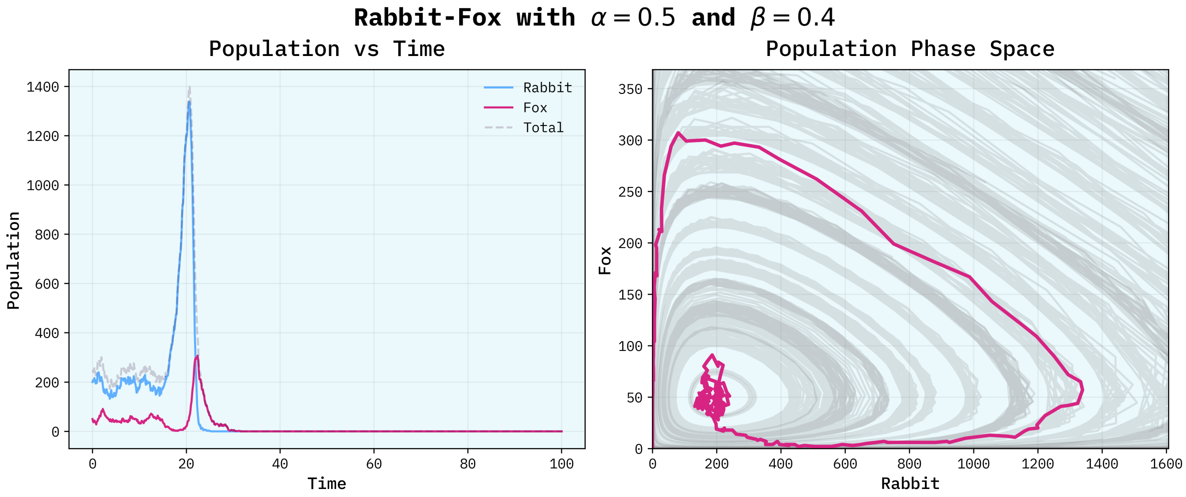

Figure 21: Lotka-Volterra model with $\alpha = 0.5$ and $\beta = 0.4$ where the population goes extinct.

I wanted to see how the death times are distributed for a fixed set of parameters. The results are shown in Figure 22 where I ran the model 1000 times with $\alpha = 2$ and $\beta = 0.5$. I tried fitting a few different statistical distributions to the data, however due to the *ghetto* method of fitting I do not have any idea what the estimated parameter errors are. The curves seem to fit the data however. I think it's pretty obvious that the Maxwell-Boltzmann distribution fits the data the best. Which is interesting but I can see why it would be so. One can maybe think of the population *thermalizing*.

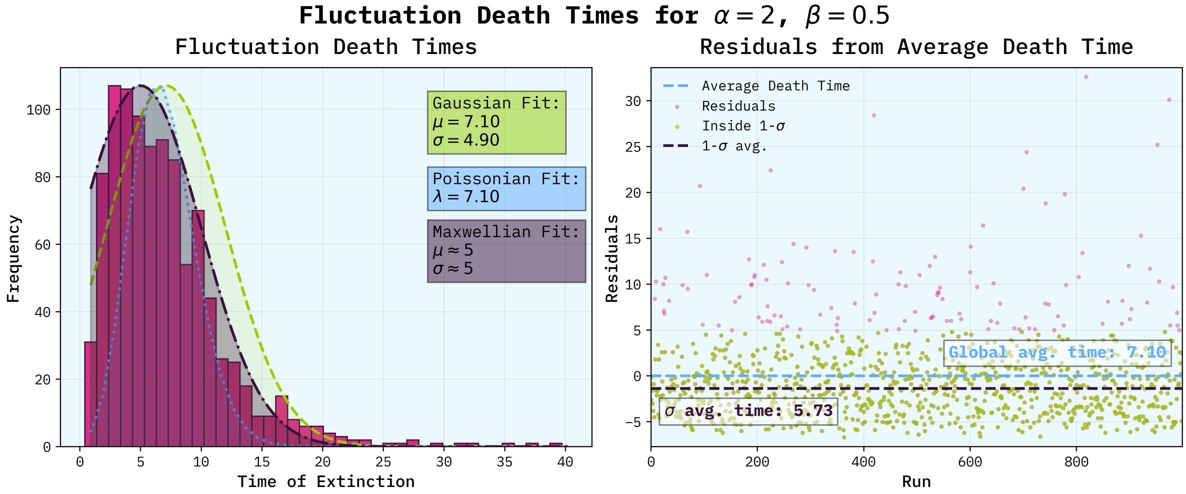

Figure 22: Distribution of death times for Lotka-Volterra with $\alpha = 2$ and $\beta = 0.5$

