Solutions for Laboratory 4
Logistic Model

Bio 535, Fall 2011

Introduction

This laboratory focused on the logistic model of population growth. The lab had 4 parts. In part 1 you practiced diagramming the model and in part 2 you again practiced using the Euler and lsoda numerical approximation techniques in part so that you could see how the error generation changed with a different function. My expectation was that you could simply modify your previous scripts to accomplish these tasks. Modifying an existing script can save time, but it also requires that you understand what the program parts do. Part 3 was your first opportunity to really play with modifying control functions. Part 4 was the most complicated part of the laboratory because were to again adapt your scripts to model fish populations using the Shaffer extension of the logistic to include fishing mortality. This is an example of how a small change to the model can make it useful to address a new ecological question, but further complicate the analysis of the model. I present the solutions to each task below.

Task 4.1: Diagram the model

Figure 1 is a diagram of the logistic growth population model using the Forrester diagramming symbols. The major difference between this and the exponential growth model is the addition of the parameter K.

Table 2 summarizes the model elements. This is a good practice to adopt. What are the units of each element?

Task 4.2: Population projections

For this task, you used the exact integral solution of the exponential growth model

$$N_t = \frac{K}{1 + [(K - N_0)/N_0]\, e^{-rt}} \tag{1}$$

to project the future size of the population. You further compared this values to estimates using the Euler solution with $dt = \{1, 0.1, 0.01\}$ and using lsoda. Figure 2 illustrates these solutions.

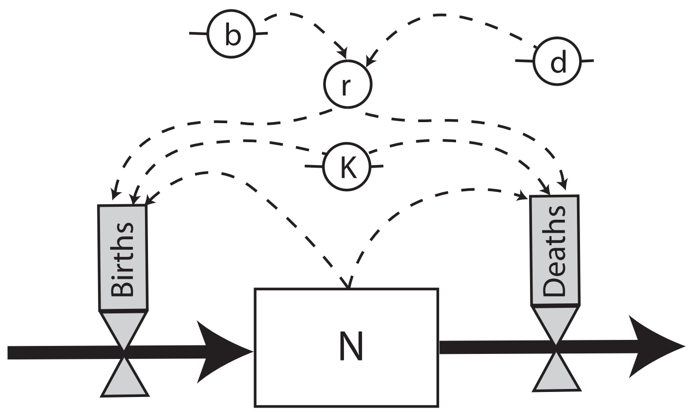

Figure 1: Forrester diagram of the logistic growth model.

Table 1: Variable and parameter definitions and nominal values for Task 6.1

Name	Symbol	Nominal Value
State Variables		
Population Density	N	5
Constant Inputs		
none, this is a homogeneous model		
Specific rate parameters		
birth rate	b	0.5
death rate	d	0.4
Control Parameters		
Carrying Capacity	K	100

Notice that the numerical approximations appear to be very close to the exact solution—nearly indistinguishable. This is reflected in the very small RMSEP values (Table 2). These are much smaller than those we generated for the exponential growth model. The reason for this is that in the exponential model, the error is always compounding in the same direction. The logistic function has a single inflection point at which point the errors made bring the solution closer to the true solution. Essentially, part of the error cancels out the other part.

The lesson you should walk away with here is that error accumulation and propagation in functions can be quite complicated to understand. Thus, we should minimize the errors we introduce into the model to the extent we can. To this end, we will be using only lsoda for our numerical approximations going forward.

Exploring the Effect of Changing the Parameters

I have written an additional program to evaluate the effect of varying the initial population size N_0 and carrying capacity K. This uses the exact solution of the equation. The results are shown in Figure 3.

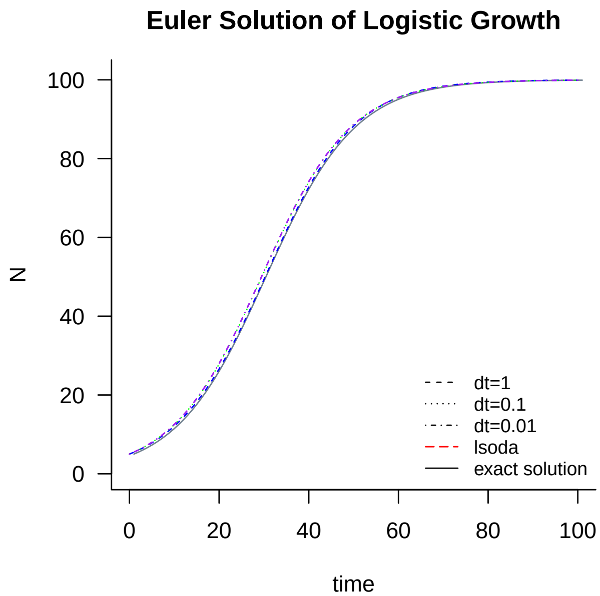

Figure 2: Effect of varying dt on the Euler solution of the logistic growth model

