Synchronisation of fertility with carrying capacity; an investigation using classical and agent based modeling.

Hugh Murrell[*], John Swart[†]

[*]Computer Science, University of KwaZulu-Natal.
[†]Deceased

ABSTRACT

A generalized Verhulst model subject to seasonal change in both fertility rate and carrying capacity is outlined. Numerical solutions to the Verhulst equations are employed to obtain optimal fertility rate phase shift with respect to carrying capacity. Possible natural selection for a preferred season of conception is investigated using agent based simulations. Both experiments indicate that synchronization of fertility rate to environment carrying capacity is beneficial to species survival.

KEYWORDS: population dynamics, differential equations, agent based model, ACM_CCS: G.1.7, G3, I.2.11, I6

1 INTRODUCTION

In 1842 Alfred Lord Tennyson wrote the poem *Locksley Hall* [1] in which appears the immortal line:

In the Spring a young man's fancy lightly turns to thoughts of love

A 50 year study of a conservative Scottish population in the first half of the 20th century undertaken by Russell et al [2] shows sinusoidal trends in birth data that leads one to suspect an environmentally driven fertility cycle. We quote from that paper:

Examination of the birth rhythm show either one or (more frequently) two peaks. The classic European pattern is one with a main peak in spring (summer conceptions) and a smaller one in the autumn (winter conceptions).

In this article we will use a modified Verhulst model to show that the timing of conceptions by a population governed by a seasonal carrying capacity has a marked effect on the resilience of the population as a whole. Our Verhulst model will suggest that the most resilient population will display a growth rate peak at the summer solstice which corresponds to autumn conceptions.

We will then make use of an agent based simulation to demonstrate that seasonal carrying capacity might be responsible for a natural selection in a human population that favors a predisposition to autumn conceptions.

Both the Verhulst model and the agent based simulation are implemented with R scripts that make use of a open source R package `simecol` [3].

Email: Hugh Murrell `murrellh@ukzn.ac.za`

2 A GENERALIZED VERHULST MODEL

A generalized Verhulst model of population growth is given by

$$\frac{dN}{dt} = r(t)N(t)\left(1 - \frac{N(t)}{k(t)}\right) \qquad (1)$$

where at time t, $N(t)$ is the population count, $r(t)$ is the per capita intrinsic growth rate and $k(t)$ is the carrying capacity of the habitat.

In the *standard* Verhulst model r and k are constant and it is well known that if $N(0) < k$, then $N(t)$ tends monotonically to k. See [4] for an analysis of standard Verhulst models.

The above result will be true even if r is a function of time provided k is constant. However, the result may not be true if k is also a function of time. In [5] the authors point out, that in this case $N(t)$ will increase if at time t both $N(t) > k(t)$ and $r(t) < 0$, indicating a breakdown in the model. In this article, in order to avoid this situation, we will restrict $r(t)$ to be non-negative.

We illustrate the generalized Verhulst model using sinusoidals to represent seasonal fluctuations in carrying capacity and growth rate. We model the growth rate as the difference between fertility and mortality. We will synchronize mortality to carrying capacity (a natural assumption) but we will allow a phase shift parameter, ϕ, in our fertility model.

$$\begin{aligned} k(t) &= C_k - A_k\cos(2\pi t) \\ m(t) &= C_m - A_m\cos(2\pi t) \\ f(t) &= C_f - A_f\cos(2\pi(t+\phi)) \\ r(t) &= f(t) - m(t) \end{aligned} \qquad (2)$$

Note that the overall growth rate has the same period as the carrying capacity and to ensure $r(t) \geq 0$ for all phase shifts, ϕ, we require

$$C_f - A_f \geq C_m + A_m \qquad (3)$$

This condition can be interpreted as restricting our model to populations whose fertility is greater than their mortality in the absence of a carrying capacity.

Also note that we have arbitrarily chosen time to start in mid-winter (December 21 in the Northern hemisphere). The middle of the year is the summer solstice, June 21.

In Figures 1 and 2 we demonstrate the effect of phase shift in seasonal fertility rate. In each figure we compute the solutions to the Verhulst model when the initial population is well below the carrying capacity. An R script for reproducing these plots is given in the appendices.

In Figure 1 the phase shift in the fertility is set to $\phi = 0$ (fertility is *in-sync* with carrying capacity) and the solution to the modified model eventually becomes periodic and oscillates just above the mean carrying capacity.

