

## Lab 10 : Parasitoid-Host Dynamics (Spatial and Nonspatial)

### Objectives:

- \*To see how the number of patches in a spatial parasitoid-host model impact persistence time.
- \*To see how dispersal rate of hosts relative to parasitoids impacts persistence time.
- \*To compare the results of the spatial and non-spatial parasitoid-host models.

### Part 1: Nicholson-Bailey Predator Prey Model

Nicholson-Bailey Model is (cf lecture-notes):

$$H_{t+1} = R H_t \exp(-a P_t)$$
$$P_{t+1} = R H_t [1 - \exp(-a P_t)]$$

where  $R$  is the per capita growth rate of the host and  $a$  is the searching efficiency of the parasitoid.

The R-code for the Nicholson Bailey Model is:

```
NB = function(R, a, T = 100, N0 = 10, P0 = 1){
  #R is the host growth rate,
  #a is parasitoid efficiency,
  #T is length of simulation (number of timesteps)
  #N0 and P0 are initial numbers

  #Note that default parameters are not provided for R and a

  N=rep(NA, T) #host series
  P=rep(NA, T) #parasitoid series

  N[1] = N0 #Initiating the host series
  P[1] = P0 #Initiating the host series

  for(t in 2:T){
    N[t] = R * N[t-1] * exp(- a * P[t-1])
    P[t] = R * N[t-1] * (1-exp(- a * P[t-1]))
  } #end of loop

  #the two vectors of results are stored in a 'list'
  res= list(N = N, P = P)

  #the list is passed out of this function
  return(res)
} #end of function
```

If we assign the output of this function under a name (e.g. `sim=NB()` ), we can extract the host and parasitoid time series as `sim$N` and `sim$P` respectively.

## Assignment 1:

(a) Assume a host growth rate of 1.1 and a parasitoid searching efficiency of 0.1. Use the `NB()` function to simulate the Nicholson Bailey model. Make a plot of host abundance against time. Superimpose the time series of the parasitoid.

(b) Plot the two time series against each other in a state space plot.

(c) Assume a sequence of searching efficiencies between 0 and 1. How does the time to extinction of the parasitoid depend on the searching efficiency? Make a plot of time to extinction as a function of searching efficiency.

## Part 2: Spatial Parasitoid-Host Model

Below is the code for the spatial Parasitoid-Host model we discussed in the lecture. Review the R-code and answer the questions at the end of the code.

```
#Dh is proportion of host that disperses to neighboring patches
#Dp is proportion of parasitoid that disperses
#Changing these can change the spatial dynamics to
```

```
#Spatial chaos or spiral waves (cf lecture notes)
Dh = 0.1
Dp = 0.1
```

```
#xlen is width of the lattice (E-W)
#ylen is height of the lattice (N-S)
xlen = 30
ylen = 30
```

```
#The below defines the function to update the local abundances
#of hosts and parasitoids according to the Nicholson-Bailey model
#previous densities of host (h) and parasitoids (p)
#need to be supplied as arguments to the function,
#in addition to other parameters
```

```
hp.dyn = function(h, p, R, a){
  #R is the host growth rate,
  #a is parasitoid efficiency,

  #hnew is the post-interaction host density
  hnew = R * h * exp(- a * p)

  #pnew is the post-interaction parasitoid density
  pnew = R * h * (1 - exp(- a * p))

  #the two vectors of results are stored in a 'list'
```

```

    res= list(h = hnew, p = pnew)

    #the list is passed out of this function
    return(res)
} #end of the H-P function

#We generate spatial coordinates -- the call returns a matrix
#with two columns. The first is the x-coordinates, the second is
#the y-coordinates. Try "expand.grid(c(1,2,3), c(1,2,3)) to see
#what the function does.

xy = expand.grid(1:xlen, 1:ylen)

#calculating the distance between all populations.
#We use the vectorized calculations discussed in the previous
#lab: 'outer' is a function that takes a vector x, and a vector y
#and then makes a matrix that has applied some function
#(in this case subtraction) to all possible pairwise combination
#of the x and y values. The below code calculates the matrix of
#squared differences in x-coordinates, then adds the matrix of
#squared difference in y-coordinates and then takes the square-
#root. Thus succeeding in calculating all spatial distances in one
#whack.

dmat=sqrt(outer(xy[,1], xy[,1], "-")^2+outer(xy[,2], xy[,2], "-")^2)

#The redistribution matrix is calculated by checking if the
#distance in the dmat is smaller than 1.5 (thus flagging all pops
#that are closer than the square-root of 2 to one-another -- the
#first neighbors).
#Then assigning a value Dh/ 8 to each of them. That is of the
#proportion that leaves, the dispersers are distributed evenly
#among the eight neighbors.

kh=ifelse(dmat<1.5,Dh/8,0)
kp=ifelse(dmat<1.5,Dp/8,0)

#then we have to insert the proportion that does NOT disperse in
#the diagonal of the redistribution matrices
diag(kh) = 1-Dh
diag(kp) = 1-Dp

#IT is number of iterations
IT <- 600

#Setting up the matrices to store the space-time data
#for each of the species. The number of rows are equal
#to the number of populations, the number of columns are
#equal to the number of iterations.
#The matrices are initiated with missing values (indicated

