Tutorial: Resilience through Adaptation

Author: George van Voorn (Wageningen University & Research)
With: Guus ten Broeke (Wageningen University & Research)

This is part 3 of the teaching module "Sensitivity Analysis for Agent Based Models". For this module we assume you (the student) are working on an Agent Based Model (ABM) built in NetLogo, for instance in the context of studying the resilience of a Socio-Ecological System. We also assume you are familiar with using R for statistical analysis and RNetLogo for the interface between R and NetLogo (developed by Jan Thiele). If this is not the case, we kindly refer you to part 1 of this tutorial. We also assume to understand the fundamentals of performing a Sensitivity Analysis to analyze the output of a model. If this is not the case, we kindly refer you to part 2 of this tutorial.

ABMs can be used as simplified representations of real-world systems including the mechanisms that may contribute to generating resilience. This module deals with means of converting the quantitative effects of shocks in the ABM to metrics for quantifying resilience. Observe that there may be many more ways of doing this, and this is just one example. Here we look at the model used in our publication Ten Broeke, Guus, George Van Voorn, Arend Ligtenberg, Jaap Molenaar. "Resilience through adaptation." *PloS one, 12(2),* e0171833 (2017).

The used model can be found on https://www.comses.net/codebases/5374/releases/1.1.0/

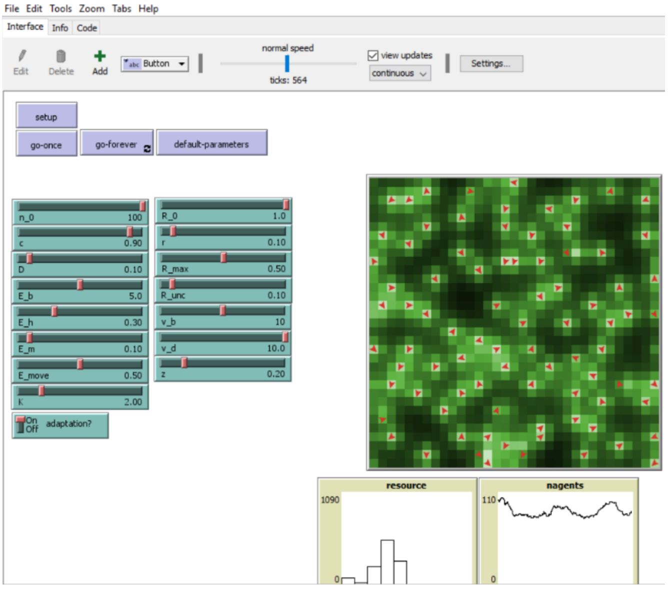
Contents

1. The ABM and general results of the model

The model description is available in the paper mentioned in the introduction. In short, the model explicitly describes agents ('consumers') living in a spatially heterogeneous world consisting of renewable resources that generate energy ('food'). Every iteration (i.e., model time-step) each agent spends a small amount of energy, even when doing nothing (i.e., maintenance costs). Then, an internal energy status check is performed, and the results are used in the stochastic decision-making processes. In addition, the agents monitor their direct surroundings for resource availability and the presence of competing agents. Based on the obtained information, agents may choose to either harvest from the present location, move to a neighbouring site, or wait for a better opportunity. An agent is likely to harvest if its internal energy is low and resource is available from the present location. It is likely to move to neighbouring sites with more resources, and fewer competing agents. Furthermore, each agent has individual trait values that modifies its tendency to move or harvest.

After taking its action, a higher energy status increases the probability an agent will split into a 'parent' and a 'child' (i.e., 'procreation'). A lower energy status increases the probability it will die.

When we open up the model it looks like this:



n_0 100
c 0.90
D 0.10
E_b 5.0
E_h 0.30
E_m 0.10
E_move 0.50
K 2.00
On Off adaptation?
R_0 1.0
r 0.10
R_max 0.50
R_unc 0.10
v_b 10
v_d 10.0
z 0.20

The variables and parameters are given in Table A2 of the ODD in the download from https://www.comses.net/codebases/5374/releases/1.1.0/

The traits w^{α} are initialized in the setup function, while parameter z gives the variation in offspring inheritance. We will return to this inheritance of traits later in the tutorial.

