Series 1: Malthus and Verhulst model

Thibault Schowing

Classical models in biology (exercises) BL.6003

Malthus model

Use the function "ode" to solve numerically Malthus' model. Show the trajectory for the two following set of parameters and initial conditions.

```
library(deSolve)
library(ggplot2)
library(tidyr)
#?ode
```

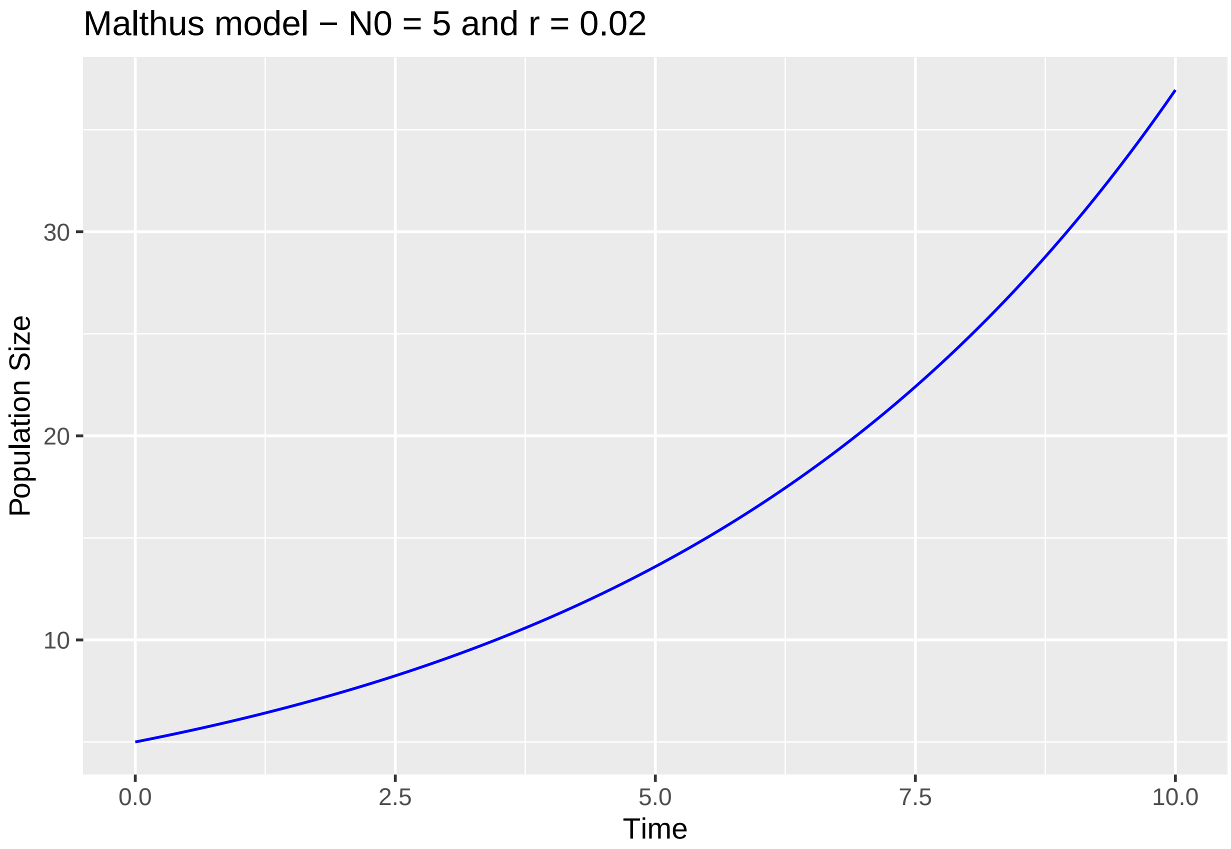
a) $r = 0.2$, and $N_0 = 5$

```
# Parameters passed to func
p <- list(r = 0.2)

# Integrator
m <- c("ode45")

# Differential equation
#
# Params:
#        - t: never used here but must remains. Offers the possibility
#             to have time dependant differential equations.
#        - N: population size at which the time derivative will be evaluated.
#             A list with one element per dimention. For Malthus it is simply
#             a list of 1 element.
#        - p: parameters list

f <- function(t,N,p){
  dN <- p$r * N
  return(list(c(dN)))
}

# Time stemps for the numerical solution
time_steps <- seq(0,10,0.01)

# Initial state values for the ODE system
N0 <- 5

# Run the "ode" function
out <- ode(y = N0, times = time_steps, func = f, parms = p, method = m)
head(out)
```

```
##      time        1
## [1,] 0.00 5.000000
## [2,] 0.01 5.010010
## [3,] 0.02 5.020040
## [4,] 0.03 5.030090
## [5,] 0.04 5.040160
## [6,] 0.05 5.050251
```