Table 2: Error between the analytical solution and three Euler method numerical approximations of the logisitc growth model as well as the lsoda calculation

dt	RMSEP
1	0.9186
0.1	0.0925
0.01	0.0096
lsoda	6.9899×10^{-5}

Task 4.3: Modifying the Logistic Control Function

Here we investigated the consequences of allowing a non-linear response of the per capita growth rate to population density. We accomplished this with the following function:

$$f(N) = \left(1 - \frac{N}{K}\right)^b . \quad (2)$$

Task 4.3.1: Control Function Plot

The first task was to evaluate the control function for different values of c, the shape parameter (Figure 4). If $c = 1$ we have the original logistic function. If c is less than one, the effect of the density feedback is delayed. That is, the population density can increase for a while before the negative feedback slows population growth. In contrast, when $c > 1$ the density

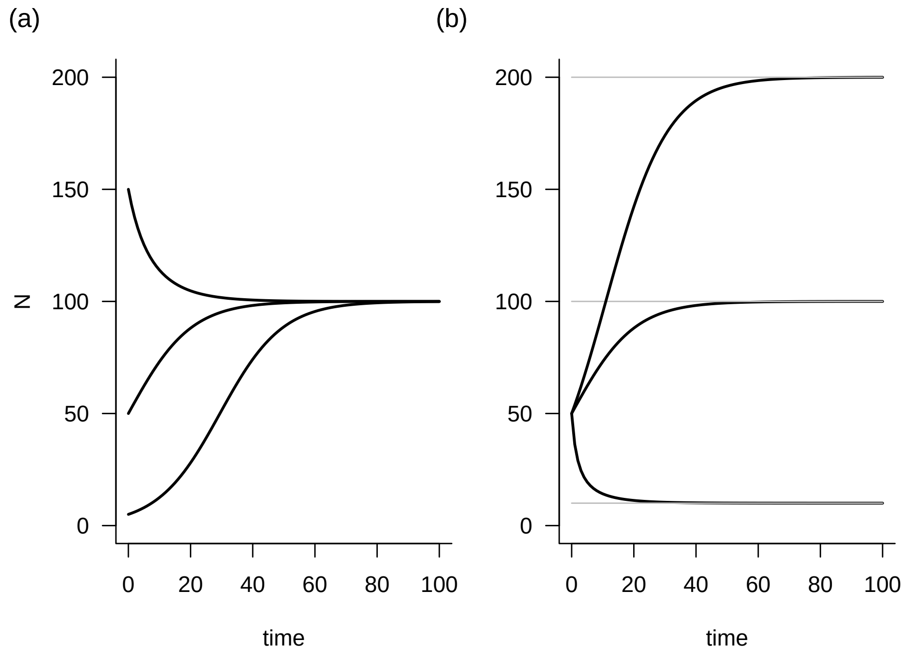

Figure 3: Effect of varying N_0 (a) and K (b) on the exact solutions of the logistic growth model.

effects the growth rate more strongly at lower population density than in the original logistic model.

4.3.2 Population projections

Figure 4 shows how the control function is altered, but it does not show us how this effects the population. To accomplish this, we needed to project the future population using this modified control function. The resultant solutions are shown in Figure 5.

Our results suggest that using the generalized logistic form as described here might make sense when $c \geq 1$, but it is mathematically awkward to use $c < 1$. Thus, this would not be a good way to model a delay or refuge from the density dependence.

Task 4.4: Shaffer Model

The Shaffer model provides us with an example of making a small change to an existing model to address a new question. Here, Shaffer added a term $(-FN)$ to the logistic model to ask a question about the impact of fisheries on the population dynamics. This small change, however, makes analysis of the resultant model more complicated.

