Laboratory 6 Solutions

Building Two State Variable Models
for Consumer–Resource Dynamics

Task 6.1: Two entity models with logistic space-control feedback

For this task, I first constructed two R files – a model and a run file – to simulate the two-state variable model described in the lab handout (Appendix A). Again, the model equations were

$$\frac{dX_1}{dt} = C - \delta_1 X_1 - \tau_{12} X_2 \cdot f(X_2) \cdot f(X_1) \quad (1)$$

$$\frac{dX_2}{dt} = \tau_{12} X_2 \cdot f(X_2) \cdot f(X_1) - \delta_2 X_2 \quad (2)$$

Where

$$f(X_2) = \left(1 - \left(1 - \frac{\delta_2}{\tau_{12}}\right)\left(\frac{X_2}{K_2}\right)_+\right)_+, \quad (3)$$

$$f(X_1) = \frac{X_1}{X_1 + 0.001} \quad (4)$$

and the variables and parameters are defined as in Table 1. Control function $f(X_1)$ is designed to mimic a step function so that if there is any phosphorus available then the uptake rate is maximized, but if phosphorus goes to zero then its uptake goes to zero. We assumed 100% efficiency in the conversion of phosphate to phosphorus in phytoplankton.

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

Name	Symbol	Nominal Value
State Variables		
Phosphorus	X_1	2
Phytoplankton	X_2	0.01
Constant Inputs		
Input of available phosphorus	C	0.5
Specific rate parameters		
Loss from available phosphorus pool	δ_1	0.001
Loss of phosphorus from phytoplankton	δ_2	0.08
Uptake (gross) of phosphorus by phytoplankton	τ_{12}	0.3
Control Parameters		
Maximum density of phytoplankton	K_2	10

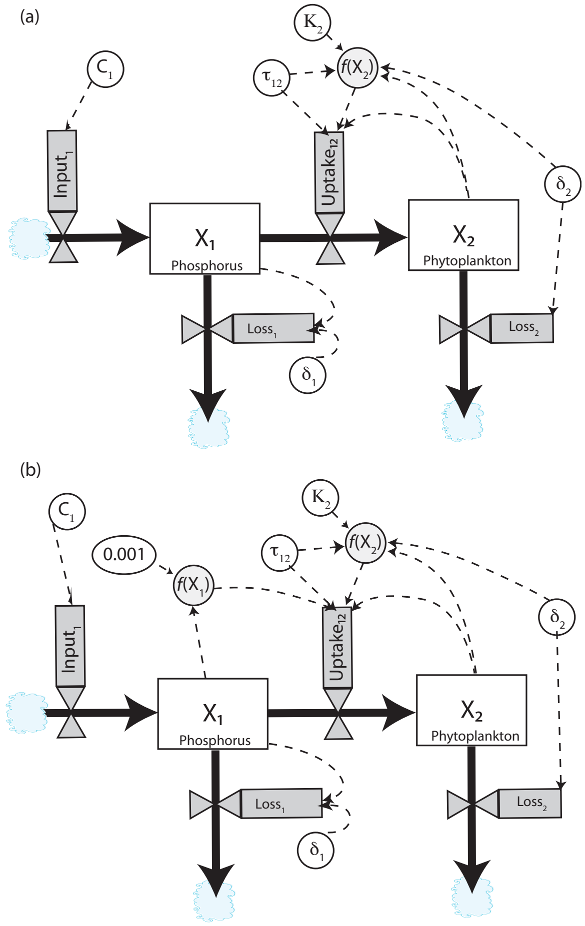

Figure 1: Forrester diagrams for the first task model. (a) shows the control arrangement as described where the model is only recipient controlled and (b) shows the addition of the donor control function ($f(X_1)$) that we added so that the phytoplankton could not consume more phosphorus than was physically available.

Task 6.1.1: Forrester Diagram

Figure 1 shows a Forrester type diagram for this assignment. The critical component for this exercise is to recognize that the uptake of phosphorus by the phytoplankton is controlled by the phytoplankton density.

Task 6.1.2: Model Sensitivity to K

For this subtask, I modified my original model program to analyze the sensitivity of the model behavior to the carrying capacity (K). I chose to use K values of 2, 5, 7, 9, 12, and 15. The results of these changes appear in Figure 2.

Results The graphs in Figure 2 show that when the phytoplankton (consumer) carrying capacity is low, the phytoplankton quickly grow to their environmental or space limited density. In this case, the Phosphorus nutrient (resource) continues to grow without bound because there is a constant input and eventually a constant loss. However, as the phytoplankton carrying capacity increases, their biomass becomes great enough so that their phosphorus consumption (plus the environmental loss) exceeds the value of the rate of resource replenishment (C). Thus, as the carrying capacity increases the effective control on the phytoplankton biomass switches from intraspecific control (competition for space or some set of unspecified resources) modeled by the logistic to a resource limitation or exploitative control. Also, notice that even when the phosphorus concentration is zero, the algae persist. This is because they are immediately consuming all of the incoming phosphorus (C).

