TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

Thomas Petzoldt
Faculty of Environmental Sciences, Institute of Hydrobiology

Simulation of dynamic models in R: Tools for science and engineering, economy and health

Wrocław, 2018-07-04

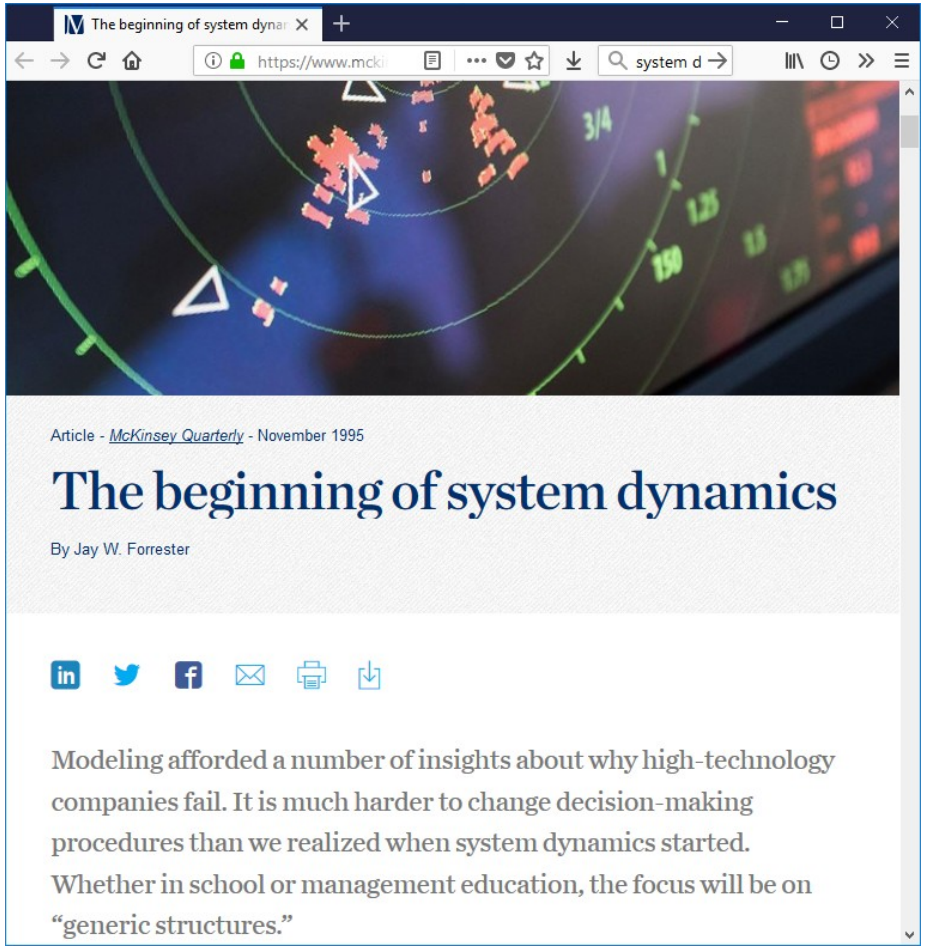
Thomas Petzoldt

- Study: biology (University Rostock, TU Dresden)
- PhD in biology (algae bloom modelling)
- Senior scientist (Faculty of environmental sciences)
- applied statistics, modelling, aquatic ecology

`http://tu-dresden.de/Members/thomas.petzoldt`

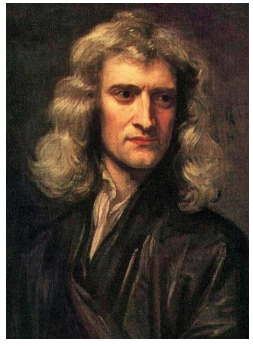
Jay W. Forrester in
McKinsey Quarterly - November 1995

https://www.mckinsey.com/business-functions/strategy-and-corporate-finance/our-insights/the-beginning-of-system-dynamics



"Many believe that system dynamics has helped them become skilled at inventing the future, either by sketching out causal loops on the back of an envelope, or by assembling equations of cause and effect in a computer model. Both approaches work."