Task 4.4.1: Fishing Yield

The first subtask was to plot the change in fishing yield at equilibrium as fishing effort E varied. Figure 6 shows the parabola defined by $Y^* = qEK\left(1 - qE/r\right)$. Again, the maximum

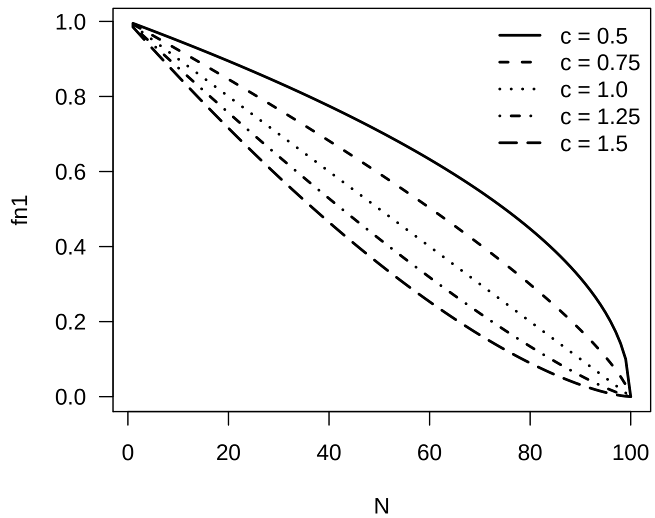

Figure 4: Effect of changing parameter c on the generalized logistic control function response to population density. Here, $fn1 = (1 - N/K)^c$.

occurs at $E^* = r/(2q)$. This is the maximum equilibrium growth rate of the population given fishing. Notice that this is simply another parabolic function.

Task 4.4.2: Intersection of Model Parts

For this task, you were to create two plots. Each plot was to show the growth rate versus population density for the logistic growth function (a parabola). Then, to each plot, you were to add a line showing how the harvest rate (qEN) changes with population density. Figure 7 shows the interaction of these parts. The left hand plot shows a case where the population is under fished (intersection is beyond MSY) while the right hand plot shows a case where the population is over fished (intersection if before MSY).

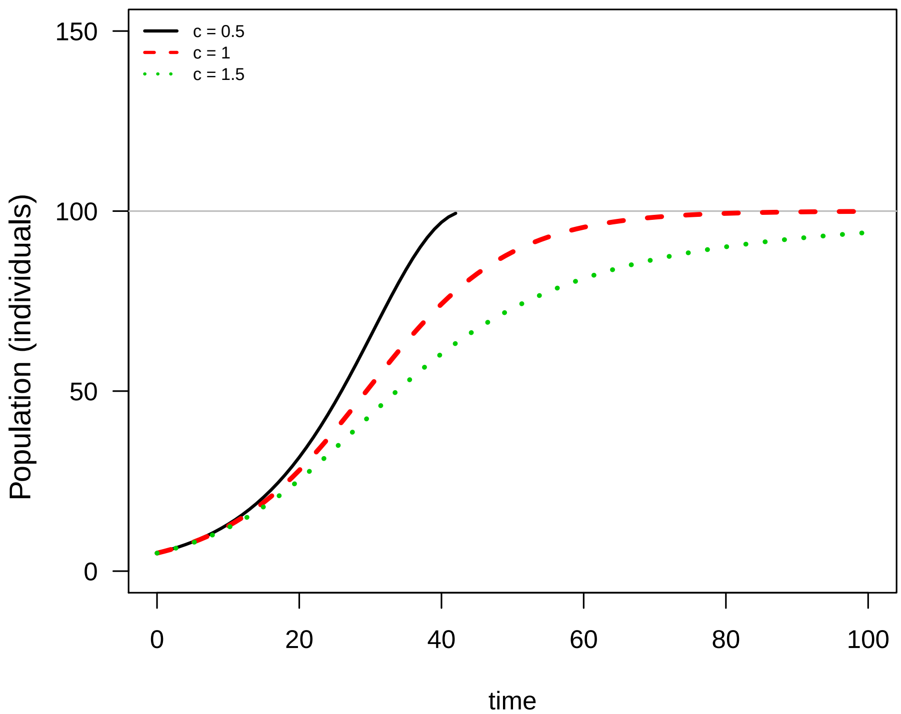

Figure 5: Effect of changing parameter c in the generalized logistic control function on the projected population size. Notice that the projection when $c = 0.5$ is truncated. This is because it stops when the population size would have exceeded carrying capacity. In this case, we were trying to calculate the square root of a negative number, which is mathematically not defined.