Let's generate a single run of the model with RNetLogo. The file 'step01.r' contains the following. First, the usual lines of code to set the working directory and load the required packages:

```
➢ ls(all = TRUE)
➢ rm(list = ls(all = TRUE))
➢ ls()
➢ ls(all = TRUE)
➢ setwd("C:/0Werk/Onderwijs/Module SA for ABM - SURF-DeSIRE/Data_for_part3")
➢ getwd()
➢ .libPaths("library")
➢ install.packages("RNetLogo")
➢ library(RNetLogo)
```

Then, we set the path for the ABM

```
➢ nl.path <- "C:/Program Files/NetLogo 6.0.4/app"
➢ nl.jarname <- "netlogo-6.0.4.jar"
➢ NLStart(nl.path = nl.path, gui = FALSE, nl.jarname = nl.jarname)
➢ NLLoadModel("C:/.../Data_for_part3/Models/RC_ABM2.nlogo")
```

(We assume you have placed the files into such a WD called Data_for_part3, with a subdirectory for the NetLogo models.)

We create some empty vectors to store the simulation data of each iteration:

```
data.time <- c()
data.resource <- c()
data.agents <- c()
```

We set the number of iterations (in the follow examples the number of iterations will be 1000, but you can set it to, e.g., 10 to first test the running of the model):

```
nticks <- 1000
```

Now, we provide the commands for actually running the model:

```
NLCommand("setup")
for(i in 1:(nticks+1)){
        tick <- NLReport("ticks")
        NLCommand("go")
        data.time[i] <- i
        data.resource[i] <- NLReport("sum [resource] of patches")
        data.agents[i] <- NLReport("count turtles")
}
NLQuit()
```

The vector 'data.time' now contains the iteration numbers, the vector 'data.resource' the sum of all the resources of all patches in the model at every iteration, and the vector 'data.agents' the total number of turtles (agents) at every iteration. We finally put the vectors together in a dataframe to print to a file:

```
df.p <- data.frame(data.time,data.resource)
df <- data.frame(df.p,data.agents)
write.table(df, "singlerunmodel.dat", sep = " ", row.names = FALSE, col.names = TRUE)
```

After running, you should have a file 'singlerunmodel.dat'. We read this back in with the file 'step02.r':

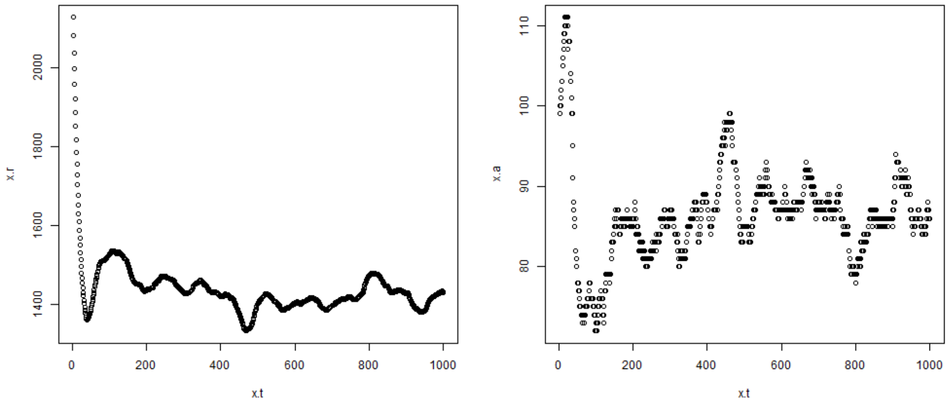
```
x <- read.table("singlerunmodel.dat", header = TRUE, sep = " ")
dim(x)
```

Indeed, we have a 1001 by 3 datafile. It may be convenient to assign shorter names to the different vectors:

```
x.t <- x$data.time
x.r <- x$data.resource
x.a <- x$data.agents
```

Now we plot the run:

```
png(filename="singlerun.png",width=1000,height=480)
par(mfrow=c(1,2))
plot(x.t,x.r)
plot(x.t,x.a)
dev.off()
```



You can play around with different parameter settings of the model (either by directly changing parameters in the ABM or indirectly via RNetLogo – the latter is recommended if you want to keep the original parameter settings) to see what type of dynamics are generated.

