

Department of Bioengineering

BE3-HMIB – Modelling in Biology (MiB), Dr Guy-Bart Stan & Dr Tom Ouldridge

Training coursework 8

Coupled population processes

In this exercise, we will consider the coupled population dynamics of a simple “predator-prey” model. You will learn how to code-up your own (2d) Gillespie simulation. The model is a stochastic version of the famous Lotka-Volterra equations, which describe the interaction between a population of predators and a population of prey. In the deterministic case, the system shows limit cycle oscillations in populations: https://en.wikipedia.org/wiki/Lotka%E2%80%93Volterra_equations.

We have two populations, rabbits (R) and foxes (F). Rabbits are born at a rate aR (a per rabbit), and consumed by foxes at a rate bRF . Foxes are born at a rate depending on the number of rabbits available, gRF , and die at a rate cF (c per fox).

1. Construct an m-file function that takes the current population, along with the birth/death parameters, as input, and returns a new state and time for transition sampled appropriately. You will need to consider the relative rates of four transitions at each step, and pick one using a random number. You will then need to draw a time from an exponential distribution. It might be useful to build an “exception handler” that deals with the possibility that $R, F = 0$ is the current (absorbing) state.
2. Use this function to explore the possible behaviour of the ecosystem by taking sample trajectories. Taking $a = 1$, $b = 0.01$, $c = 0.4$ and $g = 0.002$, and initial conditions of $R = 100$ and $F = 10$, simulate for 100000 transitions or until both populations become extinct, whichever occurs first. Plot the populations as a function of time (note: not number of transitions). Run the code several times; what types of behaviour do you observe? What has happened to the oscillations?
3. By running 100 simulations, estimate the average time (note: not number of transitions) until the foxes become extinct.

In this coursework you may need to use the following Matlab commands: `rnd` and `exprnd`. You will also find it helpful to define a separate “.m” file defining a function that actually calculates the next step. The first line of the file (in my case named `gillespie_ex.m`) should look something like: `function [rnew,fnew,deltat] = gillespie_ex(rold,fold,a,b,c,g)`. The quantities `rnew`, `fnew` and `deltat` are then calculated in the file, and automatically returned. You can then call this function from your main code each time you need to take a step. You can check the Matlab help by using `help COMMAND`.