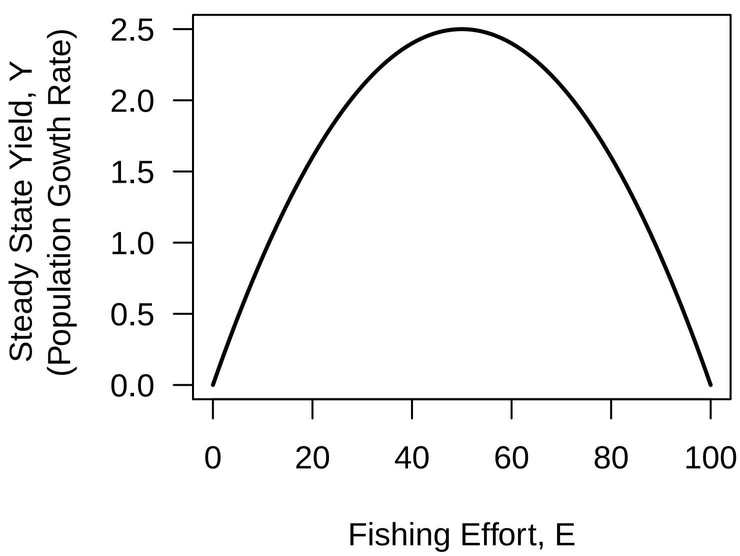

Figure 6: Change in equilibrium fishing yield Y as fishing effort E increases.

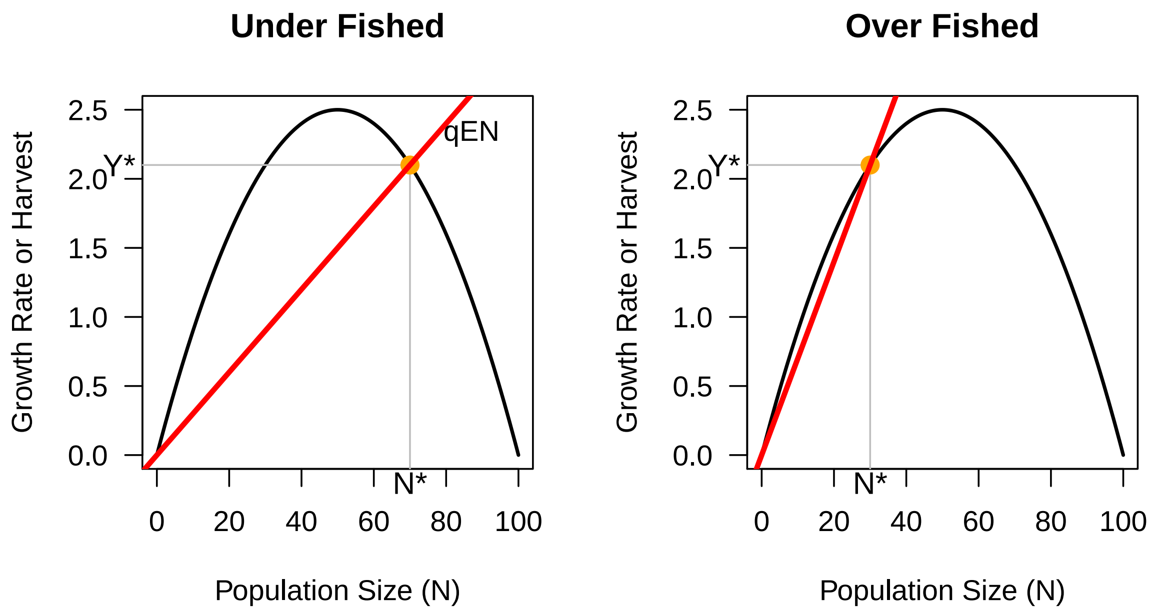

Figure 7: Intersection of the logistic growth curve and the fishing loss line in from the Shaffer model. The left plot shows a case where the population is under fished, while the right plot shows a case of over fished.

Appendix: R Scripts

Euler Solutions for Logistic Growth Model

Below is the script I used to calculate the numerical estimates of population growth using Euler and lsoda methods. It also calculates the error (RMSEP) of these solutions given the exact solution. Notice that I defined a function called "rmsep" to do this calculation because we use the same function multiple times. The "rmsep" function definition follows the main program.