The model contains different stochastic elements. For example, the decision-making of agents is stochastic, but the probabilities for each decision are also determined by the agent traits, the current state of the agent, and the observed state of its local surroundings. Let us now look at the distribution of the end points of a set of simulations at the basic settings. The file 'step03.dat' opens up the datafile 'replicates.csv':

- x <- read.table("replicates.csv", header = TRUE, sep = ",")

The input looks something like this:

```
...
V1997 V1998 V1999 V2000
1    84    83    83    83
2    93    93    93    92
3    82    82    82    82
4    85    85    86    86
5    84    84    84    84
6    91    90    90    89
```

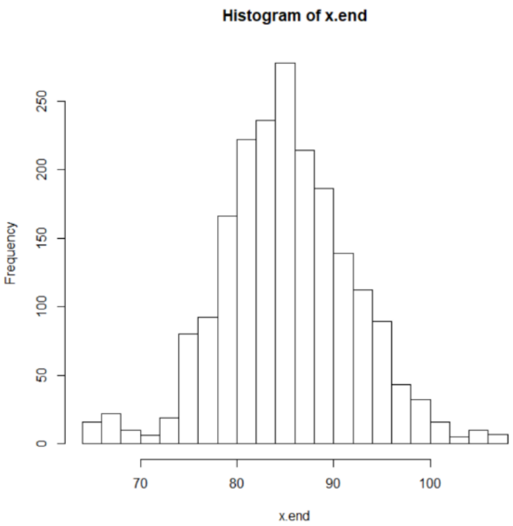
A dimensional check tells you we have a 1000 iterations and 2000 replicate runs (plus the first column, which indicates the iteration numbering):

- dim(x)
  [1] 1000 2001

We want to make a histogram of the end points. We transpose the matrix and select the final point of each of the 2000 runs:

- x.t <- t(x)
- x.end <- x.t[2:2001, 1000]
- hist(x.end, breaks = 20)

The resulting histogram looks like this:



In the first part of the tutorial we discuss how to use a test like the Shapiro-Wilks test to check for normality. Later in this tutorial we will also perform such a test. For now, let's accept this distribution looks rather like a normal distribution, and continue with looking at resilience in our ABM.

2. Resilience mechanism: Trait inheritance

Resilience refers to the ability of a system to cope with shocks, either by buffering them or by having mechanisms for restoration after the shock has had its effect. Resilience is always evaluated in the context "of what, to what?" (Carpenter et al., 2001), and possibly also "for whom?" (Cutter, 2016).*

* Carpenter, S., Walker, B., Anderies, J. M., & Abel, N. (2001). From metaphor to measurement: resilience of what to what?. *Ecosystems*, *4*(8), 765-781.
Cutter, S. L. (2016). Resilience to what? Resilience for whom?. *The Geographical Journal*, *182*(2), 110-113.

An important mechanism for resilience in our ABM is the mechanism through which agents 'procreate'. Agents have several traits (randomly initialized by a distribution that is captured by model parameters) associated with the decision-making processes involving stochasticity. These traits are inherited by the 'offspring'. The model thus has two settings. In the first setting, which we name *random traits*, whenever an agent 'procreates', a copy is made of the agent with new traits randomly drawn from uniform distribution. In other words, the 'offspring' is in no way a copy of the 'parent', but just some random new agent. This setting is the model we used to generate the above results. In the second setting, which we will name *inheritance*, whenever an agent 'procreates', the copy receives trait values that are comparable to those of the 'parent'. There is some randomness in setting these new trait values, but generally speaking the 'offspring' resembles the 'parent'.

Shocks are not implemented explicitly in our model, but we vary the values of important parameters related to the resource dynamics, resulting in resource shortage for the agent population. The general result in the simulations from 'resource shortage' shocks is that in the random traits runs, the population of agents tends to go extinct. In the inheritance runs, the agents that make 'better decisions' tend to perform better, yielding more resources and thus a higher internal energy status, in turn leading to a larger probability of procreating. The 'offspring' in turn inherits the better traits values, increasing their chance for survival and procreation, and so on. Several runs go extinct, but in many runs the population survives. After some 'adaptation' period, the population persists and consists of 'better' adapted individuals that can better cope with the resource conditions. These results are summarized in Fig. 10(a) of the paper by Ten Broeke et al., in which the number of surviving populations is depicted as function of time. The mechanism of population adaptation through a combination of inheritance of traits and natural selection may be a critical aspect of what we call resilience of a socio-ecological or socio-technical system. Implicit in this is also the heterogeneity in the agent population, i.e., there must be agents with different trait values performing differently relative to each other to have local 'winners' and 'losers'.

For the remainder of this tutorial we will look deeper into the quantification of resilience.