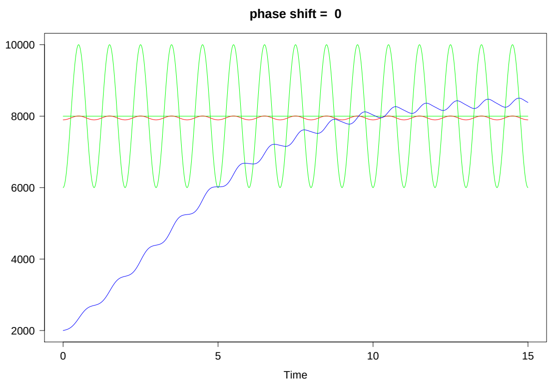

Figure 1: In-sync Verhulst model (blue)
$k(t) = 8000 - 2000\cos(t)$ (green), $r(t) = f(t) - m(t)$ (red)
$f(t) = \frac{2}{3} - \frac{1}{3}\cos(2\pi t)$, $m(t) = \frac{1}{4} - \frac{1}{16}\cos(2\pi t)$

In Figure 2 the phase shift in fertility is set to $\phi = \frac{1}{2}$ (fertility is *out-of-sync* with carrying capacity) and we observe that this could be detrimental to population's survival in the long run as now the solution to the Verhulst model takes longer to recover and does so to a level well under the mean carrying capacity. If such a population were subjected to repeated decimations due to natural disasters it may very well become extinct.

Numerical experiments show that the plots given in Figures 1 and 2 depict the two extremes in the effect of the phase shift parameter. As a sanity check one can drop carrying capacity amplitude parameter and re-run the script to confirm that these extremes will no longer be distinct. Recovery to a *constant* carrying capacity is independent of fertility phase shift.

We have thus observed that a fertility rate in-sync to a fluctuating carrying capacity maximizes the potential for a population to recover after a natural disaster. In the next section we investigate whether this phenomenon could result in a natural selection for a preferred season of conception.

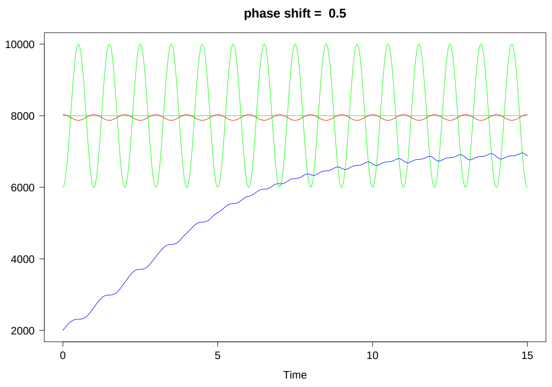

Figure 2: Out-of-sync Verhulst model (blue)
$k(t) = 8000 - 2000\cos(t)$ (green), $r(t) = f(t) - m(t)$ (red)
$f(t) = \frac{2}{3} - \frac{1}{3}\cos(2\pi(t + \frac{1}{2}))$, $m(t) = \frac{1}{4} - \frac{1}{16}\cos(2\pi t)$

3 AN AGENT BASED SIMULATION

We will now employ the same R package, `simecol` [3], to construct an agent based simulation of a maternal population of individuals evolving under the influence of a seasonal carrying capacity. Each individual in our simulation will be female with 3 state variables:

- `age:` the current age (in months) of this individual
- `pcm:` the preferred conception month of this individual (a phenotype inherited from the mother but subject to mutation)
- `fetus:` the current age (in months) of this individual's fetus

The age distribution of the initial population is biased towards young individuals by means of a beta distribution. Initially preferred conception months for each individual are distributed uniformly over the whole year and a proportion of the population possess fetuses of uniformly distributed age.

The time step of the simulation is one month. Again month 0 is arbitrarily centered on the northern hemisphere winter solstice, December 21. Spring is in month 3, Summer in month 6 and Autumn in month 9. The population undergoes monthly updates:

- `live:` each individual and each fetus in the population has their age incremented by one month.
- `survive:` using seasonal carrying capacity and a mortality model, `Siler` [6], individuals are selected for termination and removed from the population.
- `birth:` fetuses reaching full term, with number tempered by carrying capacity, are upgraded to fully fledged individuals and inherit their preferred conception month (subject to mutation) from their mothers.
- `conceive:` individuals in the appropriate age bracket that do not currently possess a fetus and whose preferred conception month is this month are selected for child bearing.