Task 6.1.3: Enrichment Variation

Results & Discussion Figure 4 illustrates that increasing the phosphorus replenishment rate C pushes the system toward a regime in which the phytoplankton are controlled by their own density (intraspecific competition for an unspecified resource like space) and the increase in phosphorus is unchecked. As mentioned, this model broadly mimics what might happen when an ecosystem becomes eutrophic.

Task 6.1.4: Variation in Phytoplankton Initial Density

Results & Discussion Figure 4 suggests that increasing the initial phytoplankton density alters the system dynamics. There are several interesting changes that occur. First, as $X_2(0)$ increases, the quantity of algae at steady state does not change; it ends at X2 6.3. Second, when the algae starts really small, it initially takes up phosphorus slower than its replaced, so it accumulates unit the algae is abundant enough to consume all of the available phosphorus. Third, if $X_2(0)$ starts larger than the carrying capacity, then the phosphorus can initially increase because the phytoplankton uptake of the phosphorus is limited by their density dependent feedback. Ultimately, however, phosphorus becomes the limiting factor in all cases shown.

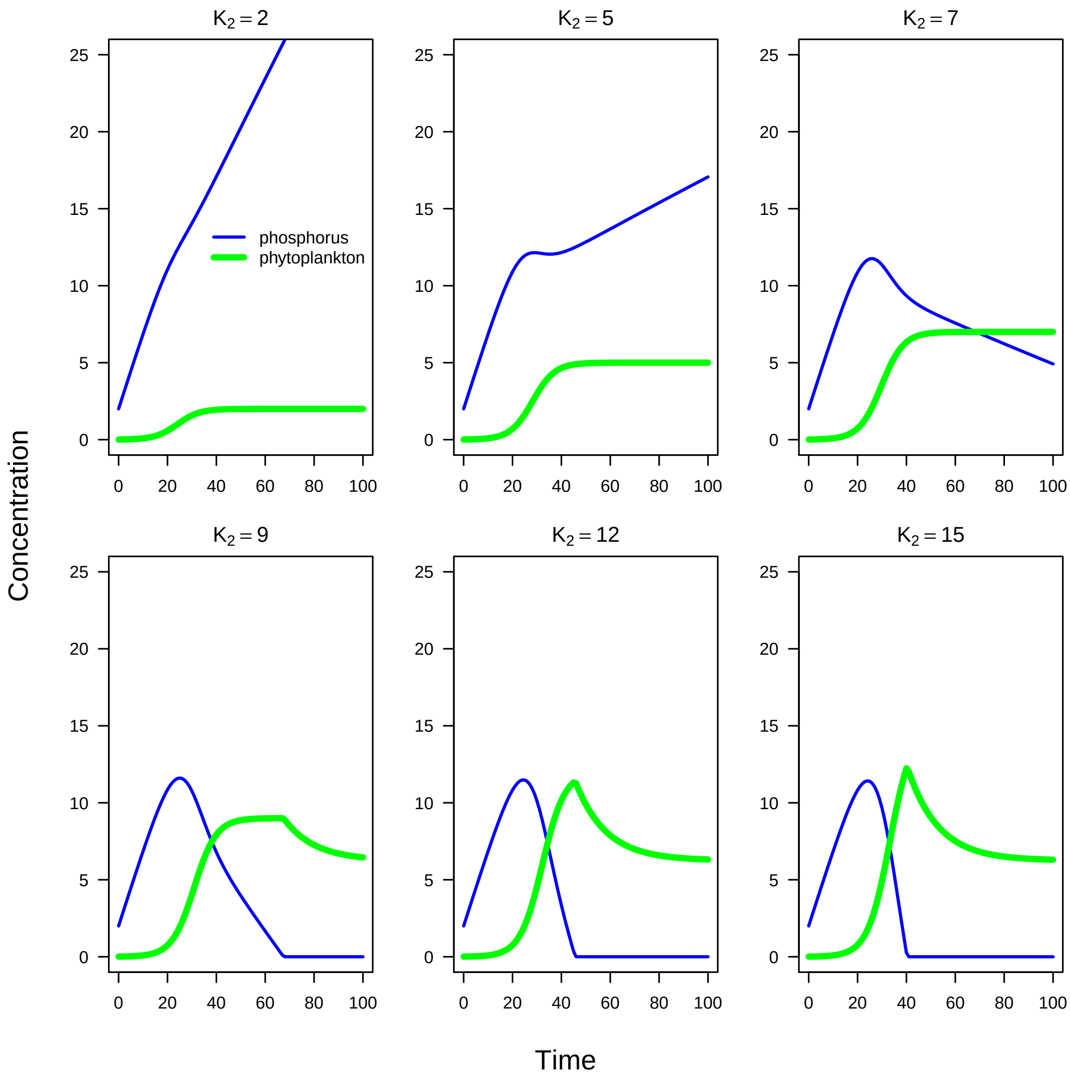

Figure 2: Sensitivity of two-entity model with logistic space-control feedback to the carrying capacity. Each panel shows the system dynamics under a different carrying capacity value ($K = \{2, 5, 7, 9, 12, and 15\}$).