I wanted to get a wider view of the death times for different parameters, but I unfortunately set the parameter ranges to be logarithmic. Since this data took a while to acquire I'm afraid it would be too costly to redo the scan. The results are shown in Figure 23. I'd like to point out that the left-hand plot contains relative population data since mostly in the case of rabbits their population tends to explode exponentially if the foxes don't deal with them appropriately. In the end I think my main point is still well illustrated by the heatmap plot on the right side. We can see a very large lack of data (in white) where the population did not go extinct. From this we can see that the population is more likely to go extinct for larger values of α and β. They both also seem to have a point where the population is likely to go extinct. The death times extend logarithmically from these points further. Why there are two outcroppings of missing data in these regions I cannot explain. I suspect it could also just be a computational issue with gathering the data in the first place.

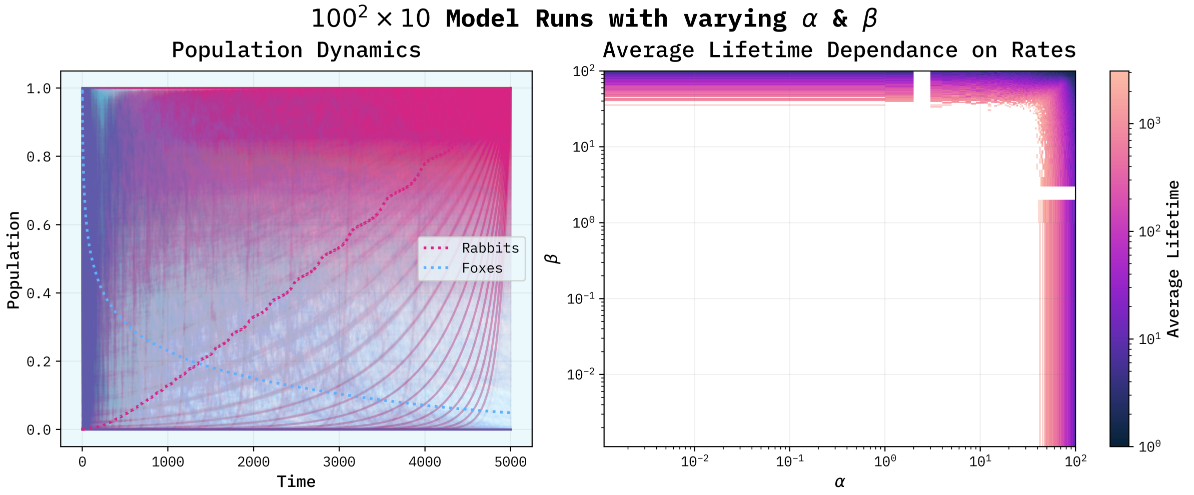

Figure 23: Runs and death times for a (unfortunately) logarithmic parameter scan.

In the subject Theory of Dynamical Systems we learned about how one can identify fixed points in phase space by looking at what are known as the Lyapunov exponents. The Lyapunov exponents are a set of numbers that describe how a point in phase space moves in time. The largest Lyapunov exponent can be thought of as the rate at which the distance between two close-by orbits in phase space grows. This means that for a fixed point the largest Lyapunov exponent is expected to be zero. A negative Lyapunov exponent would mean that the orbits are converging towards the fixed point, while a positive Lyapunov exponent would mean that the orbits are diverging, essentially into a chaotic regime.

