### Exercises 2, Dynamic Systems

Modelling Biological Systems, BIOS13  
Mikael Pontarp and YunTing Jang

#### 1. Lotka-Volterras predator-prey equations

Lotka-Volterras predator-prey equations look like this:

,  
where *n*(*t*) is the prey population size, *p*(*t*) is the predator population size, *r* is the intrinsic growth rate of the prey, *a* is the predator attack rate and *μ* is the predator mortality. (We have rescaled the predator population such that one eaten prey generates exactly one new predator. Otherwise, there should be a conversion factor in the second equation)

**a**) Write an R script that plots the isoclines of the system. Make it a function file with input parameters according to:

LV\_isoclines <- function(r,a,mu)

…

Remember that a straight line can be plotted by just specifying the coordinates of two points – the first and the last. So the command to plot a straight line in R can look something like

plot(c(x1,x2), c(y1,y2), type='l')

to plot a line between the points (x1,y1) and (x2,y2) (You may want to consult “Using the plot comand.pdf” in Canvas). An alternative is to use the abline function.

Try your function! For example, test *r* = 1, *a* = 1,  = 2:

> LV\_isoclines(1,1,2)

**b)** Now we will run some simulations. Write an R-script like in the previous exercise, using the ode function. The difference now is that we have two state variables instead of one, so the system function has to return a vector of two derivatives, put in a list as before. Something like this:

LV\_sys <- function(t, np, Parameters) {

# extract vector content:

n <- np[1]

p <- np[2]

# calculate the two growth rates:

dndt <- (…you fill it in…)

dpdt <- (…)

return(list(c(dndt, dpdt ))) # the result as a vector in a list

}

The input parameters are first of all the time t, which we don’t use, and np, the vector of state variables. The state variables are prey and predator density, in that order. Next follows a list with parameter values, which I called Parameters here (you can of course use another name).

The output should be a list, containing the time derivatives of the state variables in a vector format.

Next, write a script file that uses the ode function to simulate the system, just like in the previous exercise. Something like this:

out <- ode( y = np0, func = LV\_sys, times = timevec, parms = P)

The output out is a matrix with three columns. The first column contains the time-points you specified in timevec. The second contains the prey densities for each time-point, and the third column contains the predator densities. The order of the state variables (n,p) has to be the same in the initial condition, np0 (a vector).

To plot the two populations in the same time-plot, you can do something like

# plot the prey densities in blue:

plot( out[,1], out[,2], type=’l’, col=’blue’ )

# add the predator densities in red:

lines( out[,1], out[,2], col=’red’ )

Also add code to draw a phase-plane plot (p vs. n).

Make sure your script does the whole sequence: define the system function, set up the necessary parameters, run the simulation, and finally plot the result (time-plots and phase-plane. Finally, use the LV\_isoclines function you wrote before to add isoclines to the phase-plane plot.

Test a few different parameter values and initial conditions.

**c**) Calculate the Jacobian matrix of the Lotka-Volterra predator-prey equations. Write a function in R that takes the three parameters (*r*, *a*, *μ*) as input and returns the Jacobian matrix as output.

d) Use your function in c) to calculate the Jacobian for a set of parameter values. Also calculate the cycle period associated with the complex eigenvalues. Is it a good approximation of the length of the actual cycles? (check your plots, rerun your simulation script with the new parameter values if necessary)

*If you have the time:*

#### 2. Stabilising Lotka-Volterra

The Lotka-Volterra equations have been criticised for a number of unrealistic assumptions. For instance, the prey will grow to infinite population sizes without the predator. To correct for this, we can introduce density dependence of prey growth, following the logistic equation:



**a)** Calculate the new equilibrium!

**b)** Can you find a condition for predator existence, i.e. a necessary condition for a positive predator equilibrium?

**c**) Write an R-script, like the one you did before, which plots the isoclines of the new system.

**d**) Write an R-script to simulate this system and do some plotting, like above. Try some different parameter values. Does the equilibrium seem to be stable?

**e)** Calculate the new Jacobian matrix and its eigenvalues (in R). Conclusions?