The `simecol` package allows us to evolve the population over time keeping track population size and distribution of preferred conception month. In Figure 3 we show the results of a 250 year long simulation.

The preferred conception month drifts towards month 9 which is Autumn in our time frame.

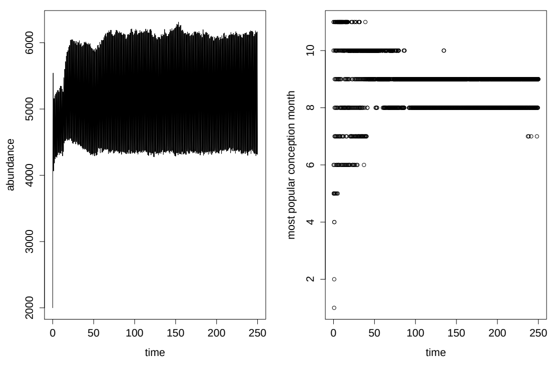

Figure 3: abundance and most popular pcm
$N_0 = 2000, k(t) = 8000 + 2000\cos(t)$

In Figure 4 we give snapshots of the evolution in the distribution of preferred conception month over the running time of the simulation again showing natural selection for Autumn conceptions.

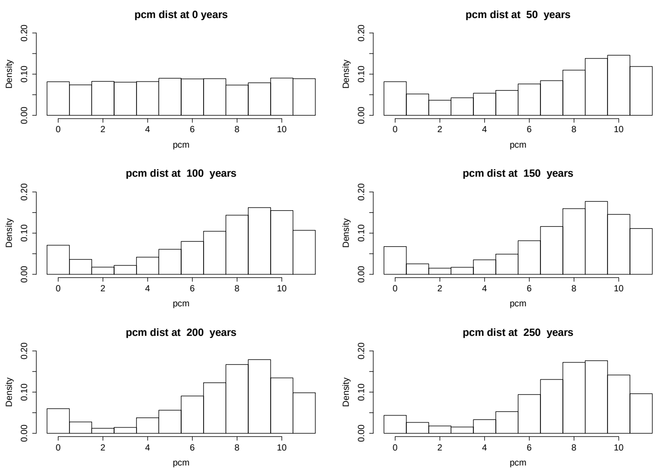

Figure 4: evolution of pcm distribution
$N_0 = 2000, k(t) = 8000 + 2000\cos(t)$

The R script that produced these figures is given in the appendices. It takes about one hour to run on a standard laptop. Setting the carrying capacity fluctuation parameter to zero and rerunning the script confirms that there is no drift in preferred conception month when the carrying capacity is constant.

4 CONCLUSION

The classic Verhulst model suggests that a population will recover best from a calamity if individuals in the population time their conceptions so that growth rate in the population has maximum increase at the peak of the carrying capacity cycle. The agent based simulation of a population evolving under the influence of a seasonal carrying capacity indicates that preferred conception will evolve under natural selection so that growth is greatest at the peak of the carrying capacity. The agent based model confirms the synchronization prediction of the classical differential equation model. However it should be noted that the two models are only weakly supportive of each other. They do not agree on the size of population at which stability is reached or on the time taken to achieve such stability.

The data set alluded to in the introduction indicated two peaks in conceptions, one in Summer and one in Winter. We realize that our models are extremely simple and cannot hope to fully explain real human growth rate data however we would like to suggest that the natural peak in Autumn may have been pulled in two directions due to social issues such as regular Summer and Christmas holidays.

We would also like to suggest that the `simecol` R package is a useful classroom tool for students of population modeling.

ACKNOWLEDGMENTS

I would like to thank the editor and reviewers for their patience in dealing with a computer scientist attempting to ply his trade outside of his normal comfort zone.

OBITUARY

The second author of this paper passed away before the paper was completed but had already made a significant contribution to the research. This short obituary was written by a colleague, Prof James Raftery, who kindly gave the first author permission to include it in this paper.

John Henry Swart, Emeritus Professor and former Head of Mathematics, passed away at his Westville home on 21 July 2012. He is survived by his wife Henda (also an Emeritus Professor of Mathematics at UKZN) and their children Christine, Sandra and Gustav.