Taking a look at the Lyapunov exponents for the Lotka-Volterra model at fixed parameters $\alpha = 2$ and $\beta = 0.5$ in Figure 24 we can see that the largest Lyapunov exponent is positive. This means that the orbits are diverging, which seems to make sense since we know that the population goes extinct due to fluctuations. In the beginning phase of the simulation the Lyapunov exponent is negative which would be expected from a fixed point or something that is converging towards it. This would also likely be the case for the gray deterministic curves we plotted earlier. However as the population starts to fluctuate the Lyapunov exponent becomes positive and the system dies. On the right-hand side I plotted the distribution of the Kolmogorov entropy which is a measure of the rate of information production in a system. The more chaotic the system the higher the Kolmogorov entropy since we need more information to describe the system.

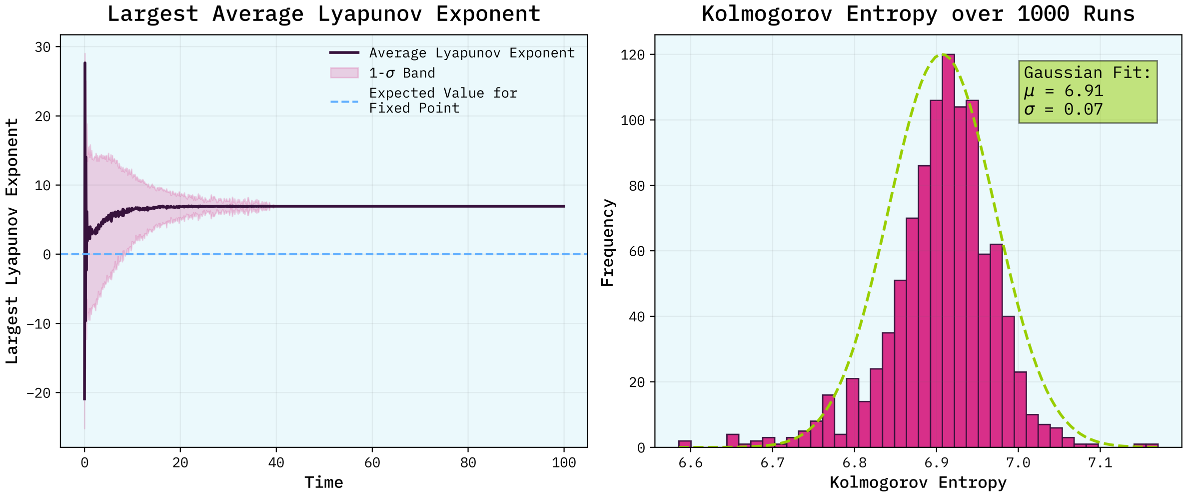

Figure 24: Lyapunov exponents and Kolmogorov entropy for the Lotka-Volterra model.

I also calculated the previous quantities for a wider range of parameters. The results are shown in Figure 25 and zoomed in in Figure 26 due to my unfortunate selection of parameter ranges. The results seem to be consistent with the previous findings. I do find the ridge to be interesting however. Maybe that is related to the death times we saw in heatmap form earlier.