3. Quantifying resilience

Although there are many ways to quantify resilience, a dominant metric is the return time, i.e., the time it takes for the system to return to the pre-shock state. This metric is heavily associated with the concept of eigenvalues from linear algebra. Assume a matrix A of which the elements are the partial derivatives of the differential equations used to describe the dynamics of the system state variables, the so-called Jacobian matrix. The eigenvalues are then found through the relation

$$Av = \rho v$$

Here v are the eigenvectors, i.e., vectors which after multiplication by the Jacobian give exactly the same vector except for a constant (the eigenvalues). An eigenvalue $\rho < 0$ indicates stability: after a perturbation the system after some time decays back to the state before the shock. An eigenvalue $\rho > 0$ indicates instability, and after a perturbation the system state deviates further and further from the pre-shock state.

In our case, we are dealing with an ABM. There is no Jacobian, and no stable state in the sense as in the case of differential equation models (as can be deduced from our single run time plot). We are looking for an alternative way of quantifying resilience that suits stochastic, multi-agent simulations.

Above we displayed a histogram of the end point of multiple simulations of the model at the same parameter setting. If we select other end points (e.g., after 500 or 5000 iterations), the resulting histograms might be different compared to our baseline histogram. In the case of the model without the trait inheritance and without shock, this is not the case: an ergodicity test confirms that the distribution stabilizes after several time steps. In an ergodicity test, we compare the histogram of the output of a number of replicate runs at a set time (e.g., t=1000) against the histogram of a single model run measured at a number of time-steps. If both are equal, the model is said to be ergodic. In the case of the model with trait inheritance, adaptation occurs also without shock. The resulting histogram starts to deviate from the baseline histogram after some time, and this deviation becomes larger as 'time' (i.e., the number of iterations) progresses. This is our basis for quantifying resilience.

A common means to compare two histograms, say P and R, is by using the Kullback-Leibler divergence, given as

$$D_{KL}(P||R) = -\sum_{x \epsilon \chi} P(x) \log\left(\frac{R(x)}{P(x)}\right)$$

This measure expresses in how far the two distributions overlap. It is of course possible that two histograms generated from two different sets of model simulations do not overlap at all – and this is also the case here when sufficient iterations have passed in the model with trait inheritance. In that case a limitation of the Kullback-Leibler divergence becomes apparent: it does not say how far away two non-overlapping distributions are. Here we therefore make use of the Earth Mover Distance (EMD) to compare how far away two distributions are. The EMD is explained in more detail in Appendix 3 of the paper by Ten Broeke et al. (https://doi.org/10.1371/journal.pone.0171833.s003), and it is implemented in the code we refer to in the next Section. In essence, the EMD computes the amount of 'work' that would be needed to move a 'pile of earth' (i.e., a histogram) from one place to another (i.e., the second histogram).

The first proposed metric to quantify resilience here is the EMD applied to the histogram of the model without trait inheritance (i.e., random traits) compared to the histogram of the model with trait inheritance:

$$Q(t) = d_e\big(P(x,t), R(x,t)\big)$$

In other words, we run the random model 1000 or 10,000 times and generate the histogram at several different time points. Then, we run the model with trait inheritance for the same number of times, and also generate those histograms. We then compare the respective histograms and quantify their difference by means of the EMD. This difference increases as the number of iterations increases. In a way, we apply a time-dependent sensitivity measure to quantify the change in the system as a result of the system input 'trait inheritance'.

It is not sufficient to quantify the difference between the two model types. After all, the populations in the simulations of both model settings could eventually go extinct. The critical thing here is that many populations in the model with trait inheritance survive, i.e., they seemingly continue indefinitely. The interpretation of resilience in this case is:

- Of what? The persistence of the agent population (indirectly also the resource, because without resource the agents cannot persist, but this is implicit). Individual agents are of no concern, and in fact individuals with lesser 'fitness' have to die in order for the population to continue to adapt. Note, that expressing resilience at the level of the individual would probably be very different;
- To what? Temporary resource shortages at population level (there is no targeted reduction in resource availability at specific locations or for specific agents);

