Exercise 2: Lake Phyto- and Zooplankton Model

ETH Zurich Course 701-0426-00L: Modelling Aquatic Ecosystems (Schuwirth)

March 5, 2025

Goals:

- Understand how a model in process table notation (introduced in section 4.1 of the manuscript) can be implemented in `ecosim`.
- Understand the behaviour of the solutions of the lake phyto- and zooplankton model described in section 11.2 under constant and seasonally varying environmental conditions.

Task 1: Model with constant driving forces

Study the implementation of the model described in section 11.2 for constant driving forces. We expand our model with two organisms, algae and zooplankton. To implement the model in the package `ecosim`, we first need to define the **process table** (Table 11.4) and **process rates** (Table 11.5).

Table 11.4: Process table of a simple lake plankton model.

Process	**Substances**	**Organisms (ALG)**	**Organisms (ZOO)**	**Rate**
	HPO_4^{2-} (gP)	ALG (gDM)	ZOO (gDM)	
Growth of algae	$-\alpha_{\mathrm{P,ALG}}$	1		$\rho_{\mathrm{gro,ALG}}$
Death of algae		-1		$\rho_{\mathrm{death,ALG}}$
Growth of zooplankton		$-\frac{1}{Y_{\mathrm{ZOO}}}$	1	$\rho_{\mathrm{gro,ZOO}}$
Death of zooplankton			-1	$\rho_{\mathrm{death,ZOO}}$

Table 11.5: Process rates of the first version of the simple lake phyto- and zooplankton model.

Rate	**Rate expression**
$\rho_{\mathrm{gro,ALG}}$	$k_{\mathrm{gro,ALG}} \cdot \frac{C_{\mathrm{HPO_4^{2-}}}}{K_{\mathrm{HPO_4^{2-},ALG}}+C_{\mathrm{HPO_4^{2-}}}} \cdot C_{\mathrm{ALG}}$
$\rho_{\mathrm{death,ALG}}$	$k_{\mathrm{death,ALG}} \cdot C_{\mathrm{ALG}}$
$\rho_{\mathrm{gro,ZOO}}$	$k_{\mathrm{gro,ZOO}} \cdot C_{\mathrm{ALG}} \cdot C_{\mathrm{ZOO}}$
$\rho_{\mathrm{death,ZOO}}$	$k_{\mathrm{death,ZOO}} \cdot C_{\mathrm{ZOO}}$

a) First, we load the required packages and define the parameters:

```r
# Model with constant driving forces
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

# load required packages:
if ( !require("deSolve") ) {install.packages("deSolve"); library("deSolve") }
if ( !require("ecosim") ) {install.packages("ecosim"); library("ecosim") }
```

```r
# definition of model parameters:
param    <- list(k.gro.ALG   = 0.5,      # 1/d
                 k.gro.ZOO   = 0.4,      # m3/gDM/d
                 k.death.ALG = 0.1,      # 1/d
                 k.death.ZOO = 0.05,     # 1/d
                 K.HPO4      = 0.002,    # gP/m3
                 Y.ZOO       = 0.2,      # gDM/gDM
                 alpha.P.ALG = 0.003,    # gP/gDM
                 A           = 5e+006,   # m2
                 h.epi       = 5,        # m
                 Q.in        = 5,        # m3/s
                 C.HPO4.in   = 0.04,     # gP/m3
                 C.HPO4.ini  = 0.04,     # gP/m3
                 C.ALG.ini   = 0.1,      # gDM/m3
                 C.ZOO.ini   = 0.1)      # gDM/m3
```

b) Next, based on the process table and process rates, we define the processes growth and death of algae and zooplankton as objects of the class `process` of the package `ecosim`.

- Each process is defined by its name, rate, and stoichiometry.
- The rate is defined as an expression that can use parameters (defined for the object of class `system` below),
- concentrations defined in objects of class `reactor` that are part of the object of class `system`.
- To define the stoichiometry, a named list of expressions must be provided that identifies the substance or organism concentrations as the names and contains the stoichiometric coefficients as expressions.