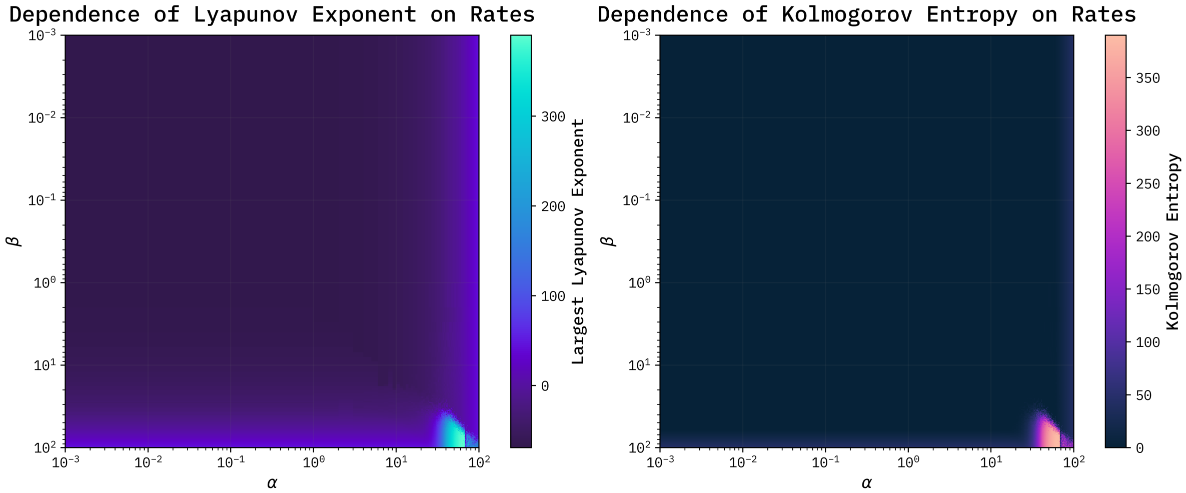

Figure 25: Lyapunov exponents and Kolmogorov entropy for the Lotka-Volterra model.

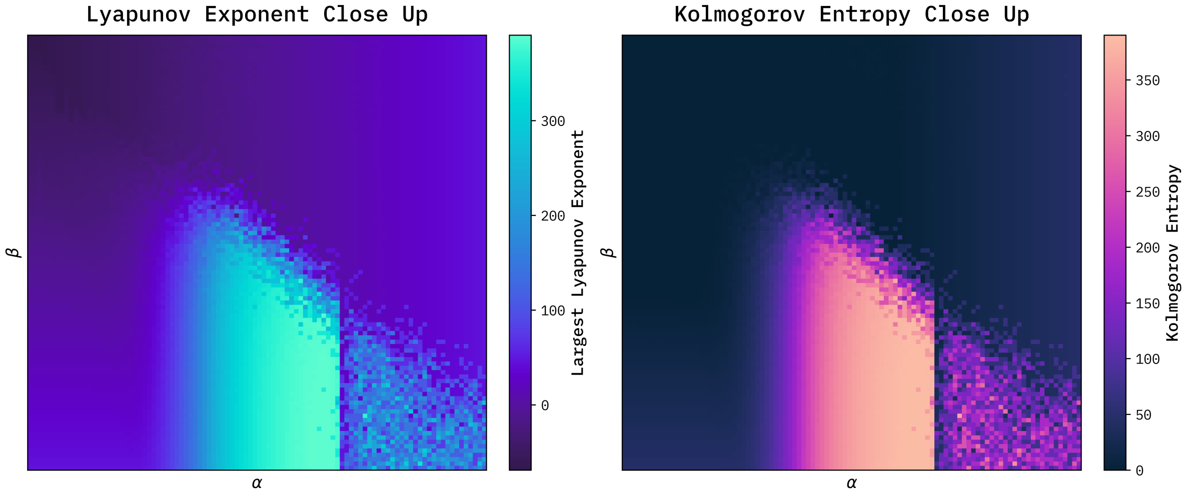

Figure 26: Previous heatmaps zoomed in.

4.4 Disease Eradication Time in an SIR model

I was drastically short on time here so I apologize for the lack of diverse plots. Figures 27 and 28 show two example results for different parameter combinations. It is interesting how one can produce a steady state by adjusting the resusceptibility rate in such a way that the disease is never eradicated (meaning there is no susceptible population, only infected and recovered). The results are consistent with the deterministic model, just noisier which is to be expected. The right-hand side of the figures show the deterministic solution. I even managed to find some weird oscillatory behavior at some parameter combinations as seen in Figure 29.