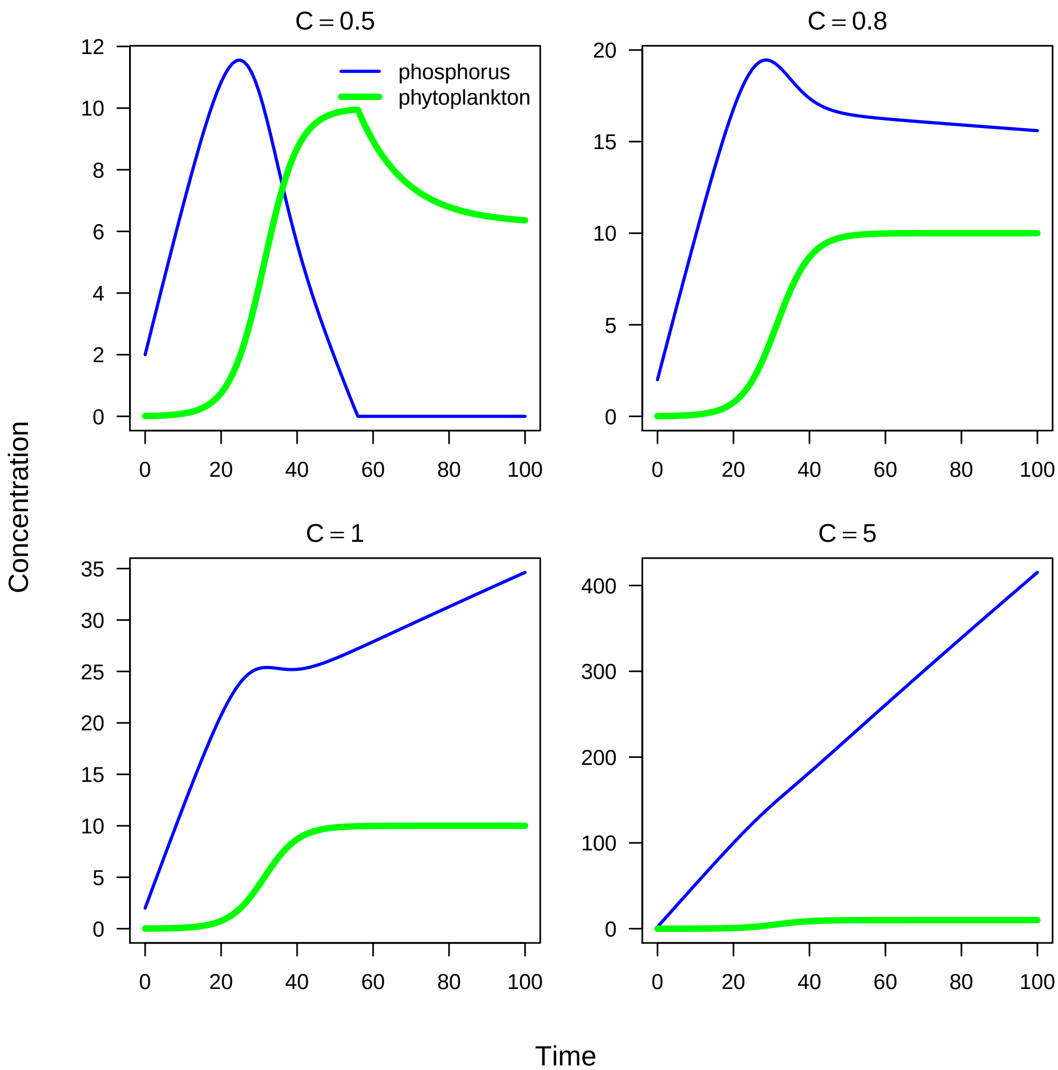

Figure 3: Sensitivity of two-entity model with logistic space-control feedback to the rate of nutrient replenishment. Each panel shows the system dynamics under a different phosphorus replenishment rate ($C = \{0.5, 0.8, 1, 5\}$).