Now, complete and run the definitions of the transformation processes describing the growth and death of algae and zooplankton.
Hint: look at what we did in Exercise 1.

```r
# definition of transformation processes

# growth of algae:

gro.ALG   <- new(Class  = "process",
                 name   = "Growth of algae",
                 rate   = expression(k.gro.ALG*
                                     C.HPO4/(K.HPO4+C.HPO4)*
                                     C.ALG),
                 stoich = list(C.ALG  = expression(1),               # gDM/gDM
                               C.HPO4 = expression(-alpha.P.ALG)))   # gP/gDM

# death of algae: TO BE COMPLETED

death.ALG <- new(Class = ...,
                 name   = ...,
                 rate   = ...,
                 stoich = ...)            # gDM/gDM

# growth of zooplankton:
```

```
gro.ZOO   <- new(Class  = "process",
                 name   = "Growth of zooplankton",
                 rate   = expression(k.gro.ZOO*
                                     C.ALG*
                                     C.ZOO),
                 stoich = list(C.ZOO  = expression(1),              # gDM/gDM
                               C.ALG  = expression(-1/Y.ZOO)))      # gP/gDM

# death of zooplankton: TO BE COMPLETED

death.ZOO <- new(Class  = ...,
                 name   = ...,
                 rate   = ...,
                 stoich = ...)           # gDM/gDM
```

c) Next, we define the mixed box describing the epilimnion of the lake as an object of the class `reactor`.

Complete and run the definition of the epilimnion. *Hint: look at what we did in Exercise 1.*

```
# definition of reactor to describe the epilimnion of the lake: TO BE COMPLETED

epilimnion <-
  new(Class             = ...,
      name              = ...,
      volume.ini        = ...,
      conc.pervol.ini   = list(C.HPO4 = ...,     # gP/m3
                               C.ALG  = ...,      # gDM/m3
                               C.ZOO  = ...),     # gDM/m3
      inflow            = ...,                   # m3/d
      inflow.conc       = list(C.HPO4 = expression(C.HPO4.in),
                               C.ALG  = 0,
                               C.ZOO  = 0),
      outflow           = ...,
      processes         = list(...))
```

d) Finally, we combine the reactor, the parameters, and the desired output times in an object of class `system`.

```
# definition of the system consisting of a single reactor:

# observe the time frame of the simulation
system.11.2.a <- new(Class    = "system",
                     name     = "Lake",
                     reactors = list(epilimnion),
                     param    = param,
                     t.out    = seq(0,2*365,by=1))
```

Note that this object contains all definitions of the configuration of reactors (in this case just a single one), the processes active in each reactor, the model parameters, and the output time points. Any simulations carried out will refer to the definitions in this object, and not to the external variables that we used to set up the elements of the system.

e) Perform a simulation and store the results by using the function `calcres` of the package `ecosim`:

```
# perform simulation:

res.11.2.a <- calcres(system.11.2.a)

# plot results with default options:

plotres(res.11.2.a)

# variables in a vector 'c()' are plotted in the same graph

plotres(res=res.11.2.a,colnames=c("C.ALG", "C.HPO4","C.ZOO"))

# variables in a list 'list()' are plotted in different graphs

plotres(res=res.11.2.a,colnames=list("C.ALG", "C.HPO4", "C.ZOO"))

# combination of the two

plotres(res=res.11.2.a,colnames=list("C.HPO4",c("C.ALG", "C.ZOO")))

# plot and save as pdf

plotres(res      = res.11.2.a,
        colnames = list("C.HPO4",c("C.ALG", "C.ZOO")),
        file     = "exercise_2_results_a.pdf",
        width    = 10,
        height   = 5)
```

Theory questions:

1. Are the algae concentrations controlled bottom-up (by phosphate limitation) or top-down (by grazing of zooplankton)?
2. What is the reason for oscillating concentrations under constant driving forces?