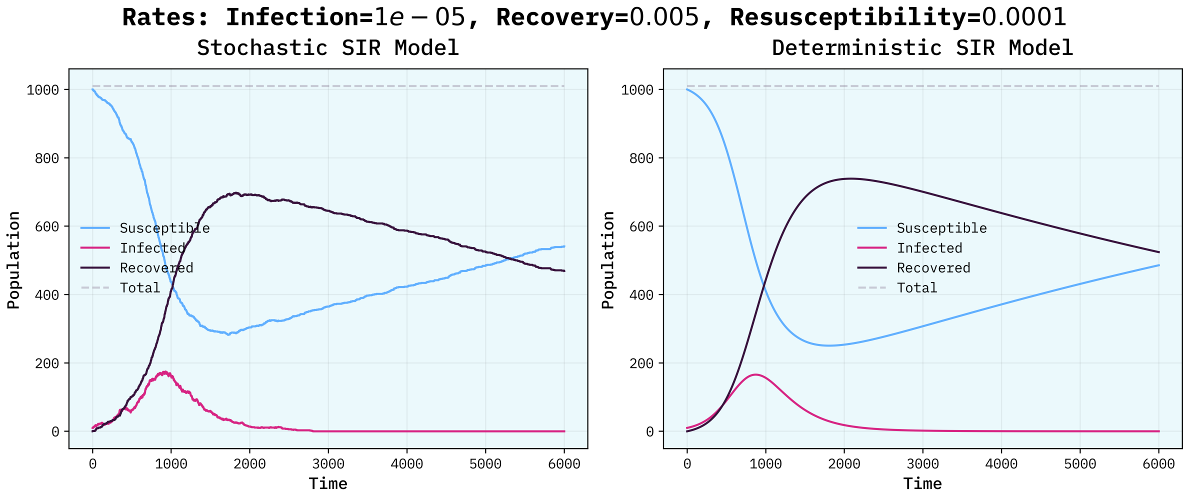

Figure 27: Example runs of the SIR model.

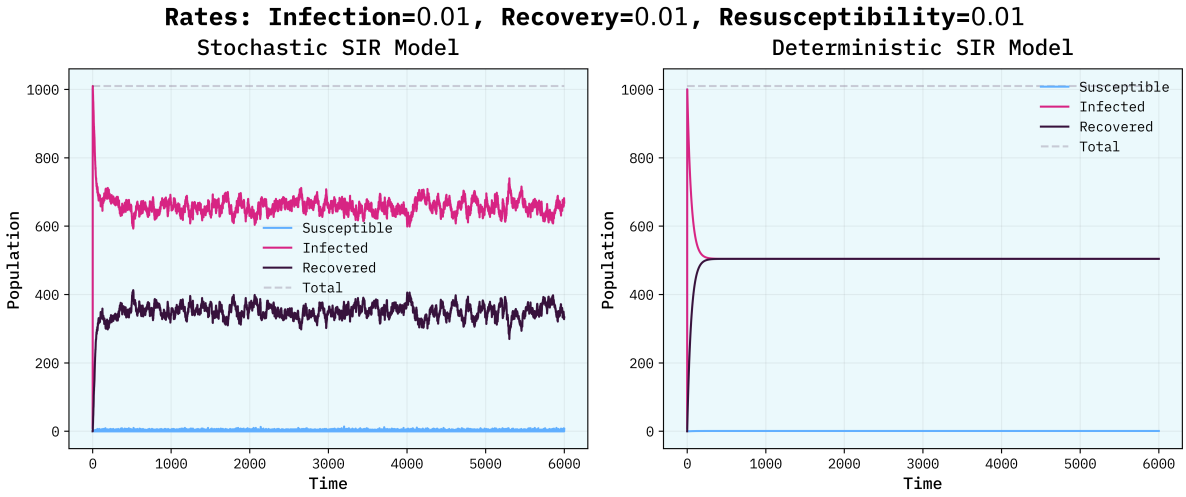

Figure 28: Example stead state of the SIR model.