The 'for whom?'-question is not relevant here, as this is a simulation model and no stakeholders are involved here. We also need to quantify the persistence of the agent population in the model with trait inheritance relative to the random inheritance model. In Fig. 11 of Ten Broeke et al. a picture is given that plots λ as a function of E_h, the energy it costs for an agent to make a harvest. The variable λ is calculated based on the following equation

$$N_+(t) = N_0 \mathrm{e}^{-\lambda t}$$

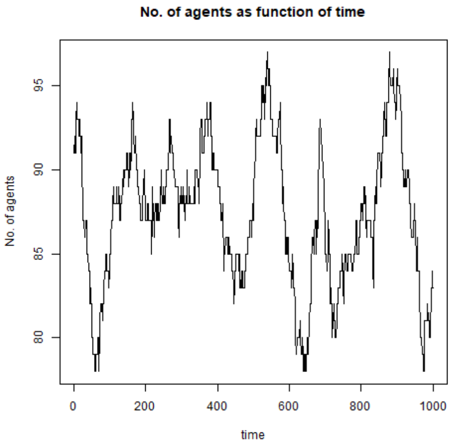
Here N_0 is the total number of simulation runs, and N_+ the number of runs remaining at time t with a persisting population (i.e., number of agents larger than zero). To estimate λ, this equation is fitted on model output from a number of replicate runs. It can be interpreted as an 'extinction parameter' not unlike the dominant eigenvalue of a Jacobian matrix, i.e., it quantifies the rate at which populations go extinct over time under changing E_h. For the random model, λ increases for decreasing E_h. In the inheritance model, the probability of extinction remains near zero as a result of the adaptation of the population. In particular in those runs the EMD will continue to grow as iterations progress. The value of E_h for which there is a transition from $\lambda > 0$ to $\lambda = 0$ can be interpreted as a 'tipping point'. And again, we use a sensitivity analysis approach (OFAT) for quantifying resilience.

The below Section lets you run through the full numerical experiment of going from ABM single simulation to generating the important pictures displaying resilience. ***Warning***: *The simulation time for several steps may be considerable. We do not recommend running this as part of the actual live tutorial, but the file(s) may be helpful for the students to browse and discover*.

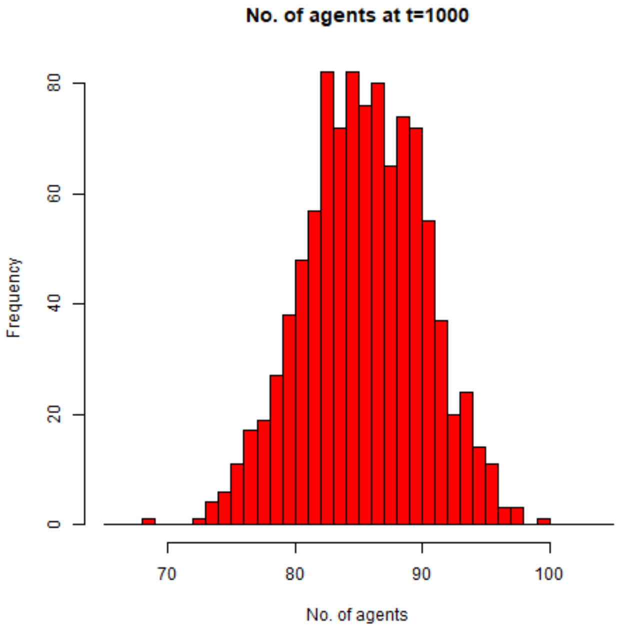
Also note, that you may need to change several lines with path assignments to the correct working directories.

4. The full experiment in one go: an R file to run it yourself

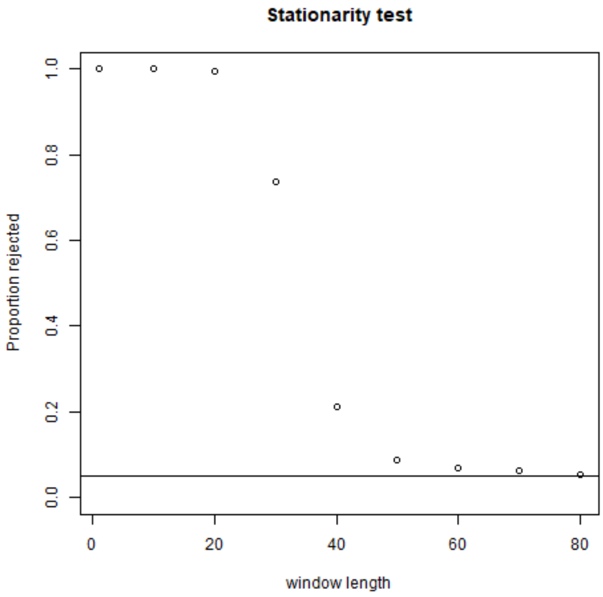
1. The enclosed R file 'tutorial.r' lets you take all the steps going from single run to performing an OFAT analysis involving the parameter 'Eh' (i.e., the amount of energy it costs for an agent to make a harvest). Plot a figure of the model output as function of time in the default parameter setting (single model run).