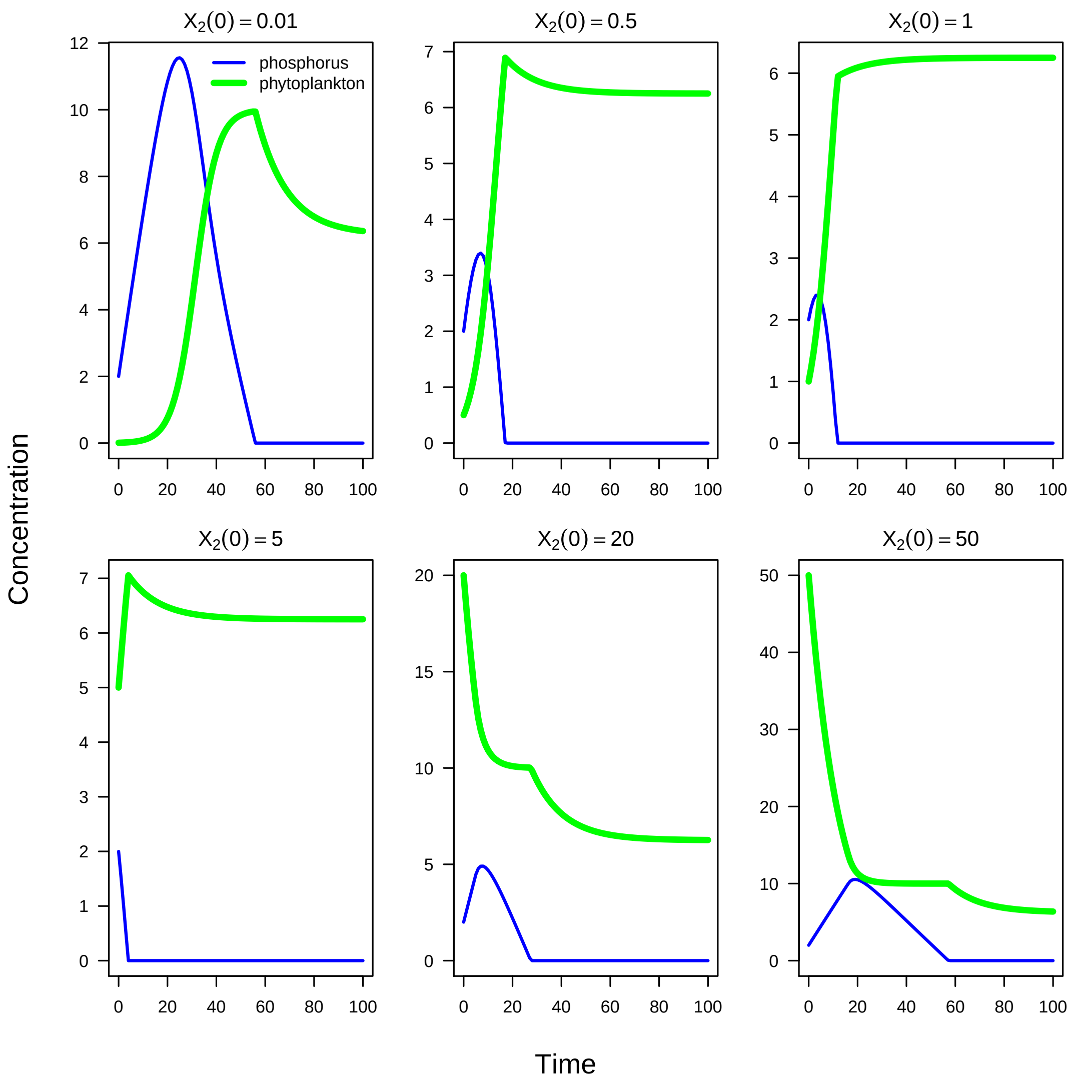

Figure 4: Sensitivity of two-entity model with logistic space-control feedback to the initial value of Phytoplankton (X_2). Each panel shows the system dynamics under a different initial Phytoplankton biomass ($X_2(0) = \{0.01, 0.5, 1, 5, 20, 50\}$).

Task 6.2: Two entity models with logistic space-control feedback

For the second task, I again wrote a series of R programs that inlcuded a model embedded in a run file. The model is based on the following equations. Unless otherwise noted, I used the same variables, parameters, and nominal values as shown in Table 1.

Equations

$$\frac{dX_1}{dt} = C - \delta_1 X_1 - \tau_{12} X_2 \cdot f(X_1) \quad (5)$$

$$\frac{dX_2}{dt} = \tau_{12} X_2 \cdot f(X_1) - \delta_2 X_2 \quad (6)$$

Where $f(X_1)$ can be one of several functional forms

$$f(X_1) = \left(1 - \frac{k_1}{(k_1 + X_1)}\right)_+ \quad (7)$$

$$f(X_1) = \left(1 - \frac{\alpha_{12}}{X_1}\right)_+ \quad (8)$$

$$f(X_1) = \left(a \frac{X_1^b}{X_1^b + k_1^b}\right)_+ \quad (9)$$

Where equation (7) is the half-saturation form of the hyperbolic control function for resource uptake, equation (8) is the refuge form of hyperbolic resource control, and equation (9) is the modified Michaelis–Menten or Monod functional response.

Task 6.2.1: Forrester Diagram

Figure 5 shows a Forrester type diagram for this assignment. The critical component for this subtask is to recognize that the uptake of phosphorus by the phytoplankton is only controlled by the donor.

Task 6.2.2: Hyperbolic Form of the Michaelis–Menton Function

Results & Discussion With this formulation of resource control, a higher 1/2 saturation value k_1 implies that it takes a greater resource concentration for the uptake process to reach its maximal values (Figure 9). As Figure 6 illustrates, this translates into a faster initial accumulation of phosphorus in the system, a larger steady-state value, and a slower growth of the phytoplankton. Notice that when k_1 is 5 or 10, the steady state quantity of phosphorus is larger than phytoplankton, but this reverses when k_1 is 20.

Task 6.2.3: Hyperbolic Form of the Michaelis–Menten Function with Refuge

Results & Discussion The refuge form of the Michaelis–Menten control function prevents the phytoplankton form consuming the phosphorus until a certain threshold concentration is achieved α_{12}. Like the increase in k_1 in the 1/2 saturation form, an increase in α_{12}, shown in Figure 7, delays the phosphorus and phytoplankton peak concentrations. The maximum and steady state phosphorus quantities generally increase with α_{12}, but there appears to be little effect on the phytoplankton quantities.