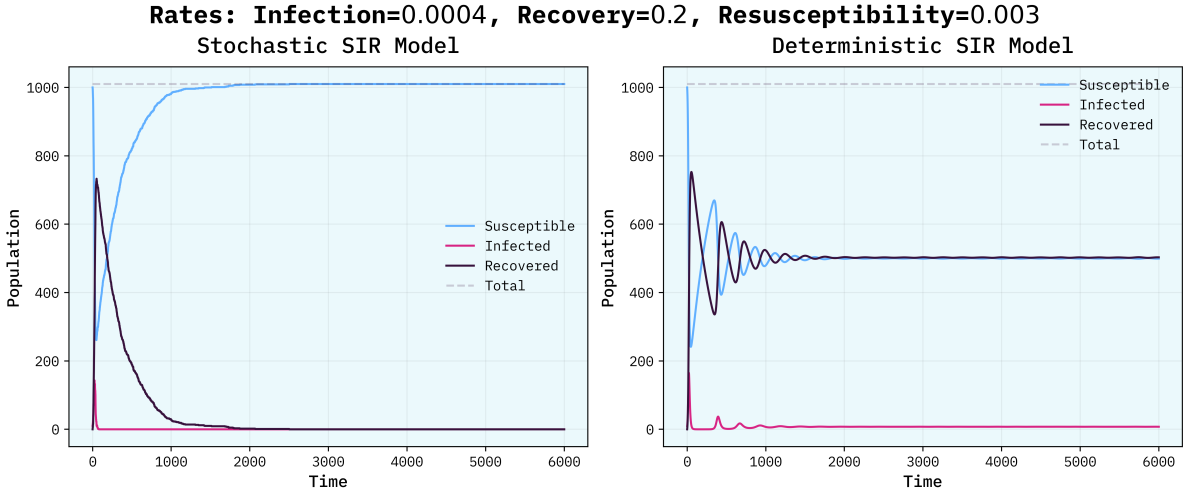

Figure 29: Example oscillatory deterministic solution of the SIR model.

I've only plotted more or less what the instructions demanded. The results are shown in Figures 30 and 31. The initial population size was $N_S = 100$, $N_I = 1$ and $N_R = 0$. The first figure shows how the runs of the stochastic model SIR model behave in time. The data is somewhat truncated since `matplotlib` was having trouble plotting all the data. On the right-hand side we can see a heatmap which aims to show for which combination of parameters the disease was eradicated. This plot was acquired by solving the system of differential equations and looking at the result. It also helped to eliminate parameter combinations that would on average be unsuccessful. The pink pixels represent the parameter combinations where the epidemic did not die out. The dark blue pixels represent the parameter combinations where the epidemic died out. I can't explain why the heatmap looks the way it does, like a barcode. Could be some inherent issue with the way I generated random numbers.