Differential calculus



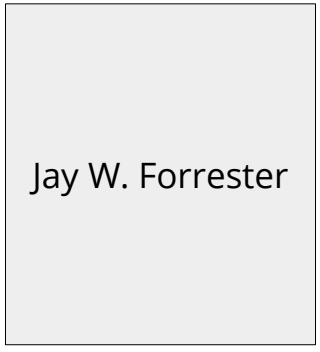
Isaac
Newton
1643 - 1727

Gottfried Wilhelm
Leibniz
1646 - 1716

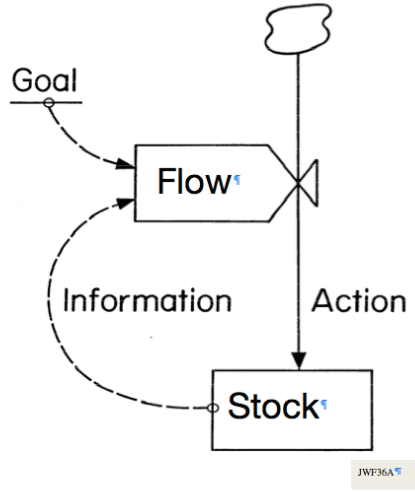
$$\frac{dN}{dt} = r \cdot N$$

Newton, Leibniz: Wikipedia, public domain

System dynamics



Jay W. Forrester
1918 -2016



Jay Forrester: https://www.systemdynamics.de/artikel/in-gedenken-an-jay-w-forrester/ (C)

Forrester (2009) Some basic concepts in system dynamics

Dynamic systems?

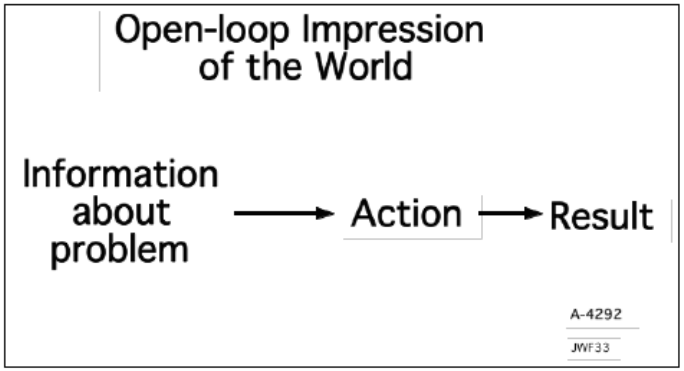


Figure 1

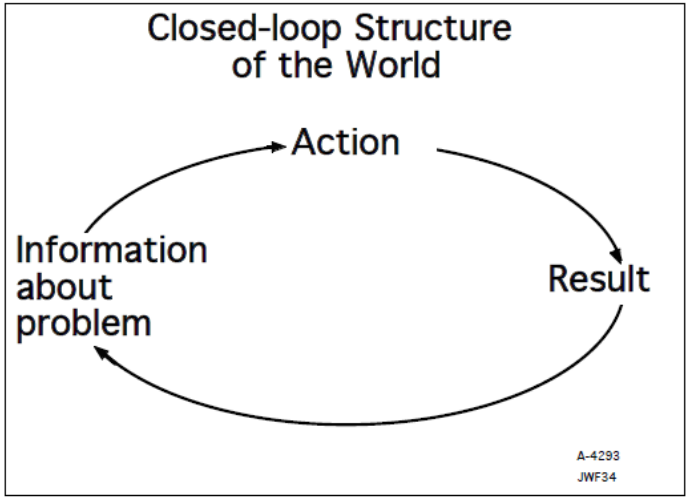


Figure 2

Source:

Forrester, Jay W. (2009) Some Basic Concepts in System Dynamics. Sloan School of Management, Massachusetts Institute of Technology.

Dynamic systems?