Plot of the output

```
# Simple plot
#plot(out, type = "l", xlab = "Time", ylab = "Population Size")

# ggplot looks nicer
data <- as.data.frame(out)
colnames(data) <- c("Time", "Population_size")

ggplot(data, aes(x=Time, y=Population_size)) +
  geom_line(color="blue") +
  ggtitle("Malthus model - N0 = 5 and r = 0.02") +
  xlab("Time") +
  ylab("Population Size")
```



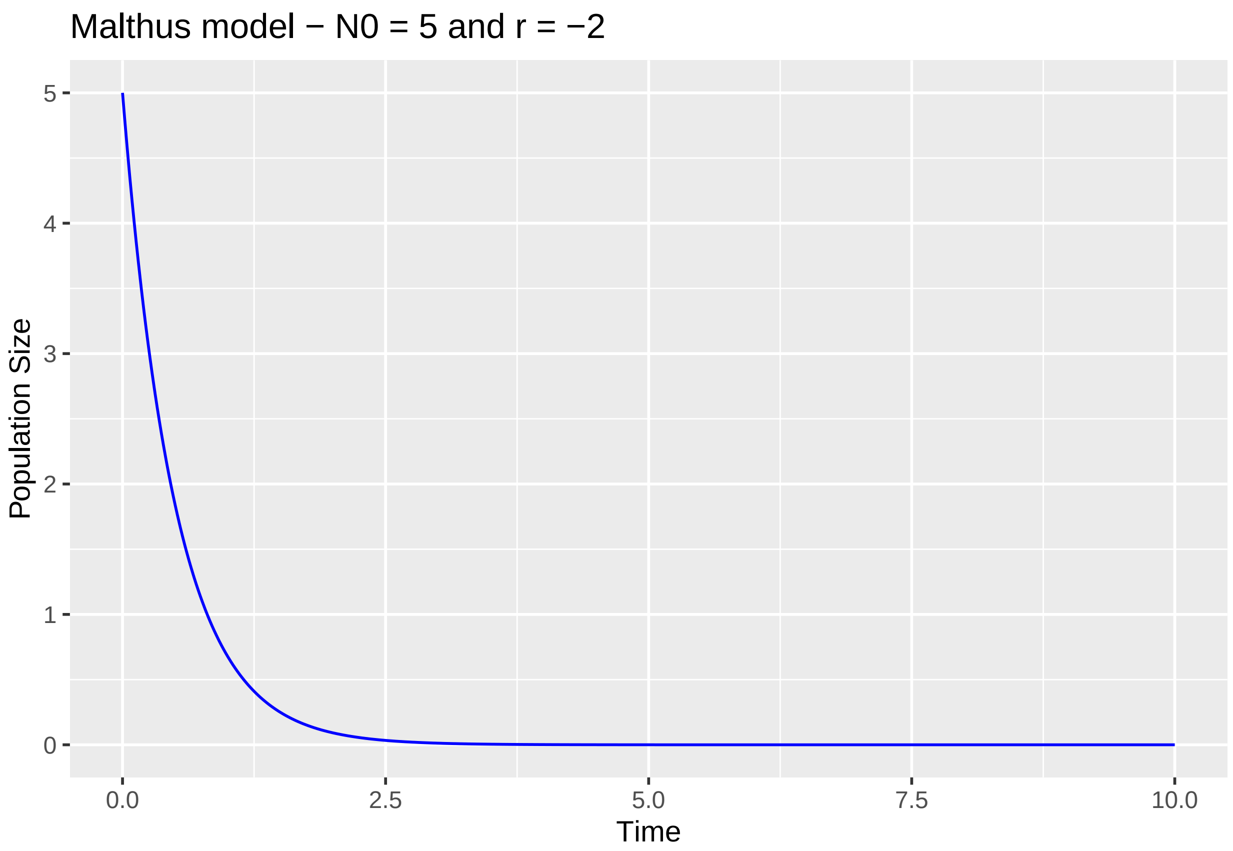
b) $r = -2$, and $N_0 = 5$

```
# Parameters passed to func
p <- list(r = -2)
```

```
# Integrator
m <- c("ode45")

# Differential equation
#
# Params:
#         - t: never used here but must remains. Offers the possibility
#              to have time dependant differential equations.
#         - N: population size at which the time derivative will be evaluated.
#              A list with one element per dimention. For Malthus it is simply
#              a list of 1 element.
#         - p: parameters list

f <- function(t,N,p){
  dN <- p$r * N
  return(list(c(dN)))
}

# Time stemps for the numerical solution
time_steps <- seq(0,10,0.01)

# Initial state values for the ODE system
N0 <- 5

# Run the "ode" function
out <- ode(y = N0, times = time_steps, func = f, parms = p, method = m)
head(out)
```

```
##      time        1
## [1,] 0.00 5.000000
## [2,] 0.01 4.900993
## [3,] 0.02 4.803947
## [4,] 0.03 4.708823
## [5,] 0.04 4.615582
## [6,] 0.05 4.524187
```

```
data <- as.data.frame(out)
colnames(data) <- c("Time", "Population_size")

ggplot(data, aes(x=Time, y=Population_size)) +
  geom_line(color="blue") +
  ggtitle("Malthus model - N0 = 5 and r = -2") +
  xlab("Time") +
  ylab("Population Size")
```



