Project 1 –Population Growth

Deterministic Birth and Death model with constant birth and death rates

Consider a population of N cells. Individual cells divide at rate r and die at a rate v. The rate of production of new cells is rN and the rate of death of cells is vN. The total rate of change of the population is

$$\frac{dN}{dt} = (r - v)N$$

This is a very simple differential equation for which the exact solution is known.

$$N(t) = N_{init} e^{(r-v)t}$$

The initial population at time t = 0 is $N(0) = N_{init}$. If $r > v$, the population grows exponentially; if $r = v$, the population remains constant, and if $r < v$, the population decays exponentially.

Here is an example, beginning with N_{init} = 100, setting the death rate v = 1, and using three different division rates r = 1.5, 1 and 0.2.

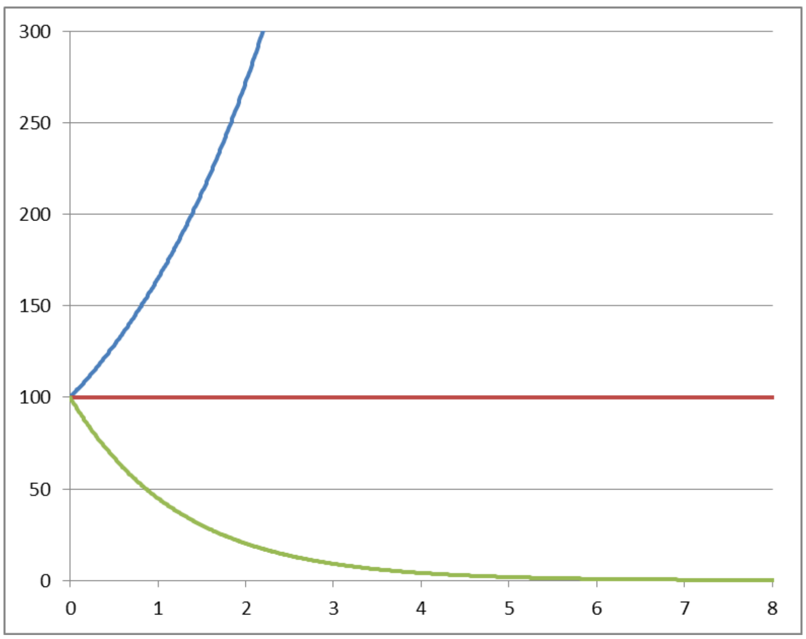
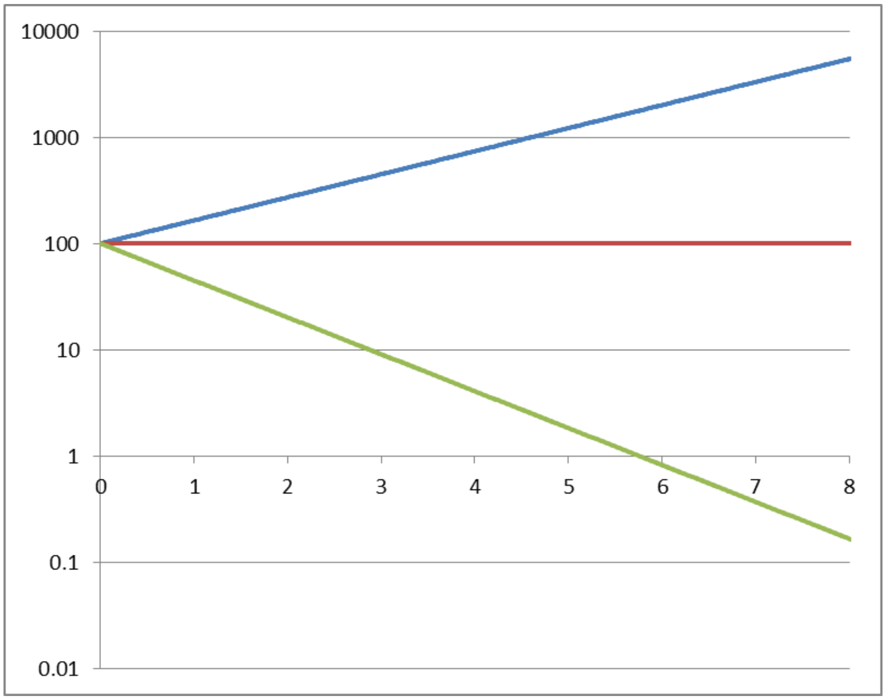


Fig 1

A note on units: If we say the death rate is "1", we mean 1 per unit time. So if our time unit is the hour, then we mean one per hour, or $v = 1\ \mathrm{hr}^{-1}$. The mean lifetime of a cell is $1/v$, which is 1 hr in this case. Similarly division rates are per hour, so if $r = 0.2\ \mathrm{hr}^{-1}$ then the mean time until cell division is $1/r$ = 5 hr. This is longer than the mean time till death. Hence the population decreases if $r < v$.