Main Program

```
# Solving Continuous Time Equations: Logistic Growth
# Stuart Borrett
# Oct. 2011
#   Purpose: this script is an example of how to sovle the continous time Logistic
#   model using the Euler solution.
# ------------------------------------------------------
rm(list=ls())
source("rmsep.r");  # loads rmsep function
library("deSolve")
source("logistic.r");

# PROGRAM PARAMETERS
line.type=c(2,3,4,5)
line.color=c("blue","green","purple","grey")

# MODEL PARAMETERS
ti=0     # initial time
tf=100   # final time
tspan=seq(ti,tf,by=1)
parameters=c(b = 0.5,   # per capita birth rate
  d = 0.4,# per capita death rate
  K = 100)              # carrying capacity
state=c(N=5)      # initial states

## -- EXACT ANALYTICAL SOLUTION --##
# Calculate solution
Na <- with(as.list(c(parameters,state)),{
  r=b-d
  Na = (K*N*exp(r*tspan))/(K + N*(exp(r*tspan)-1))  # http://www.math.usu.edu/~powell/ys
  return(Na)
})

                                                    #
```

```
# Plot analytical Solution
fn="../figures/euler-logistic1.pdf"
pdf(fn, height=5,width=5)
opar <- par(las=1,oma=c(1,1,1,1),mar=c(4,4,3,1))
plot(Na,type="l",col="slategray",lwd=1,
     xlab="time",ylab="N",
     main="Euler Solution of Logistic Growth",
     ylim=c(0,101),xlim=c(0,100),
     bty="l")

## -- EULER NUMERICAL APPROXIMATION -- ##
#
# Euler Solutions
N=list()
rmse=list()
dt=c(1,0.1,0.01)
#
for(i in 1:3){   # loop performs dt sensitiity
  N[[i]] = ode(y=state,times=tspan,func=logistic,parms=parameters,
     method="euler",hini=dt[i])
  points(N[[i]],type="l", col=line.color[i],
         lty=line.type[i],
         lwd=1)
  rmse[[i]]=rmsep(Na,N[[i]][,2])
}
#
# lsoda Solution
N[[4]] = ode(y=state,times=tspan,func=logistic,parms=parameters)
rmse[[4]] = rmsep(Na,N[[4]][,2])
points(N[[4]],type="l",
         col=line.color[i],
         lty=line.type[i],
         lwd=1)
legend("bottomright",
       legend=c("dt=1","dt=0.1","dt=0.01","lsoda","exact solution"),
       col=c("black","black","black","red","black"),
       lwd=1,lty=c(line.type,1), bty="n",cex=0.8,
     )
dev.off()   # close PDF
cmd <- paste("open",fn)
system(cmd)
```

Logistic Model Function

```
# Logistic Growth -- Model
```

```
# Borrett, 2011
# ----------------------------------------------------------------------

logistic=function(t,state,parameters){
  with(as.list(c(state,parameters)),{

    # parameters
    r = b-d    # algebraically determine r

    # Aux. Var.
    fn1 = (1-N/K);  # logistic control function

  # equations (ODE)
    dN = r*N*fn1
  return(list(dN)); # returns the solution to the diff eq.
  })
}
```

RMSEP function definition

```
# RMSEP
# this function calculates the root mean squared error between two vectors
# Borrett
# Sept. 2010
# -------------------------

# I am defining this function becuase we are using it multiple times
# in bio534.  This is aalso a good example of defining a function.

# A and B are vectors of the same length

rmsep = function(A,B){
  sum.of.squared.error = sum((A-B)^2);
  normalized.sum.of.squared.error = sum.of.squared.error/length(A);
  rmsep = sqrt(normalized.sum.of.squared.error);
}
```