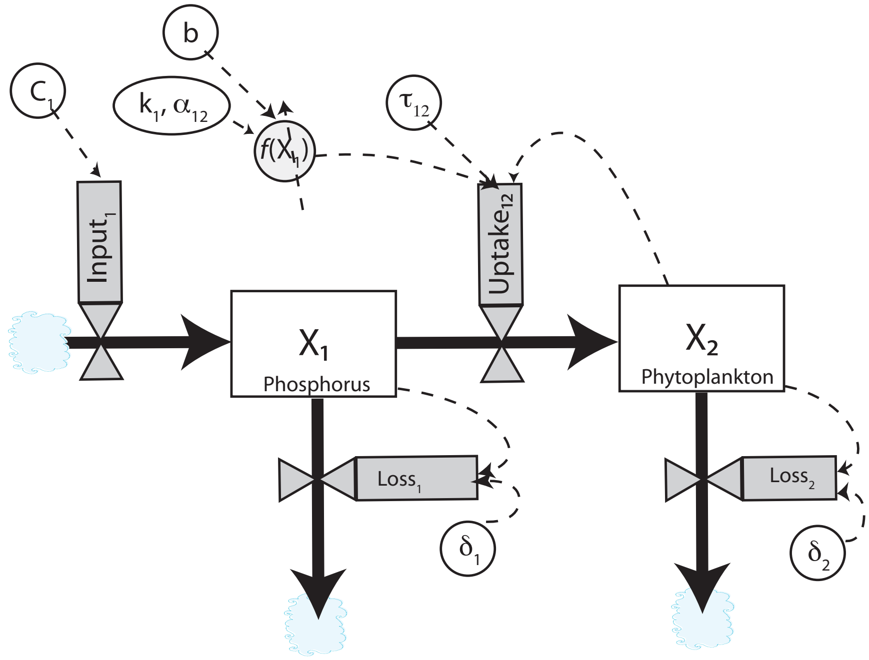

Figure 5: Forrester diagram for the model in the second task. Notice that Uptake_{12} is now only donor controlled.

When α_{12} was set to 0.2 or less (not shown), many of the model values became undefined. This is why some of your plots stop before the model time is completed. Does it make sense for α_{12} to be less than 1?

Task 6.2.4: Generalized Form of the Michaelis–Menten Function

Increasing the exponent b in the generalized form of the Michaelis–Menton control function caused the peak quantities to occur earlier (Figure 8). When all else was equal, the increase in b caused uptake to be smaller at lower resource concentration values, but the transition to reduced uptake occurred more rapidly as the 1/2 saturation constant was approached. This is evident in the realized control function plots in Figure 8 and in the potential control function behavior illustrated in Figure 9.

Task 6.2.5: Classifying the Controls

Figure 9 shows the *potential* behavior of the three control functions we investigated in the second task. This shows that the first two functions have a Holling Type II functional response, while the third function can be made to have a sigmoid or Holling Type III response when b is greater than unity.