John Swart was born on 11 February 1940 in Kimberley. He grew up in Williston. His undergraduate studies and MSc were completed at Stellenbosch University. In 1969 he obtained his PhD under the supervision of Professor Hanno Rund (who was based at UNISA at that time).

John's lifelong and infectious enthusiasm for mathematics influenced the community in which he lived and worked for many decades. His research concerned partial differential equations and biological mathematics. He was an active member of the South African Mathematical Society and served on its executive. He took an interest in local secondary schooling as well, delivering multiple talks at schools and meetings of mathematics teachers.

John's association with UKZN began when he was appointed as a lecturer at the former University of Durban-Westville in 1962. He took up a position at the former University of Natal (Durban) in 1965, where *he served as Head of the Department of Mathematics and Applied Mathematics from 1980 to 1985. He served two further terms as Head of the School of Mathematical Sciences, from 1993 to 2000. He gave very extensive service to the university through the Boards of Science and Engineering, as well as Senate, Council and countless university committees, culminating in a term as Pro-Vice Chancellor. He retired from UKZN in 2005.*

John's vast experience, his forthright and articulate persona, and the breadth of his involvement in university affairs make him an unforgettable figure in the institution's history. As such he will be remembered fondly by colleagues young and old, and by generations of science and engineering students.

REFERENCES

[1] A. l. Tennyson. *Alfred lord Tennyson, Poems.* W. D. Ticknor, Boston, 1842.

[2] D. Russell, A. Douglas and T. Allan. "Changing seasonality of birth - a possible environmental effect". *Journal of Epidemiology and Community Health*, vol. 47, pp. 362–367, 1993.

[3] T. Petzoldt and K. Rinke. "simecol: An Object-Oriented Framework for Ecological Modeling in R." *Journal of Statistical Software*, vol. 22, pp. 1–31, 2007.

[4] A. Tsoularis. "Analysis of Logistic Growth Models." *Res. Lett. Inf. Math. Sci,*, vol. 2, pp. 23–46, 2001.

[5] J. Swart and H. Murrell. "A generalized Verhulst model of a population subject to seasonal change in both carrying capacity and growth rate." *Chaos, Solitons and Fractals.*, vol. 38, pp. 516–520, 2008.

[6] W. Siler. "A competing-risk model for animal mortality". *Ecology*, vol. 60, no. 4, pp. 750–757, 1979.

APPENDIX A

```
# an R script for the Verhulst ode mode
install.packages(pkgs=c("simecol"),
                 repo="http://R.research.att.com")
library(simecol)
help(simecol)

# sinusoidal carrying capacity
k <- function(t, ck, ak){
  ck - ak * cos(2 * pi * t)
}

# sinusoidal mortality in-sync with carrying capacity
m <- function(t, cm, am){
  cm - am * cos(2 * pi * t)
}

# sinusoidal growth via fertility with phase shift
f <- function(t, cf, af, phi){
  cf - af * cos(2 * pi * (t + phi))
}
r <- function(t, cf, af, cm, am, phi){
  f(t,cf,af,phi)-m(t,cm,am)
}

# set up the ode model
ode <- new("odeModel",
           main = function(time, init, parms){
             with(as.list(c(init, parms)), {
               dN1 <- r(time, cf, af, cm, am, phi) *
                 N1 - r(time, cf, af, cm, am, phi) *
                 N1 * (N1/k(time,ck,ak))
               list(c(dN1))
             })
           },
           # (cf-af) should be greater than (cm+am)
           # to ensure positive growth at all times
           # and all phase shifts
           parms = c(cf=2/3, af=1/3, cm=1/4, am=1/16,
                     ck=8000, ak=2000, phi=0.5),
           # start the simulation from
           # mid winter (from = 0)
           # or mid summer (from = 1/2)
           times = c(from = 0, to = 15, by = 0.01),
           init = c(N1=2000),
           solver = "lsoda"
)

# define a special plot function for the model
setMethod("plot", c("odeModel", "missing"),
          function(x, y, ...) {
            o <- out(x)
            p <- parms(x)
            capacity <- function(t){
              k(t,p["ck"],p["ak"])
            }
            growth <- function(t){
              gro <- r(t, p["cf"], p["af"],
                       p["cm"], p["am"], p["phi"])
              if(gro < 0)
                stop("negative growth rate")
              return(gro)
            }
            pop <- o[2]
            ccp <- sapply(o$time,capacity)
            mid <- rep(p["ck"],length(o$time))
            grp <- mid + 200 *
              (sapply(o$time,growth) - p["cf"])
            matplot(o$time,cbind(mid,ccp,grp,pop),
                    xlab = "Time",ylab="",
                    type = "l",
                    lty = c("solid", "solid",
                            "solid","solid"),
                    col = c("green","green",
                            "red","blue"),
                    las = 1,
                    main =
                      paste("phase shift = ",
                            toString(p["phi"]))
            )
          }
)

# run the model
ode <- sim(ode)
plot(ode)
```