When we are interested in the qualitative behaviour of models, we are usually interested in relative rates. So we can set one parameter = 1 (like v = 1) in this case, and measure other parameters relative to that. When we are fitting real data, we need real units and real numerical values. In either case, we should remember that rates have units of time^{-1}.

When curves are increasing or decreasing exponentially, it is useful to plot them on a log scale, so that they become straight lines. Here is the same data on a log scale.

Fig 2 

Stochastic Birth and Death model with constant birth and death rates

The above model is <u>deterministic</u>. That means the behaviour depends on the parameters, and on the initial conditions. If you run a deterministic program again with the same parameters and the same starting point, it will give the same answer.

In reality, there is an integer number of individuals in a population. There cannot be fractional numbers of organisms. If we are dealing with very large populations, the deterministic model is OK. For example, if the units of *N* are 'millions of bacteria per cm^3', then if the population falls to 0.1 at the end of the run, then there are 10^5 bacteria per cm^3. However, if *N* represents individual animals, the integer numbers are important. There cannot be 0.1 dodos left on an island. There is either 1 or 0.

When small populations are important, we need <u>stochastic</u> models. This means that individual birth and death events are random events that happen with specified probabilities. In stochastic models, the sequence of random events will be different each time you run the program. So the outcome will be slightly different each time, even when the parameters and the starting point are the same.

We will use the program **exp-growth.nlogo**

The program works in time steps called ticks. Each tick represents a short time δt. The time after n_{ticks} ticks is $t = n_{ticks}\delta t$.

The number of individuals at any time t is an integer $N(t)$, and we begin with $N(0) = N_{init}$. In each time step, each individual has a probability $r\delta t$ of reproducing and a probability $v\delta t$ of dying. In the version given, v is set to 1 in the code, and r is variable on the slider.

The important part of the code is the 'birth-death' routine. The 'ask turtles' loop means look at each individual once in a random order. This loop uses the function 'random-float 1.0'. This means generate a random real number in the range 0 to 1. The probability of birth is $r\delta t$, so if the random number comes out less than $r\delta t$ we hatch a new turtle. If not there is no birth. The probability of death is just δt (because v = 1). So if the random number comes out less than δt the turtle dies. Note that a different random number is supplied by the program every time it gets to random-float. So the outcome for each turtle (either birth or death or neither) will be different.

The rest of the program is to do with setup and plotting functions. Make sure you understand how all parts of the program are working. The exact deterministic solution $N = N_{init}e^{(r-1)t}$ is also calculated in the program and is called nexact.

Run the program several times. Notice that the simulation is slightly different each time. Notice that the N from the stochastic simulation is fairly close to the exact solution, but there is an observable difference between them. Here is a screen shot of one run where the stochastic model (green) comes out slightly higher than the exact result (black).

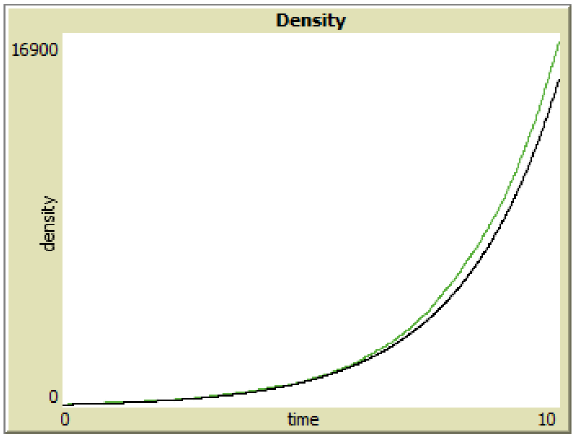


Fig 3

The graphs plotted by netlogo are useful as a quick guide, but for your assignment reports it is better to Export the data from the graph and save it as a .csv file that can be read by Excel. Here is an example of data from the program plotted with Excel on a log scale.