Verhulst model

a) $r = 2$, $\alpha = 0.1$, and $N_0 = 1$.

b) $r = 2$, $\alpha = 0.1$, and $N_0 = 30$.

c) $r = -2$, $\alpha = 0.1$, and $N_0 = 50$.

```
p_a <- list(r = 2, alpha = 0.1)
p_b <- list(r = 2, alpha = 0.1)
p_c <- list(r = -2, alpha = 0.1)


# Verhulst differential
f_verhulst <- function(t,N,p_a){
  dN <- N * (p_a$r - (p_a$alpha * N))
  return(list(dN))
}

time_steps <- seq(0,10,0.05)

N0_a <- 1
N0_b <- 30
N0_c <- 50

out_a <- ode(y = N0_a, times = time_steps, func = f_verhulst,
```

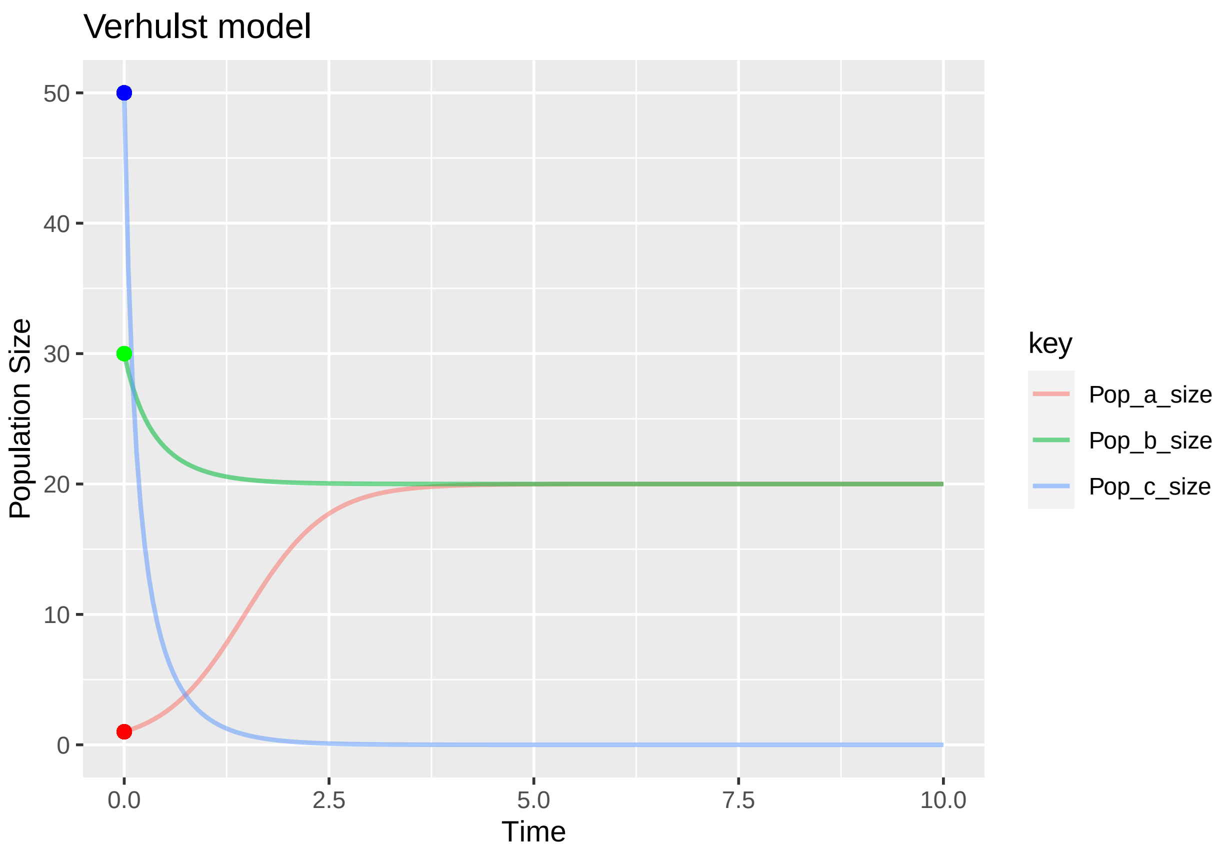
```r
             parms = p_a, method = c("ode45"))
out_b <- ode(y = N0_b, times = time_steps, func = f_verhulst,
             parms = p_b, method = c("ode45"))
out_c <- ode(y = N0_c, times = time_steps, func = f_verhulst,
             parms = p_c, method = c("ode45"))
```

Prepare the Data frame for ggplot2

```r
data <- as.data.frame(out_a)
data["2"] <- out_b[,2]
data["3"] <- out_c[,2]

colnames(data) <- c("Time", "Pop_a_size", "Pop_b_size", "Pop_c_size")
```

Plot of the three population trajectories

```r
test_data <- data
test_data %>%
  gather(key,value, Pop_a_size, Pop_b_size, Pop_c_size) %>%

  ggplot(aes(x=Time, y=value, colour=key)) +
  geom_line(alpha=0.55, size=0.8) +
  geom_point(mapping=(aes(x=0,30)), color="green", size=2) +
  geom_point(mapping=(aes(x=0,50)), color="blue", size=2) +
  geom_point(mapping=(aes(x=0,1)), color="red", size=2) +
  ggtitle("Verhulst model ") +
  xlab("Time") +
  ylab("Population Size") +
  xlim(c(0,10))
```



- Populations:

a) $r = 2$, $\alpha = 0.1$, and $N_0 = 1$.

b) $r = 2$, $\alpha = 0.1$, and $N_0 = 30$.

c) $r = -2$, $\alpha = 0.1$, and $N_0 = 50$.