APPENDIX B

```
# agent based model of drift
# in prefered concepton month (pcm)
#
library("simecol")

startPop <- 2000
initFert <- 0.5

setClass("agentBasedModel",
         representation(parms = "list",
                        init = "data.frame"),
         contains = "simObj"
         )
abm <- new("agentBasedModel",
           # main simulation loop
           # time is an integer
           # time step is one month
           # init contains a list of all agents
           # parms is a list of model parameters
           main = function(time, init, parms) {
             np <- nrow(init)
             # age the population by one month
             # (agents and their fetuses)
             init <- live(init, parms)
             # apply a mortality function
             # controlled by carrying capacity
             init <- survive(init, parms)
             ns <- nrow(init)
             # cause full-term fetuses
             # to undergo birthing
             init <- birth(init, parms)
             nb <- nrow(init)
             # cause agents with matching
             # prefered conception month
             # to conceive
             init <- conceive(init, parms)
             # report on population, births and deaths
             print(paste("time =",toString(time),
                         "population =",toString(np),
                         "deaths =",toString(np-ns),
                         "births =",toString(nb-ns)))
             init
           },
           parms = list(
             # parameters of the model
             lifeMax = 60 * 12, # (60 years)
             minFert = 12 * 12, # (12 years)
             maxFert = 40 * 12, # (40 years)
             # Siler mortality model parameters
             # m(a) = a1*exp(-b1*a) + a2 + a3*exp(b3*a)
             a1 = 0.5,
             b1 = 0.5,
             a2 = 0.001,
             a3 = 0.003,
             b3 = 0.05,
             # carrying capacity
             # cap(t) = c1 - c2 * cos(2 * pi * t / 12)
             # note maximum in July (t=6)
             Ck = 8000, # mean
             Ak = 2000 # fluctuation
             # growth rate influences both deaths
             # and conceptions each month
             # growthRate = 0.5
           ),
           # we use a Beta distribution
           # to model inital age distribution
           # initially prefered conception month
           # is distributed uniformly
           # initially all agents have not conceived
           init = data.frame(
             age=floor(rbeta(startPop,0.8,2)*60*12),
             pcm=floor(runif(startPop)*12),
             fetus = floor(9 * runif(startPop) *
                     (runif(startPop) < initFert ))
           ),
           # we run the simulation over 250 years
           # takes about 60min to run
```

```
          times = c(from=0, to=250 * 12, by=1),
          # we write our own solver routine
          solver = "myiteration",
          # which makes use of many
          # equations at each time step
          equations = list()
)

equations(abm) <- list(
  # compute the most popular class in a vector
  myMode = function(vec){
    mean(t(which(
      table(vec) == max(table(vec)))-1)[1])
  },
  # sinusoidal carrying capacity
  # with a maximum in July (month 6)
  carryingCap = function(tim){
    with(parms,{
      Ck - Ak * cos(2 * pi * tim / 12)
    })
  },
  # allow for genetic drift in pcm phenotype
  pcmDrift = function(pcm){
    (pcm+sample(c(-1,0,1),1))%%12
  },
  # sample a subset of the population
  # with full-term fetuses.
  # compute new members dependent on
  # current carrying capacity
  newBorns = function(ready,nc) {
    n <- nrow(ready) *
      (carryingCap(time) / nc)^2
    if (n < 1) return(NULL)
    npcm <- sapply(sample(
      ready$pcm,n,replace=TRUE),
                   pcmDrift)
    return(data.frame(age = 0,
                      pcm = npcm,
                      fetus = 0))