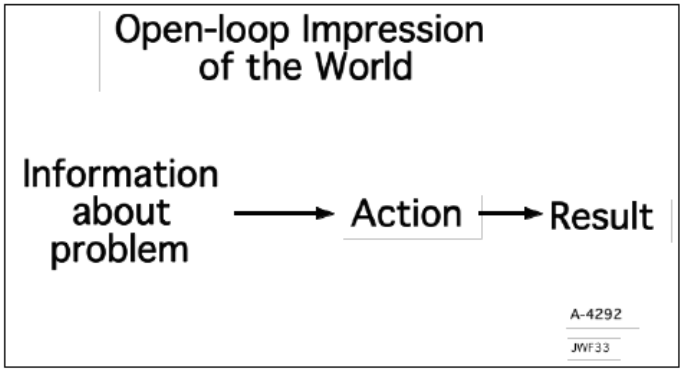


Figure 1

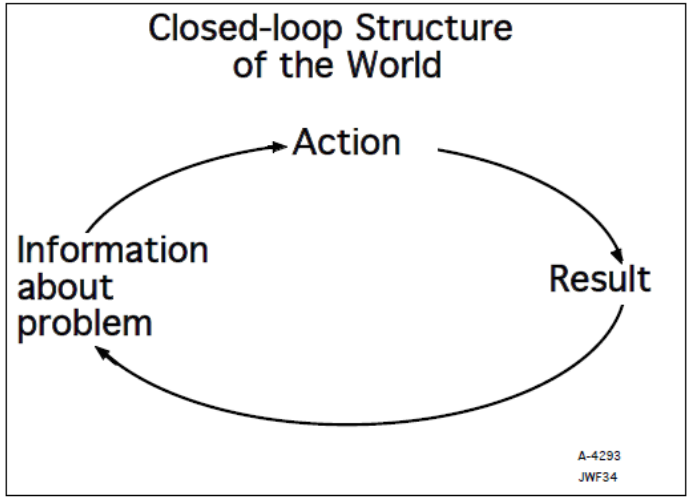


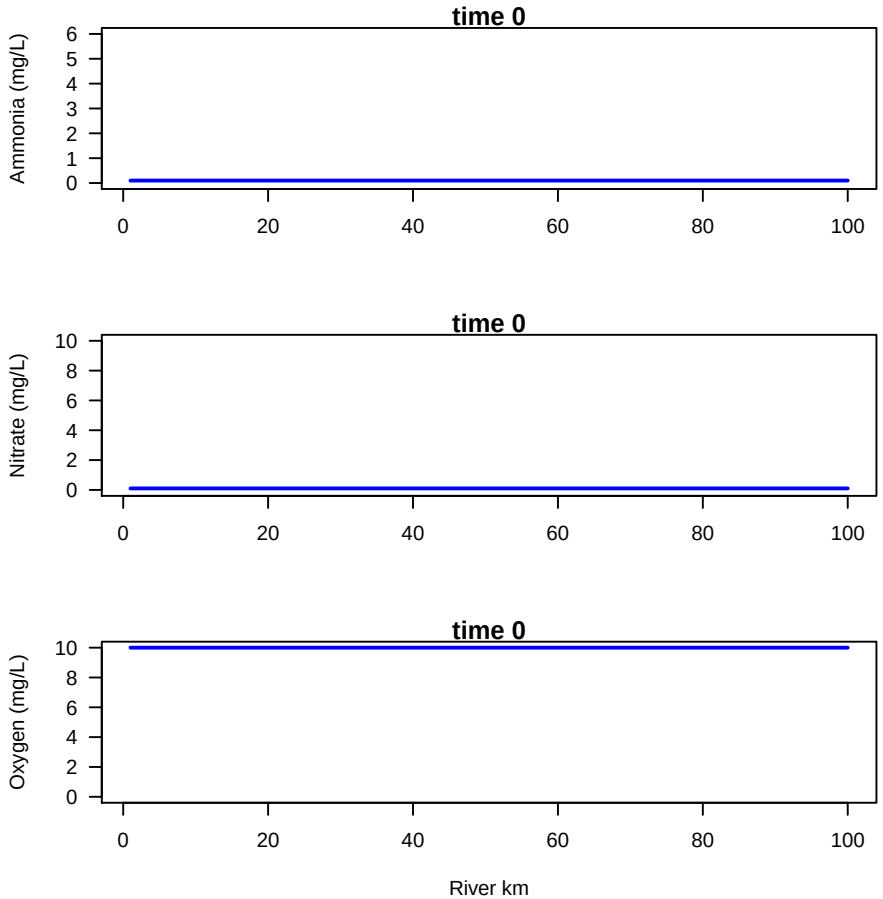
Figure 2

Source:

Forrester, Jay W. (2009) Some Basic Concepts in System Dynamics. Sloan School of Management, Massachusetts Institute of Technology.

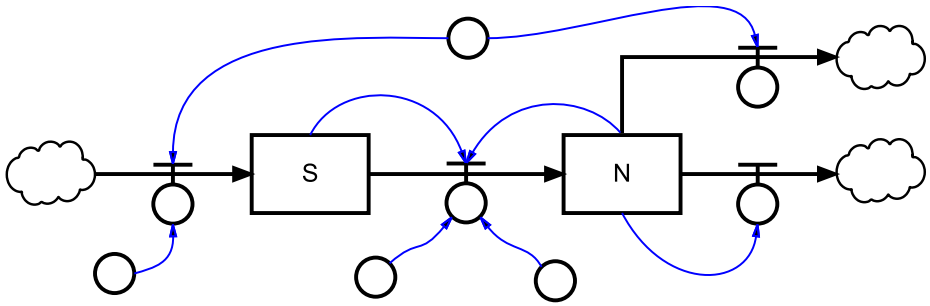
1. time-dependent state, evolves from accumulation of changes
2. existence of feedback loops

Time-dependency + feedback $\Longrightarrow$ dynamics



A pollution wave, traveling down a river.