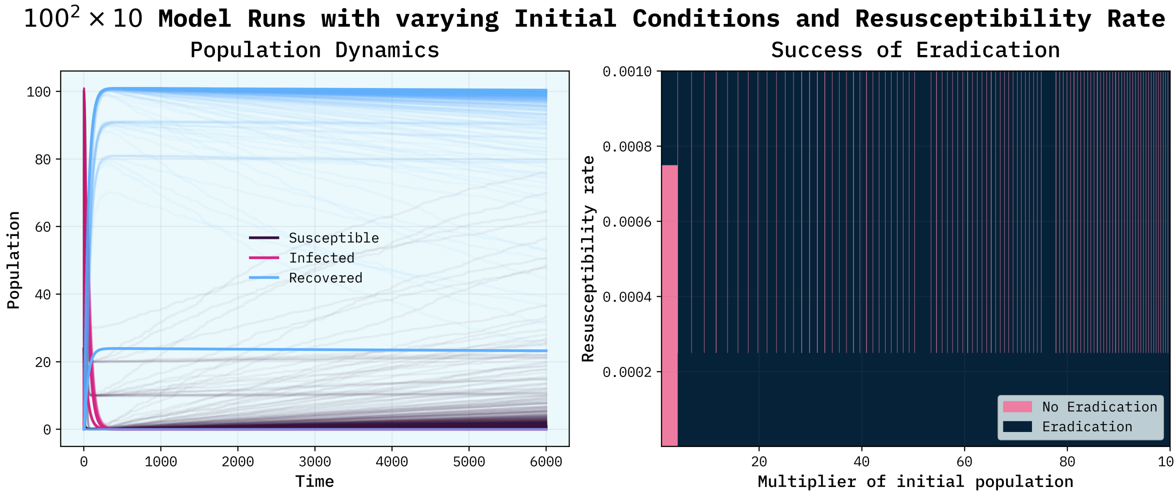

Figure 30: Average Runs and Extinction

The second figure shows the time to eradication for different resusceptibility rates and integer multiples of the initial population size. Essentially the same data is shown just in two different ways. On the left we have a plot with a logarithmic scale for the scaling of the initial population size. Since the resultant curve is a straight line we can conclude that the time to eradication has some exponential dependance. Data is more clearly shown for each pair of parameters in the form of a heatmap on the right-hand side. Notice that dark blue correspond to the parameter combinations where the disease wasn't eradicated. Again I must warn that the left-hand plot contains population data that is scaled according to the initial population size. This was done to better visualize the relative differences and have a friendlier scale for the plot.

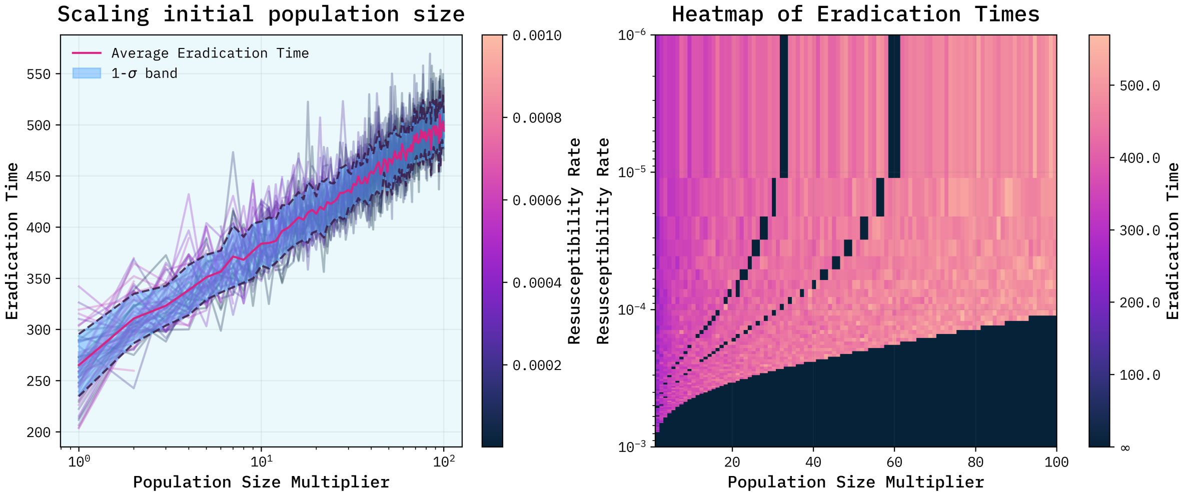

Figure 31: Times to Eradication

5 Conclusion and Comments

I've spent entirely too long on this task and despite that I still feel I've barely done the essentials. I feel as if there is so much more to explore and many more ideas I could realize if I had more time. It is particularly frustrating that I had some personal issues in between which put me *out-of-service* for a few days. I generated a astonishing amount of data for this task and I've barely made use of it. A lot of the plots I included in the report contain blanket averages over the parameter sweeps, which is not necessarily what I wanted to do. It would be interesting to explore the spread of data for each parameter combination and such. Likewise in the case of the Lyapunov exponents and the Kolmogorov entropy. Both of those datasets contain a lot of information that could be statistically significant but due to the lack of time, and to be honest the lack of motivation, after being stuck on this task for so long, I did not explore that data in further detail. Overall I think that the work I **have** included is decent and meaningful. I hope I did the task justice and that I did not disappoint.

Figure 32: Reject modernity, embrace tradition; a funny to finish off, so that the page is less empty.

References

[1] Wolfgang Dahmen and Charles A. Micchelli. Using the refinement equation for evaluating integrals of wavelets. *SIAM Journal on Numerical Analysis*, 30(2):507–537, 1993.

[2] Simon Širca and Martin Horvat. *Verjetnost v fiziki*. Matematika-fizika : zbirka univerzitetnih učbenikov in monografij. DMFA - založništvo, 2016.