```

```

#by the NA in the function call.

#hmat contains the time-series of host
hmat = matrix(NA, nrow=xlen*ylen, ncol = IT)

#pmat of parasitoid
pmat = matrix(NA, nrow=xlen*ylen, ncol = IT)

#initial conditions are zeros everywhere (i.e., the first
#column in each matrix gets zeros), except for an arbitrary
#population (in this case 23) which starts with 4 hosts
#and 1 parasitoid

hmat[,1] = 0
pmat[,1] = 0
hmat[23,1] = 4
pmat[23,1] = 1

#The next is to do the spatial simulation.
#We simulate from generation two to IT,
#storing the results on the way

for(i in 2:IT){
  #growth - the list returned from function hp.dyn is
  #stored in tmp, host and parasitoid densities from
  #previous generations are passed as arguments to the function

  tmp = hp.dyn(h = hmat[, (i-1)], p = pmat[, (i-1)], R = 2, a = 1)

  #redistribution - the vector of pre-dispersal hosts 'tmp$h' is
  #redistributed through matrix multiplying the vector by the
  #redistribution matrix 'kh'. the resultant post-dispersal
  #abundances are stored in the correct column of the
  #output matrix

  hmat[,i] = tmp$h%*%kh;

  #ditto for parasitoid
  pmat[,i] = tmp$p%*%kp;

  #a counter showing the progress
  cat(i, " of ", IT, "\n")
} #end of simulation

```

```

#open graphics window:
win.graph()
#remember to tile the windows

#plot the last 100 generations for the parasitoid
for(i in 1:100){
  x=xy[,1]
  y=xy[,2]
  z=pmat[,i+500]
  symbols(x,y, fg=2, circles=z, inches=0.1, bg=2, xlab="",
ylab="")
  Sys.sleep(.1) #this is to slow down the plotting
}

```

## Assignment 2:

(a) *Change the size of the map lattice from (i) 30 by 30 to (ii) 15 by 15 to (iii) 10 by 10 to (iv) 5 by 5 to (v) 3 by 3. How does this impact persistence length? In which case did you see global extinction? How long does it take before global extinction?*

(b) *Change the dispersal rate of the host and parasitoid such that (i) the parasitoid dispersal rate is greater than the host (ii) the host dispersal rate is greater than the parasitoid. How does this change persistence length? How does this affect the conclusions in (a)*

(c) *How do the results of the spatial model compare to the non spatial Nicholson -Bailey model? What do the results of the model suggest about the impact of space on population dynamics?*