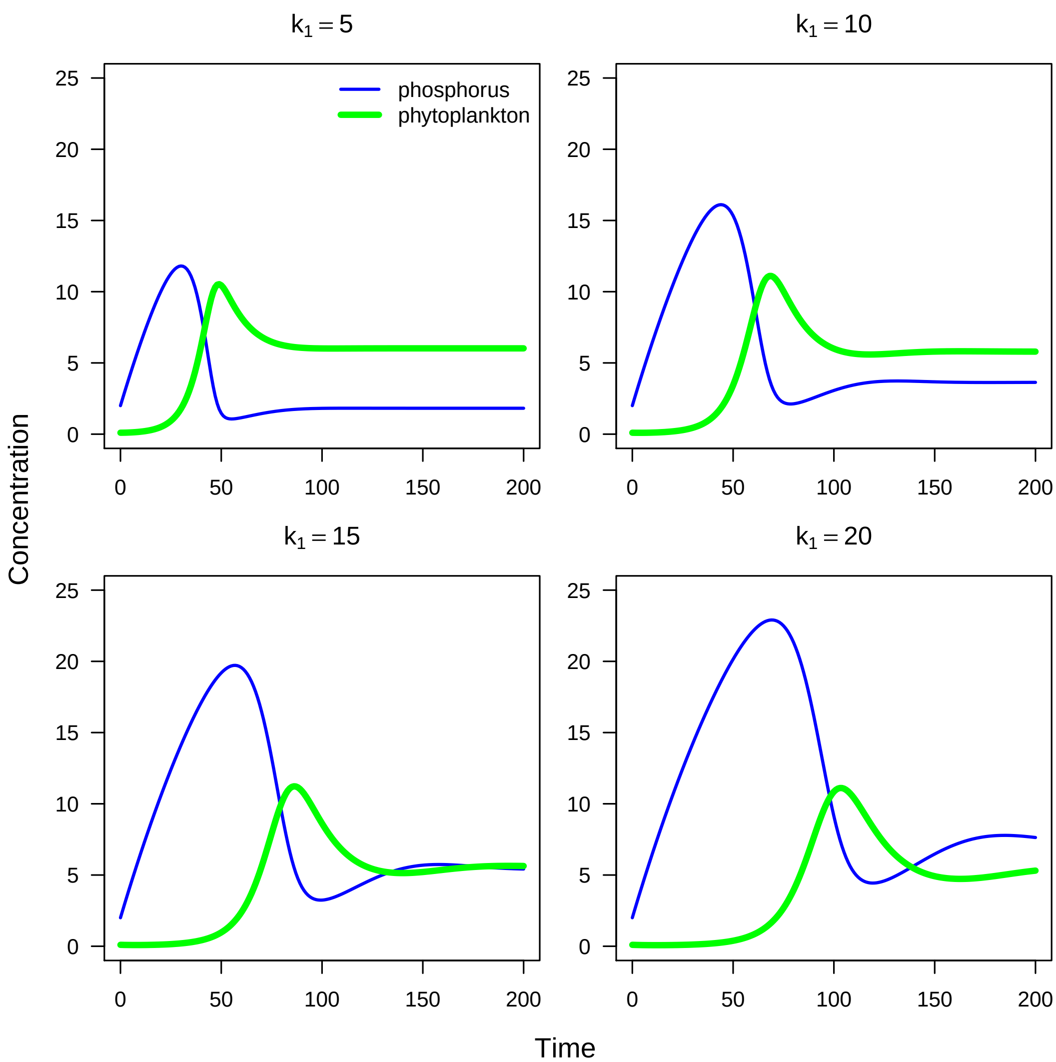

Figure 6: Hyperbolic form of the 1/2 saturation control used to control phytoplankton (X_2) uptake of phosphorus (X_1). This is an example of donor control of consumption (exploitative control). Each panel shows the system dynamics under a different 1/2 saturation constant ($k_1 = \{5, 10, 15, 20\}$).

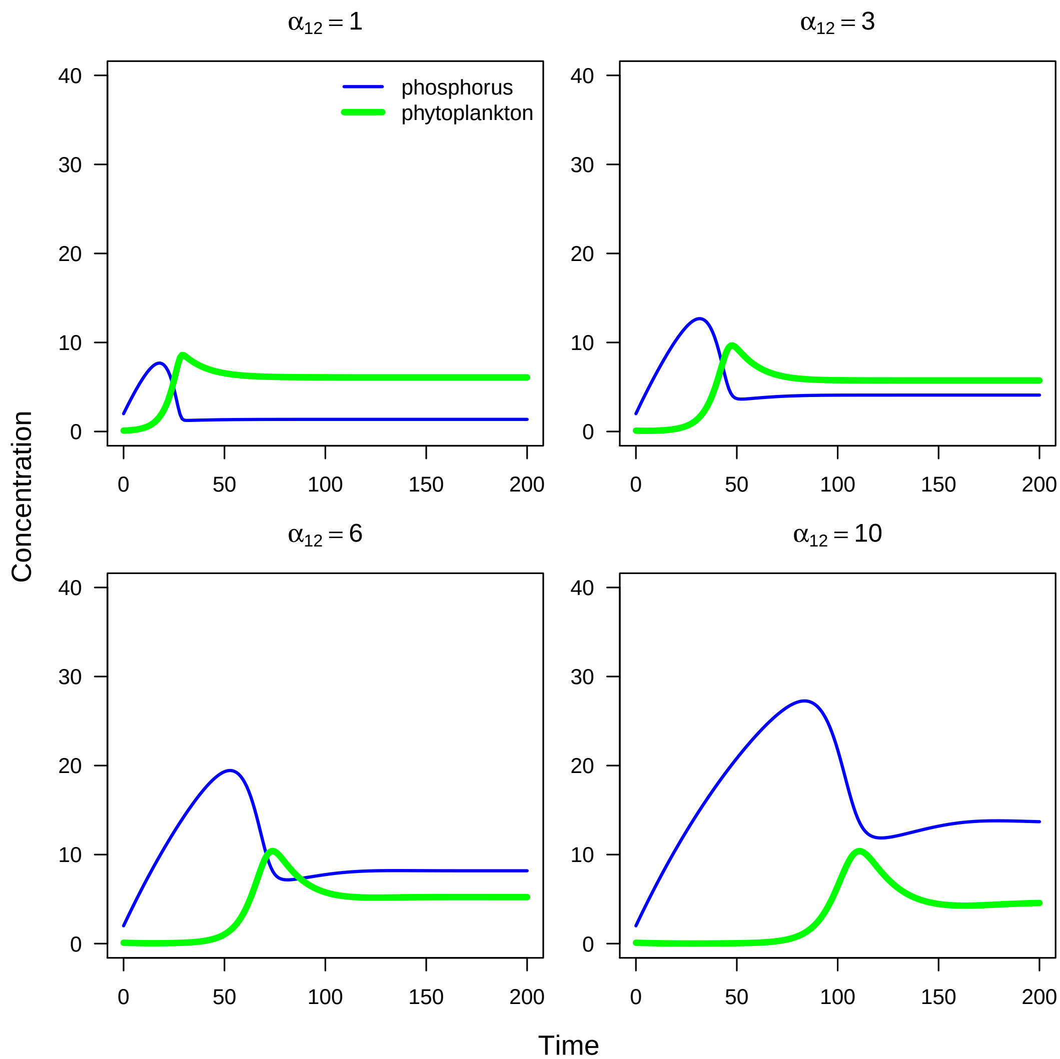

Figure 7: Hyperbolic form of the Michaelis-Menten control function with a refuge used to control phytoplankton (X_2) uptake of phosphorus (X_1). This is an example of donor control of consumption (exploitative control). Each panel in the right column shows the system dynamics under a different refuge value ($\alpha_{12} = \{1, 3, 6, 10\}$). Panels on the right show the realized behavior of the control function $f(X_1)$