Logistic Model Sensitivity Analysis

```
# Solving Continuous Time Equations: Logistic Growth
# Stuart Borrett
# Oct. 6, 2008; Modified Oct 2011
#   Purpose: this script is an example of how to sovle the continous time Logistic
#   model using the Euler solution.
#   This builds on euler-solution-logistic.r to explore the effects of changing N0 and K
# ------------------------------------------------------

# MODEL PARAMETERS
ti = 0 # initial time
tf = 100 # final time
N0 = 5 # initial population density
b = 0.5 # per capita birth rate
d = 0.4 # per capita death rate
K = 100                 # carrying capacity
r = b-d                 # calculate intrinsic growth rate

# CALCULATE EXACT ANALYTICAL SOLUTION
t=ti:tf;
pdf(file="logistic2.pdf",width=7,height=5)    # open PDF file
opar <- par(las=1,oma=c(1,1,1,1),mar=c(4,4,1,0),mfrow=c(1,2))

for (N0 in c(5,50,150)){
  # solve model with exact solution
  N = (K*N0*exp(r*t))/(K + N0*(exp(r*t)-1))  # http://www.math.usu.edu/~powell/ysa-html/

  if(N0 == 5){
    plot(t,N,type="l",col="black",lwd=2,
         xlab="time",ylab="N",
         #main="Varying Initial Condition",
         ylim=c(0,200),xlim=c(0,100),
         bty="l")
    mtext("(a)",side=3,adj=0,cex=1.2,outer=TRUE)
  } else {
    points(t,N,type="l",lwd=2,col="black") }
}

N0=50;
for (K in c(10,100,200)){
  # solve model with exact solution
  N = (K*N0*exp(r*t))/(K + N0*(exp(r*t)-1))
        # http://www.math.usu.edu/~powell/ysa-html/node8.html
```

```
  if(K == 10){
    plot(t,N,type="l",col="black",lwd=2,
         xlab="time",ylab="",
         #main="Varying Carrying Capacity",
         ylim=c(0,200),xlim=c(0,100),
         bty="l")
    mtext("(b)",side=3,adj=0.5,cex=1.2,outer=TRUE)
    points(t,rep(K,length(t)),type="l",lwd=1,col="grey")
  } else {
    points(t,N,type="l",lwd=2,col="black")
    points(t,rep(K,length(t)),type="l",lwd=1,col="grey")
  }
}

dev.off()   # close PDF file
rm(opar)

system("open -a preview logistic2.pdf") # on the Mac this
                                          # command opens the PDF file
```

Control Function Plot

```
# Logistic Control
# Bio534, logistic-control.r
# Borrett, 2011
# ---------------
rm(list=ls())
dev.off()

# parameters
b = 0.5                         # specific birth rate
d = 0.4                         # specific death rate
r = b - d                       # intrinsic growth rate
K = 100                         # Carrying Capacity
cvec= c(0.5,0.75,1,1.25, 1.5);   # Shape parameters

# Start Plot
fn="../figures/control.pdf"
pdf(fn,height=4,width=5)
opar <- par(las=1,oma=c(0,0,0,0),mar=c(4,4,1,1))  # change default plotting parameters
#
# -- LOOP to evaluate effect of c
for(i in 1:length(cvec)){
  # Control Function
  N = 1:100
```

```
  fn1 = (1-N/K)^cvec[i]
#
  # plot
  if(i ==1){
    plot(N,fn1,lwd=2,type = "l")
  } else {
    lines(N,fn1,lwd = 2, lty = i)
  }
}
#
legend("topright",
       legend=c("c = 0.5","c = 0.75","c = 1.0",
                 "c = 1.25","c = 1.5"),
       lwd = 2, lty=1:5,bty="n" )
#
rm(opar)    # reset default plotting parameters
dev.off()   # close PDF file
cmd <- paste("open",fn)   # create system command (mac)
system(cmd)               # execute system command (mac)

# Alternative Code
# ---------------------------------------
# tmp <- sapply(c,function(x) (1-N/K)^x)
# matplot(tmp,type="l")
# legend("topright",
#       legend=c("c = 0.5","c = 0.75","c = 1.0",
#                 "c = 1.25","c = 1.5"),
#       lwd = 2, lty=1:5,bty="n",col=1:5 )
```