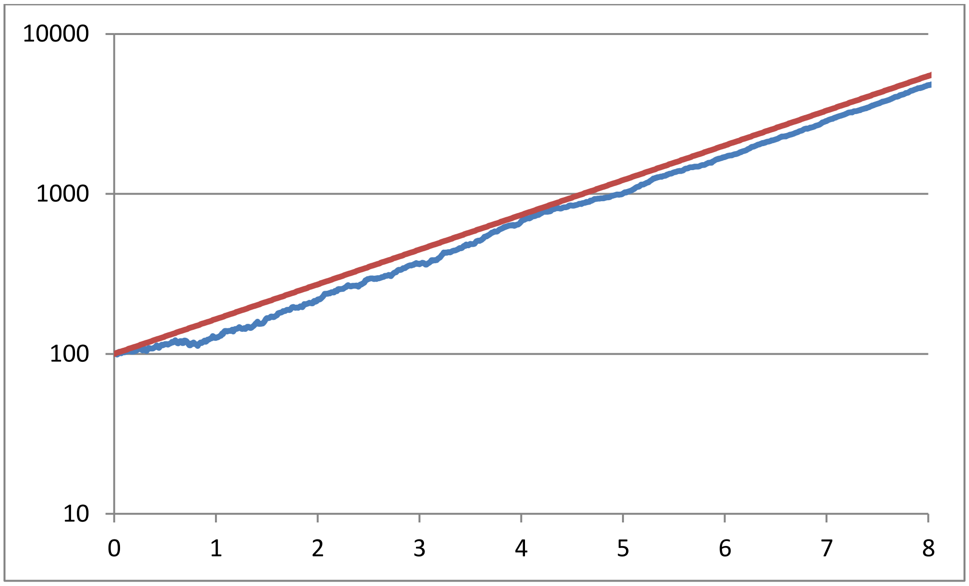


Fig 4 – This graph is missing axes labels and a caption to tell us what it is!

Q1 – Export the data for three different runs of the program with r = 1.5. Plot all three on the same graph on a log scale and include the straight line plot from the exact result.
Now do three runs with r = 1, and three runs with r = 0.2, and plot a graph for each case. Comment on these results and explain why the three runs for each value of r are slightly different. You will see that the population dies out altogether when $r < 1$.

Note – Whenever there is a **Q** in this worksheet, it means there is a question where you need to write something in the assignment report. Please take time to make the graphs in the report clear and legible. For example, Fig 4 would need to have labels on the axes (Population and time) and a caption to say that this corresponds to r = 1.5. You can either include a legend to say that the red curve is the exact result and blue curve is the stochastic model, or you can write this information in the caption.

Deterministic Differential Equation for Logistic Growth

$$\frac{dN}{dt} = rN(1 - N / N_{\max}) - vN \tag{1}$$

This equation describes the growth of a population with limited resources, such that the maximum possible number of individuals that can exist is N_{max}. When $N << N_{max}$, the reproduction rate is r, but the reproduction rate goes down when the population goes up because of the factor $1 – N/N_{max}$. In this version, we suppose that there is a death rate v that does not change with the population size.

The equation can be simplified a bit by defining the net growth rate $R = r - v$, and the carrying capacity $K = N_{\max}(r - v)/r$, and writing it as

$$\frac{dN}{dt} = RN(1 - N/K). \tag{2}$$

General solution of logistic equation is

$$N(t) = \frac{Ke^{Rt}}{C + e^{Rt}} \tag{3}$$

When $Rt >> 1$, $N(t)$ tends to K, *i.e.* the carrying capacity is the limiting value of the population at which point growth balances death. The constant C can be any value, because equation 3 is a solution to equation 2 for any value of C. However, if we know the initial population at $t = 0$ is N_{init}, then we must have $N_{init} = \frac{K}{C+1}$.

$$\text{Therefore } C = \frac{K}{N_{init}} - 1. \tag{4}$$

Q2 Check the mathematics!

- Do the algebra to show that equation 2 is the same as equation 1.
- Substitute equation 3 into equation 2 and prove that it satisfies the differential equation for any value of C.

Stochastic Model for Logistic Growth

Use the program **logistic-growth.nlogo.** Look at the code to make sure you understand how it works. The lattice size is 100 x 100, so the number of patches is 10000. We make the rule that no more than one individual can be on a patch. This means that N_{max} = the number of patches. In the setup routine, random patches are selected and a check is made whether the patch is vacant. If it is vacant, a new turtle is added here. If it is already occupied, a different patch is selected. This continues until there are N_{init} turtles, all on different patches. Compare this with the setup routine in **exp-growth.nlogo** and make sure you understand why it is different.

In the birth-death routine, the probability of birth is $r\delta t$ as before. If a birth occurs, the program picks a random patch to put the new individual on. If this patch is vacant, the new individual is placed here. If the patch is already occupied, the new individual has no place to go. This individual is not added. The probability that the site is vacant is $1 - N/N_{max}$; therefore the probability that an individual gives birth to a surviving offspring is $r(1\text{-}N/N_{max})\delta t$. The mean number of new individuals in one time step is $rN(1\text{-}N/N_{max})\delta t$. This corresponds to the growth term in the differential equation 1.