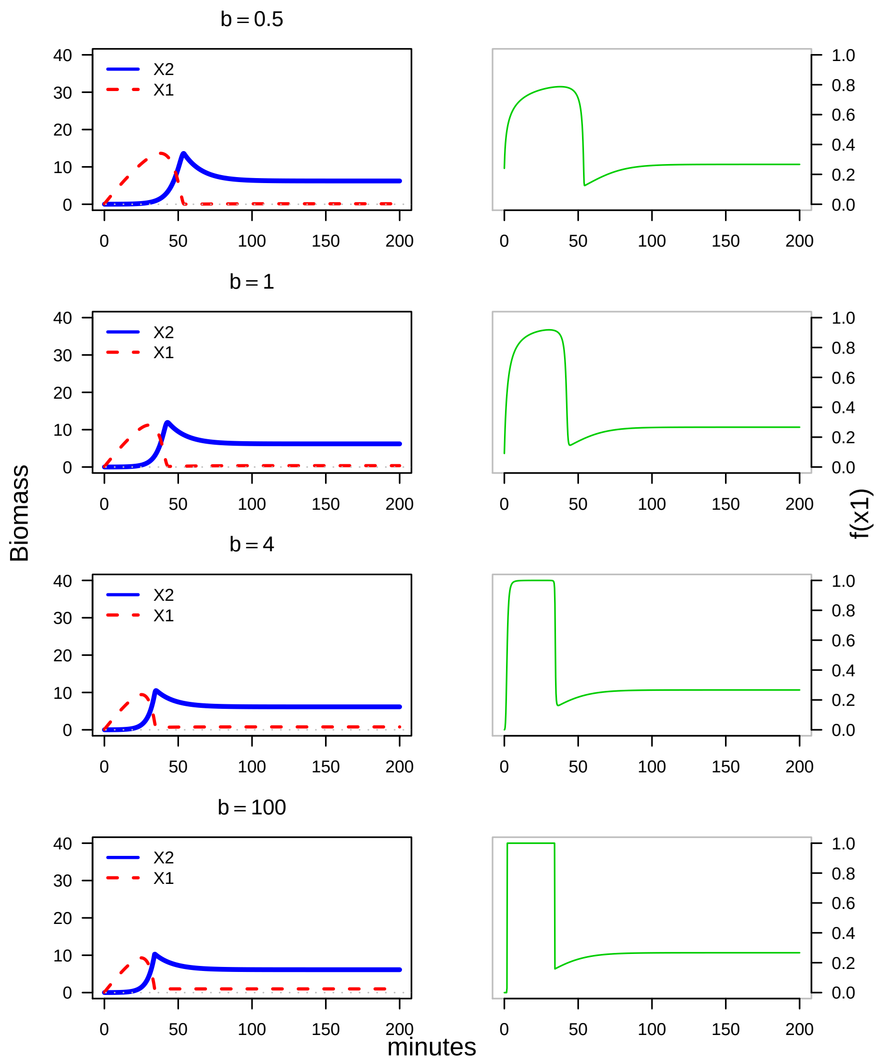

Figure 8: Generalized form of Michaelis–Menten 1/2 saturation control function used to control phytoplankton (X_2) uptake of phosphorus (X_1). This is an example of donor control of consumption (exploitative control). Each panel in the right column shows the system dynamics when $k_1 = 10$, and $b = \{0.5, 1, 4, 100\}$).

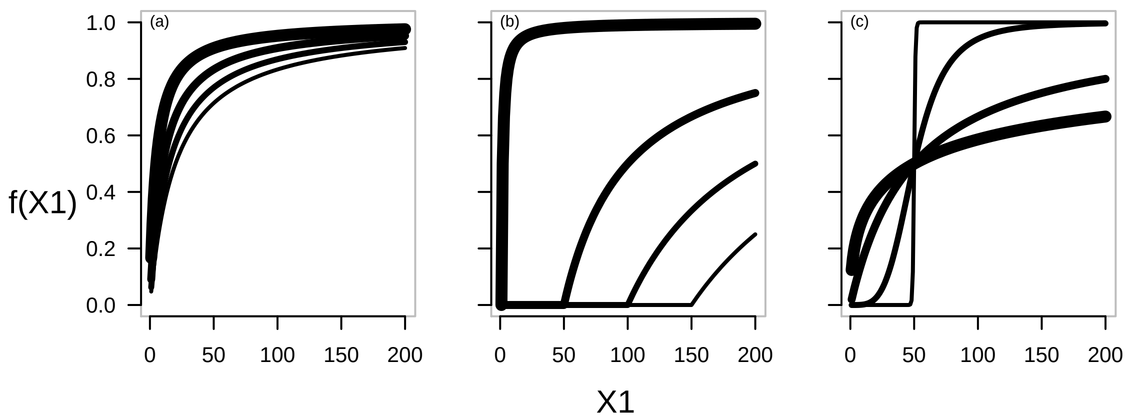

Figure 9: Potential values of the (a) hyperbolic form of the Michaelis-Menten 1/2 saturation control function, (b) Hyperbolic Form of the Michaelis–Menten Function with Refuge, and the (c) generalized form of Michaelis–Menten control function.