Task 2: Model with seasonally varying driving forces

Study the implementation of the model extension to seasonally varying driving forces. First, run the model implementation below.

a) We adapt system definitions to introduction of seasonally varying driving forces: Now the process rates look different:

Table 11.6: Process rates of the extended version of the simple lake phyto- and zooplankton model.

Rate	Rate expression
$\rho_{\text{gro,ALG}}$	$k_{\text{gro,ALG,T0}} \cdot \exp\left(\beta_{ALG}(T-T_0)\right) \cdot \frac{1}{\lambda h} \log\left(\frac{K_I+I_0}{K_I+I_0 \exp -\lambda h}\right) \cdot \frac{C_{\text{HPO}_4^{2-}}}{K_{\text{HPO}_4^{2-},\text{ALG}}+C_{\text{HPO}_4^{2-}}} \cdot C_{\text{ALG}}$
$\rho_{\text{death,ALG}}$	$k_{\text{death,ALG}} \cdot C_{\text{ALG}}$
$\rho_{\text{gro,ZOO}}$	$k_{\text{gro,ZOO,T0}} \cdot \exp\left(\beta_{ALG}(T-T_0) \cdot C_{\text{ALG}} \cdot C_{\text{ZOO}}\right.$
$\rho_{\text{death,ZOO}}$	$k_{\text{death,ZOO}} \cdot C_{\text{ZOO}}$

```r
# Model with seasonally varying driving forces
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

# copy the previous system definition:

system.11.2.b <- system.11.2.a

# extend model parameters:

param  <- c(param,
            list(beta.ALG    = 0.046,    # 1/degC
                 beta.ZOO    = 0.08,     # 1/degC
                 T0          = 20,       # degC
                 K.I         = 30,       # W/m2
                 lambda.1    = 0.10,     # 1/m
                 lambda.2    = 0.10,     # m2/gDM
                 t.max       = 230,      # d
                 I0.min      = 25,       # W/m2
                 I0.max      = 225,      # W/m2
                 T.min       = 5,        # degC
                 T.max       = 25))      # degC

# extend growth of algae and zooplankton by considering environmental factors:

gro.ALG.ext <-
   new(Class  = "process",
       name   = "Growth of algae extended",
       rate   = expression(k.gro.ALG*
                           exp(beta.ALG*(T-T0))*
                           C.HPO4/(K.HPO4+C.HPO4)*
                           log((K.I+I0)/
                               (K.I+I0*exp(-(lambda.1+lambda.2*C.ALG)*h.epi)))/
                           ((lambda.1+lambda.2*C.ALG)*h.epi)*
                           C.ALG),
       stoich = list(C.ALG  = 1,                             # gDM/gDM
                     C.HPO4 = expression(-alpha.P.ALG)))     # gP/gDM

gro.ZOO.ext <-
            new(Class  = "process",
                name   = "Growth of zooplankton",
                rate   = expression(k.gro.ZOO*
                                    exp(beta.ZOO*(T-T0))*
                                    C.ALG*
                                    C.ZOO),
                stoich = list(C.ZOO  = expression(1),              # gDM/gDM
                              C.ALG  = expression(-1/Y.ZOO)))      # gP/gDM

# re-define processes in the reactor "epilimnion":

epilimnion@processes <- list(gro.ALG.ext,death.ALG,gro.ZOO.ext,death.ZOO)

# make environmental conditions (light and temperature) time dependent:
```

```
epilimnion@cond <- list(I0 = expression(0.5*(I0.min+I0.max)+
                                        0.5*(I0.max-I0.min)*
                                        cos(2*pi/365.25*(t-t.max))),   # W/m2
                        T  = expression(0.5*(T.min+T.max)+
                                        0.5*(T.max-T.min)*
                                        cos(2*pi/365.25*(t-t.max))))  # degC
```