There is an important detail here: when the new individual is created, it tries to grow in a random patch, which can be anywhere on the lattice and not necessarily next to the parent. This gives a population with no spatial structure. The green turtles are just randomly positioned on the lattice with no correlation. This feature of the model will be changed later in this exercise.

Q3 Run the program for several different values of r. Plot the exact result and the stochastic simulation result for three different values of r, all on the same graph. You don't need a log plot for this, because the maximum population is bounded. You should see that the simulation is very close to the deterministic result in all cases. The stochastic fluctuations are not so big in this case when the lattice is large and the population is bounded.

Spatial growth model

Save the **logistic-growth.nlogo** file as a new file called **spatial-growth.nlogo** and modify the new file.

The spatial growth model in this section differs from the logistic growth model in only one respect. When an individual is created, it must be placed in a patch that is neighboring the parent. This might model the spreading growth of plants or bacterial colonies, whereas the original model might apply to a species with rapid dispersal. For the new model, when an individual is born, select one of the eight neighbors of the parent site. If this site is vacant, the new individual is added; otherwise, the new individual cannot grow, as before.

Change the section of the code that selects the site for the new individual. You will need to read the netlogo manual here. Things that might be useful:

- one-of neighbors – this means choose one of the eight neighbors at random
- you might also need 'sprout' instead of 'hatch'. A patch can sprout a new turtle, whereas a turtle can hatch a new turtle. These are different in the syntax of netlogo.

Keep the setup and the exact solution part the same as in the logistic-growth case.

Q4 Run the program with several values of *r*, as for Q2, and plot the graphs of the results of the spatial model together with the deterministic solutions of the logistic growth model. You should see that the stochastic model is no longer the same as the exact solution (whereas it was the same in Q2). This means that the solution is not the correct solution for the new model. There is quite a lot of difference between the spatial growth model and the logistic growth model. The spatial model grows more slowly and reaches a population size that is smaller than the carrying capacity *K* of the logistic model. Why is this? Comment on the distribution of the green turtles in the spatial model, and how it differs from the logistic model.

Q5. The mean population size in the logistic model is $K = N_{\max}(r - v)/r$ at long times, when the population has reached a stationary state. Plot a graph of this as a function of *r*, with N_{max} = 10000 and v = 1. What is the minimum reproduction rate *r* at which the population can survive?

Now run the spatial model and measure the mean population size in the stationary state as a function of *r*. Plot this on the same graph. This should be lower than *K*. What is the minimum value of *r* at which the population can survive? Explain why it is different from the logistic model.

Using Patches Instead of Turtles

The programs we used so far have focused on the turtles. We have thought in terms of turtles reproducing, and we have asked questions such as "is there already a turtle on this patch?"

However, the spatial model above is very simple because each patch can have either one or zero turtles on it. In fact we can write this model in a different way that uses only patches and no turtles. Save the file **spatial-growth.nlogo** as a new file called **patch-growth.nlogo** and modify the new file. We will use the color of the patch (the variable pcolor) to indicate whether the patch is occupied. If we set pcolor = green, it means that it is occupied, whereas if pcolor = white, it means the patch is empty.

Change the setup routine. Do not create any turtles. Start with all the patches white. Instead of saying "If there is no turtle here, create a turtle", say "if this patch is white, make it green". Add green patches randomly until there are N_{init} patches with pcolor = green. This is the equivalent of having N_{init} patches with one turtle each.

Change the birth and death routine. You cannot have loops with ask turtles any more. You have to change these to ask patches. Within the ask patches loop, we say: "If the patch is green it reproduces with probability $r\delta t$. If it reproduces, ask one of the neighbors to set its pcolor = green". This is simpler than before. It might be that the neighbor was already green, in which case nothing happens. But if the neighbor was white, then we will change it to green, in which case we have increased the population by one.

For the death part, we replace the line that asks turtles to die. We simply have to say "If the patch is green, set pcolor = white with probability δt."
Finally, in the plotting routine, we change the line "set npop (count turtles)"
to "set npop (count patches with [pcolor = green])".

Q6 – The new patch-growth model should be exactly the same as the spatial-growth model. Save a screen shot of the patch model to show that you have this working. Save the population as a function of time for a couple of different parameter values with the patch model and also with the spatial-growth model. Plot them on the same graph in order to show that they are the same.