Appendices

A Programs for Task 6.1.2

```
# Laboratory 6.1 - Run (NOMINAL)
# BIOL534
# Borrett
# October 2011
# ---------------------------------

rm(list=ls())
library(deSolve)
#source("L6t1-model-new.r")

## --- MODEL --- #
model=function(t,state,parameters){
  with(as.list(c(state,parameters)), {

    # Auxillary Variables
    wc = 1-delta2/tau12
    fX1 = X1/(X1+k1)
    fX2 = pmax(0, (1 - wc * pmax(0, X2/K2)))

    # Diff Eqs.
    dX1 = C - delta1 * X1 - tau12*X2 * fX2 * fX1
    dX2 = tau12 * X2* fX2 * fX1 - delta2*X2
    return(list(c(dX1,dX2)))
  })
}

## -- RUN MODEL -- ##

# Program Parameters

# Model Parameters
ti = 0
tf = 100
tspan=seq(ti,tf,by=1)

fn <- "plot-L6t1t2.pdf"
pdf(fn,height=7,width=7)
opar <- par(las=1,mfrow=c(2,3),mar=c(3,3,2,1),oma=c(2,2,0,0))
K2.vec=c(2,5,7,9,12,15)
for(i in 1:length(K2.vec)){

parameters=c(C = 0.5,
  delta1 = 0.001,
  delta2 = 0.08,
  tau12 = 0.3,
  K2 = K2.vec[i],
  k1=0.00001)

state = c(X1 = 2,
```

```
  X2 = 0.01)

out=ode(y=state,times=tspan,func=model,parms=parameters)

plot(out[,1],out[,2],col="blue",type="l",lwd=2,
     ylim=c(0,25),
     main=bquote(K[2] == .(K2.vec[i])))
points(out[,1],out[,3],col="green",type="l",lwd=4)

if(i ==1){
legend("right",
       legend=c("phosphorus","phytoplankton"),
       pch=NA,
       lwd=c(2,4),
       col=c("blue","green"),bty="n")
}

}

mtext("Time",side=1,line=1,outer=TRUE,cex=1.1)
mtext("Concentration",side=2,line=0.8,outer=TRUE,las=3,cex=1.1)

dev.off()
cmd <- paste("open",fn)
system(cmd)
```

B Program for Task 6.2

```
# Laboratory 6, Task 2
# BIO534
# Borrett, Oct 2011
# ------------------------------
rm(list=ls())
library(deSolve)

## MODEL ------------------------------------
model=function(t,state,parameters){
  with(as.list(c(state,parameters)), {

    # Control Functions
    if(sw ==1){fX1 = pmax(0,(1 - k1/(X1+k1)))    }
    if(sw == 2){fX1 = pmax(0,(1 - alpha12/X1))  }
    if(sw == 3){fX1 = pmax(0, X1^b/(X1^b+k1^b)) }

    # Diff Eqs.
    dX1 = C - delta1 * X1 - tau12*X2 * fX1
    dX2 = tau12 * X2 * fX1 - delta2*X2
    return(list(c(dX1,dX2)))
  })
}
# ---

## Model Run ----------------------------------
```

```
ti = 0
tf = 200
tspan=seq(ti,tf,by=1)

# set up plot
fn <- "plot-L6t2t1.pdf"
pdf(fn,height=7,width=7)
opar <- par(las=1,mfrow=c(2,2),mar=c(2,2,3,1),oma=c(2,2,0,0))

k1.vec <- c(5,10,15,20)
for(i in 1:length(k1.vec)){

parameters=c(C = 0.5,
  delta1 = 0.01,
  delta2 = 0.08,
  tau12 = 0.3,
#  K2 = 10,
  k1=k1.vec[i],
  alpha12 = 1,
  b = 10,
  sw = 1)  # switches which control function is used

state = c(X1 = 2,
  X2 = 0.1)

# solve model
out=ode(y=state,times=tspan,func=model,parms=parameters)

# output
plot(out[,1],out[,2],col="blue",type="l",lwd=2,
     ylim=c(0,25),ylab="",xlab="",
     main=bquote(k[1] == .(k1.vec[i])))
points(out[,1],out[,3],
       col="green",type="l",lwd=4)
if(i ==1){
legend("topright",
       legend=c("phosphorus","phytoplankton"),
       pch=NA,
       lwd=c(2,4),
       col=c("blue","green"),bty="n")
}

}
mtext("Time",side=1,line=1,outer=TRUE,cex=1.1)
mtext("Concentration",side=2,line=0.8,outer=TRUE,las=3,cex=1.1)

dev.off()
cmd <- paste("open",fn)
system(cmd)
```