b) we can also visualize the varying environmental conditions for two years:

```
# plot the environmental conditions

t <-  seq(1,2*365) # for two years
I0 <- numeric(0)
T <- numeric(0)
for(i in 1:length(t))
{
  I0[i] <- eval(epilimnion@cond$I0, envir=c(param, t=t[i]))
  T[i]  <- eval(epilimnion@cond$T, envir=c(param, t=t[i]))
}
par(mfrow=c(1,2),xaxs="i",yaxs="i",mar=c(4.5,4.5,2,1.5)+0.1)
plot(t, I0, type="l")
plot(t, T, type="l")
```

c) We update a parameter values here. You can also notice we can update parameter values in `ecosim` objects by using the `@` operator.

```
# re-define the reactor "epilimnion" in the system definition:

system.11.2.b@reactors <- list(epilimnion)

# increase algal growth rate to compensate for new limitations:

param$k.gro.ALG <- 0.8

# replace parameters in the system definition:

system.11.2.b@param <- param
```

d) Then we perform the simulation and store the results:

```
# redo simulations and plot results:
res.11.2.b <- calcres(system.11.2.b)

plotres(res=res.11.2.b, colnames=list("C.HPO4",c("C.ALG", "C.ZOO")))

plotres(res=res.11.2.b[1:365,], colnames=list("C.HPO4",c("C.ALG", "C.ZOO")))

plotres(res      = res.11.2.b,    # plot to pdf file
        colnames = list("C.HPO4",c("C.ALG","C.ZOO")),
        file     = "exercise_2_results_b.pdf",
        width    = 10,
        height   = 5)
```

e) We compare the two simulations.

```
# comparison of the two simulations:

plotres(res      = list(const=res.11.2.a,dyn=res.11.2.b),
        colnames = list("C.HPO4","C.ALG","C.ZOO"))

plotres(res      = list(const=res.11.2.a,dyn=res.11.2.b),
        colnames = list("C.HPO4","C.ALG","C.ZOO"),
        file     = "exercise_2_results_b.pdf",
        width    = 10,
        height   = 8)
```

Theory question:

3. What is the difference in the oscillations between the simulation with constant and periodic driving forces?
4. What are the main deficits of the model compared to a real lake?

Task 3: Sensitivity analyis for constant driving forces

A good way to increase your understanding of the model is to manually change a parameter value, try to make a prediction how this will change the results, redo the simulation, compare the result with your prediction, and try to understand differences between your prediction and the result.

A more systematic way of doing this is to perform a sensitivity analysis.

The goal for this task is to perform a sensitivity analysis of the model with constant driving forces for to the following parameters

- $C_{in,HPO_4^{2-}}$
- Q_{in}, $k_{gro,ALG}$
- $k_{death,ALG}$
- $k_{gro,ZOO}$,
- $k_{death,ZOO}$
- Y_{ZOO}

Modifying their values by factors of 2 and 1/2 using the function `calcsens` and interpret the results.

Hint: look at the solution of the Task 5 of Exercise 1.

```
# Sensitivity analysis with constant driving forces
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

# calculate results of sensitivity analysis: TO BE COMPLETED
res.11.2.a.sens <- calcsens(...,
                            param.sens=...)

# plot results:
plotres(res      = res.11.2.a.sens,
        colnames = list("C.HPO4","C.ALG", "C.ZOO"))

plotres(res      = res.11.2.a.sens,
```

```
        colnames = list("C.HPO4","C.ALG", "C.ZOO"),
        file     = "exercise_2_results_a_sens.pdf",
        width    = 10,
        height   = 8)
```

Theory question:

5. What is your expectation regarding the response of the model to the change in each parameter, does the result match your expectation and can you explain the observed changes?

Task 4 - Homework: Sensitivity analyis for seasonally varying driving forces

Do a sensitivity analysis for seasonally varying driving forces and discuss the differences to the case with constant driving forces.

```
# TO BE COMPLETED
```