Population Projections with Modified Logistic

Run File

```
# Logistic Model -- Run File
# - uses a generalized logistic function.
# Borrett, 2011
# -----------------------------------------------------------------------------
rm(list=ls())
# install any necssary libraries not normally loaded with the base package
library(deSolve)

#### PROGRAM INPUT ####
# make sure the model function is loaded.
source("logistic-model-mod.r")
```

```
# Set Initial Values and Simulation Time
tspan=0:100      # vector of times for which you want a solution

# Assign Parameter Values
p=c(b = 0.5,   # per capita birth rate
  d = 0.4,# per capita death rate
  K = 100)              # carrying capacity
state=c(N=5)      # initial states
#
c.vec = c(0.5,1,1.5);  # vector of possible c values

out=list()
fn <- "../figures/logistic_mod.pdf"
pdf(fn, height=5, width=6)
opar<-par(las=1,oma=c(1,1,0,0),mar=c(4,3,1,1))
#
for(i in 1:length(c.vec)){
  parameters<-c(p,c=c.vec[i]);   # changes c value
  show(parameters)
  #
  #### Program CORE ACTION #######
  # Numerically Solve the differential equation(s)
  out[[i]] = ode(y=state,times=tspan,func=logistic,parms=parameters)
  #
  if(i==1){
    plot(out[[i]],type="l",
         lwd=2,
         col=i,ylim=c(0,150),
         xlab="time",ylab="",main=""
         )
    #
    mtext("Population (individuals)",side=2,line=0,
          las=0,outer=TRUE,cex=1.2)
    abline(h=as.list(p)$K,col="grey",lwd=1)
    #
  } else {
    points(out[[i]],col=i,lwd=3,lty=i,type="l")
  }
}
legend("topleft", legend=c("c = 0.5","c = 1", "c = 1.5"),
       lty=1:3, lwd = 2,bty="n",cex=0.7,col=1:length(c.vec),
       )
dev.off()
rm(opar)
cmd <- paste("open",fn)
```

```
system(cmd)
```

Model

```
# Logistic Growth -- Model
# Borrett, 2007
# --------------------------------------------------------------------

logistic=function(t,state,parameters){
    with(as.list(c(state,parameters)),{

    # parameters
    r = b-d    # algebraically determine r

    # Aux. Var.
    fn1 = (1-N/K)^c;  # logistic control function

  # equations (ODE)
    dN = r*N*fn1
  return(list(dN)); # returns the solution to the diff eq.
  })
}
```

Shaffer Model

Schaffer Equilibrium Yield

```
# Shaffer Equilibrium Yield
# Based on problem in Mangel 2006
# Borrett
# Sept. 2010
# -------------------------------

# parameters
r = 0.1;
K = 100;
q = 0.001;

E=0:110;

Y = q*E*K*(1-q*E/r);

fn <- "../figures/shaffer_equilibrium_yield.pdf"
pdf(file=fn,height=3,width=4)
```

```
opar <- par(las=1,oma=c(0,0,0,0),mar=c(4,5,1,1))
plot(E,Y,type="l",lwd=2,
     ylab="Steady State Yield, Y \n (Fish caught per unit time)",
     xlab="Fishing Effort, E",
     )
abline(h=0,col="grey")
points(c(50,50),c(0,2.5),type="l",col="red")
dev.off()
cmd <- paste("open",fn)
system(cmd)
```

Maximum Sustainable Yield

```
# Plotting Maximum Sustainable Yield
# Schaefer Model
# as in Mangle 2006
# Borrett
# Sept 2011
# ------------------------------------

fn <- "../figures/msy.pdf"
pdf(fn,height=4,width=7)
opar <- par(las=1,mfrow=c(1,2))
#
r=0.1
K=100
q=0.001
q = 0.1
for(i in 1:2){
if(i ==1){ E = 0.3} else {E=0.7}
#
NS = K*(1-q*E/r)
Y = q*E*K*(1-q*E/r)
#
N=seq(0,100,by=0.5)
f=r*N*(1-N/K)
#
if(i==1){tmp="Under Fished"} else {tmp="Over Fished"}
plot(N,f,
     type="l",lwd=2,
     ylab="Growth Rate or Harvest",
     xlab="Population Size (N)",
     main=tmp
     )
#
```

```
points(c(-5,NS),c(Y,Y),
       type="l",
       col="grey")
points(c(NS,NS),c(-5,Y),
       type="l",
       col="grey")
#
points(NS,Y,pch=20,cex=2,col="orange")
abline(0,q*E,col=2,lwd=3)
if(i==1){mtext("qEN",side=3,line=-1.5,adj=0.9)}
#
text(-15,Y,labels="Y*",cex=1.1,
     adj=0,xpd=TRUE)
text(NS,-.2,labels="N*",cex=1.1,
     xpd=TRUE)
}
dev.off()
cmd <- paste("open",fn)
system(cmd)
```