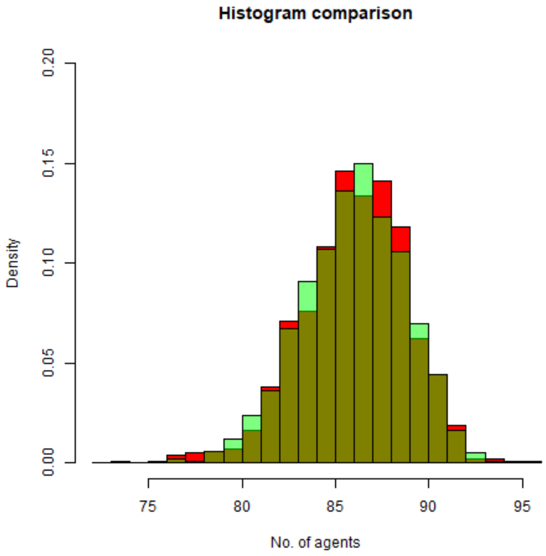
2. Draw a histogram of the model output from 'replicates.csv', at t = 1000, after the model has converged.



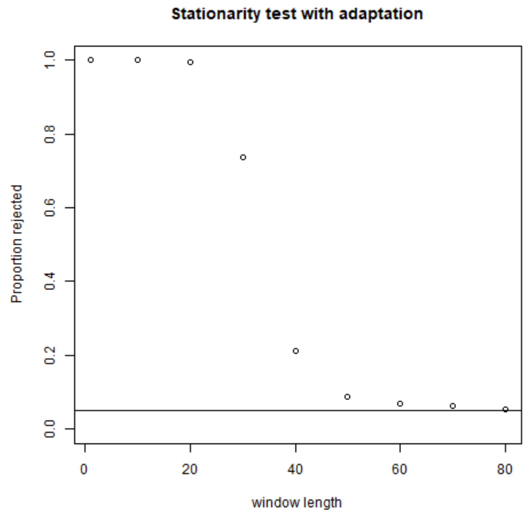
3. Perform a stationarity test, between t = 1000 and t = 2000, to verify that the output has converged before we look into ergodicity. To remove short-term fluctuations, we need to average over time-windows. We do not know in advance what may be a suitable window length. Therefore we try different lengths. For each length, we perform a runs test to verify stationarity. We use a significance level of 0.05, and plot the percentage of model runs for which stationarity is rejected. Around a window-length of 50, the proportion of rejected runs converges to the significance level. This indicates that the output is stationary.

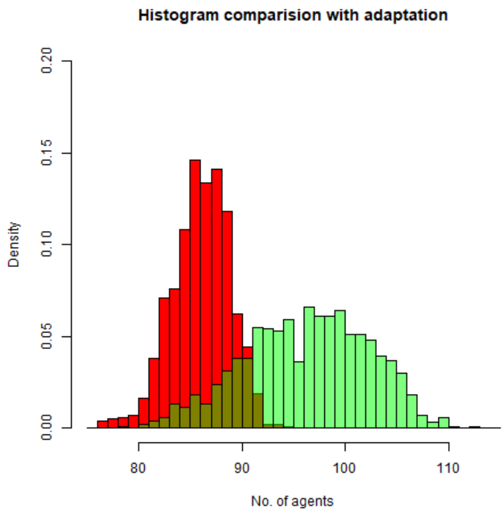


4. Run a single model run for a long time, until t = 100,000. This gives 'timedep.csv'. Compare the histogram over time to the histogram over replicate runs (like in the previous step, we average over time-windows in both cases).