  },
  # compute Siler's mortality probability
  # depending on age in years
  mort = function(age){
    with(parms,{
      a1 * exp(-b1*age) +
        a2 + a3 * exp(b3*age)
    })
  },
  # age the agents and fetuses
  # by one month
  live = function(inds, parms){
    with(parms,{
      ninds <- nrow(inds)
      inds$age <- inds$age + DELTAT
      inds$fetus <- ifelse(
        inds$fetus > 0,
        inds$fetus + DELTAT,
        0)
      as.data.frame(inds)
    })},
  # prune the population using
  # mortality model
  # and carrying capacity
  survive = function(inds, parms) {
    np <- nrow(inds)
    killIndividuals = function(age){
      # Siler mortality
      ifelse(runif(1) <
        mort(age/12) *
          (np/carryingCap(time))^3,
            -1,age)
    }
    inds$age <- sapply(inds$age,
                       killIndividuals)
    inds <- subset(inds, age>=0)
    as.data.frame(inds)
  },
  # compute conception hits
  # for a vector of pcms
  # current month
  conceiveProb = function(pcms,time){
    return(pcms == time %% 12)
  },
  # allow agents in fertile age band to conceive
  conceive = function(inds, parms) {
    inds$fetus <- ifelse(
      inds$fetus <= 0 &
      inds$age > parms$minFert &
      inds$age < parms$maxFert,
      inds$fetus + conceiveProb(inds$pcm,time),
      inds$fetus)
    as.data.frame(inds)
  },
  # bring fetuses to life at full term
  birth = function(inds, parms) {
    newinds <- NULL
    retinds <- inds
    nc <- nrow(retinds)
    ready <- subset(inds, fetus > 7)
    newinds <- newBorns(ready,nc)
    retinds$fetus <-
      ifelse(inds$fetus > 7,
             0,
             inds$fetus)
    retinds <- rbind(retinds, newinds)
    return(as.data.frame(retinds))
  }
  )


## a user supplied solver
# with an observer function
myiteration <- function(y, times=NULL,
                        func=NULL, parms=NULL,
                        animate=FALSE, ...) {
  observer <- function(res) {
    # record at each time step
    # population size, mean age, most pop pcm,
    # pcm counts and no of fetuses
    number <- nrow(res)
    meanage <- mean(res$age)
    fetuses <- nrow(subset(res,fetus>0))
    winpcm <- myMode(res$pcm)
    pcmDist <- table(res$pcm)
    c(number=number, meanage=meanage,
      winpcm=winpcm, fetuses=fetuses,
      pcmDist=pcmDist) }
  init <- y@init
  times <- fromtoby(y@times)
  func <- y@main
  parms <- y@parms
  inputs <- y@inputs
  equations <- y@equations
  equations <- addtoenv(equations)
  environment(func) <- environment()
  parms$DELTAT <- 0
  par(mfrow=c(3, 2))
  # plot initial pcm distribution
  hist(init$pcm,
       breaks=seq(from=-0.5,to=11.5,by=1),
       freq = FALSE,
       ylim = c(0, 0.2),
       xlab="pcm",
       main="pcm dist at 0 years")
  res <- observer(init)
  out <- res
  # iterate throgh all time steps
  for (i in 2:length(times)) {
    time <- times[i]
    parms$DELTAT <- times[i] - times[i-1]
    # advance the simulation by one time step
    init <- func(time, init, parms)
    # plot pcm distributions at selected times
    if( (time %% (50 * 12)) == 0 )
      hist(init$pcm,
           breaks=seq(from=-0.5,to=11.5,by=1),
           freq = FALSE,
           ylim = c(0, 0.2),
           xlab="pcm",
           main = paste("pcm dist at " ,
                        toString(time/12),
                        " years"))
    res <- observer(init)
    out <- rbind(out, res)
  }
  row.names(out) <- NULL
  out <- cbind(times, out)
  as.data.frame(out) }


## a plotting function to match observer function
setMethod("plot", c("agentBasedModel", "missing"),
          function(x, y, ...) {
  o <- out(x)
  par(mfrow=c(1, 2))
  plot(o$times / 12, o$number,
       type="l", xlab="time",
       ylab="abundance")
  plot(o$times / 12, o$winpcm,
       type="p", xlab="time",
       ylab="most popular conception month")
})

## RUN the MODEL
abm <- sim(abm)
plot(abm)
```