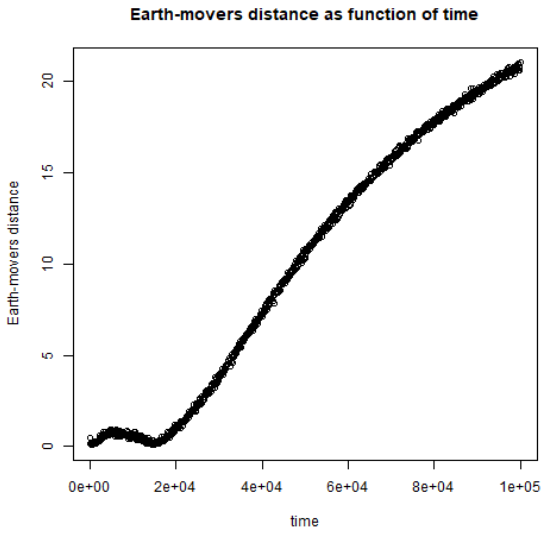


5. Run an ergodicity test. The outcome is that the null-hypothesis that the output is ergodic is *not* rejected.
6. We repeat step 2-5, but now with adaptation added to the model. The output still appears stationary on shorter time-scales in 'replicates_adapt.csv'. But, the histograms show that on longer time-scales (file 'timedep_adapt.csv'), adaptation does affect the model output. The ergodicity test now indicates that the output is not ergodic (i.e., the null-hypothesis of ergodicity is rejected). This is also clear based on a visual comparison of the histograms.

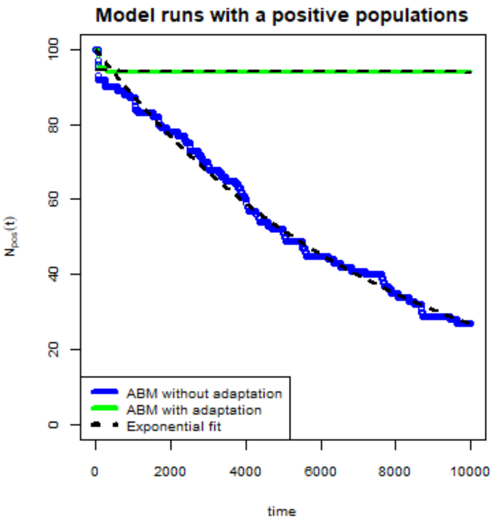


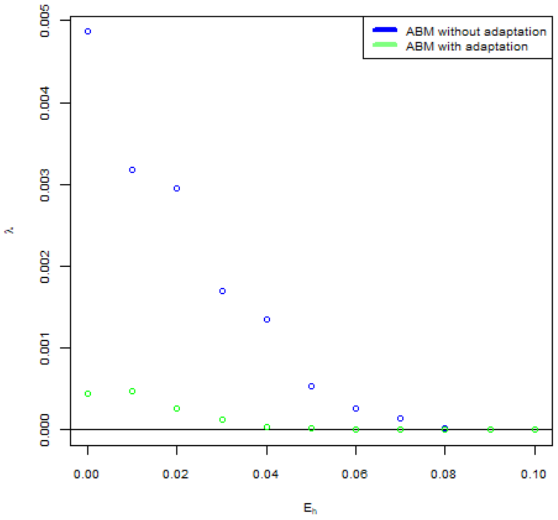


7. The difference between both histograms can be measured by the earth-movers distance. This distance can thus be used to measure the effect of adaptation on the model output. The computation of this distance as a function of time shows that the effects of adaptation increases over time (note: this computation will probably take a few minutes)



8. We perform a OFAT sensitivity for one parameter Eh, which represents the energy cost needed for an agent to harvest during a time-step. For each value of Eh, we use 100 replicate runs (output ‘OFAT_Eh.csv’; note, that this takes a while to run). It turns out there is a tipping point; for large values of Eh harvesting is prohibitively expensive, causing extinction of the population. We investigate how this tipping point is influenced by adaptation. First, we plot for one value of Eh close to the tipping point, how many of the 100 replicate runs maintain a population, as function of time. This number decreases exponentially if adaptation is absent. With adaptation, there is a small initial decrease, after which no further extinction occurs. Thus, it seems that adaptation increases the survival of the population, i.e., its resilience against the types of shocks in this study. Next, we perform the same analysis for the remaining values of Eh. We plot the extinction coefficient, estimated using an exponential fit. The plot confirms that adaptation decreases the rate of extinction.





Thank you for following this tutorial. Please refer to our paper in case you use any of this in your own work or education:
Ten Broeke, Guus, George Van Voorn, Arend Ligtenberg, Jaap Molenaar. "Resilience through adaptation." *PloS one, 12(2),* e0171833 (2017).
The used model can be found on https://www.comses.net/codebases/5374/releases/1.1.0/

#