Dynamic systems everywhere . . .



. . . everywhere

- growth of organisms, cellular pathways, spread of diseases
- mechanics, fluids, turbulence
- chemical reactions, industrial production
- economy, traffic, financial markets
- equilibria, cycles, chaos, (avoidance of) crash

Dynamic systems: how, why and whyR?

Back of an envelope or computer model? (Forrester, 1995)

- Let's speak about computers ...
  - continuous time: differential equations
  - discrete time: another interesting story

Dynamic systems: how, why and whyR?

Back of an envelope or computer model? (Forrester, 1995)

- Let's speak about computers ...
  - continuous time: differential equations
  - discrete time: another interesting story

Why numerical integration?

- Not all systems have an analytical solution.
- Numerical solutions allow discrete input.

Dynamic systems: how, why and whyR?

Back of an envelope or computer model? (Forrester, 1995)

- Let's speak about computers ...
  - continuous time: differential equations
  - discrete time: another interesting story

Why numerical integration?

- Not all systems have an analytical solution.
- Numerical solutions allow discrete input.
- Forrester 2009: "...students can deal with high-order dynamic systems without ever discovering that their elders consider such to be very difficult."

Dynamic systems: how, why and whyR?

Back of an envelope or computer model? (Forrester, 1995)

- Let's speak about computers ...
  - continuous time: differential equations
  - discrete time: another interesting story

Why numerical integration?

- Not all systems have an analytical solution.
- Numerical solutions allow discrete input.
- Forrester 2009: "...students can deal with high-order dynamic systems without ever discovering that their elders consider such to be very difficult."

Why R?

- Seamless integration of dynamic models in the data analysis.

A brief history

2001 odesolve (Setzer) contained two ODE solvers, lsoda, rk4

2003 simecol (Petzoldt) object oriented model structure

2009 deSolve (Soetaert, Petzoldt, Setzer, 2010): complete set of solvers

Dynamical simulations produce plenty of data. That's why I learned R. Around 100 add-on packages existed at that time, one of them odesolve.

I was very excited and planned to use it for a workshop. But two weeks before, I missed an urgently needed feature. I looked in the source code and got in contact with the developer, Woodrow Setzer. Then, R took over and replaced almost all our modeling tools – and our productivity increased.

In 2009, Karline Soetaert's first book appeared. She brought R's dynamic toolbox a big step forward, with a full set of solvers and many practical tools. I got the invitation to join the team.

- Since then, a "dynamic ecosystem" evolved ...

Packages that import, depend on or suggest deSolve

Depend: BacArena, bvpSolve, DAMOCLES, DDD, deBInfer, deTestSet, diffEq, ecolMod, ecosim, embryogrowth, EpiModel, expoTree, FME, GPoM, growthrates, hgm, hisse, insideRODE, neuRosim, nlmeODE, ODEnetwork, ODEsensitivity, pauwels2014, phaseR, primer, ReacTran, scaRabee, simecol, SoilR, TDCor, TESS, TreePar, tseriesChaos

Import: asbio, AssetPricing, bdynsys, bioinactivation, capm, CollocInfer, DAISIE, deGradInfer, diversitree, dMod, DSAIDE, dynatopmodel, EpiDynamics, EvolutionaryGames, evoper, flexsurv, geiger, gpDDE, HomoPolymer, httk, hypergeo, microPop, microsamplingDesign, mizer, mkin, nLTT, PBD, pomp, PSM, RLumModel, rodeo, RPANDA, spdynmod, stagePop, streambugs, TIMP, ZeBook

Suggest: AquaEnv, calibrar, cOde, Data2LD, fda, PBSmodelling, PopED, Sim.DiffProc, smfsb

Source: `https://cran.r-project.org/package=deSolve`

State-of-the-art solvers + extensions

Example problem	Type	In R?
stiff systems	lsode, vode, radau	(1)
algebraic constraints	DAE (diff. algebraic eq.)	(1)
time and space	PDE (partial diff. eq.)	(1,2,3)
time delays	DDE (delay diff. eq)	(1)
time dependent external control	forcing functions	(1)
abrupt changes of states	roots + events	(1)
identify parameters	sensitivity, calibration	(4)

(1) deSolve – (2) rootSolve – (3) ReacTran – (4) FME

Convenience and performance

Unified interface of all solvers

- model definition, time, states, parameters, “hyper-parameters”
- `plot`-, `image`- and `hist`- methods (S3)
- comparison between scenarios and with data

Models can be in R, C or Fortran

- direct communication: solver $\longleftrightarrow$ model (both are DLLs)
- fast: avoids overhead of R callbacks
- flexible: input - output handling in R

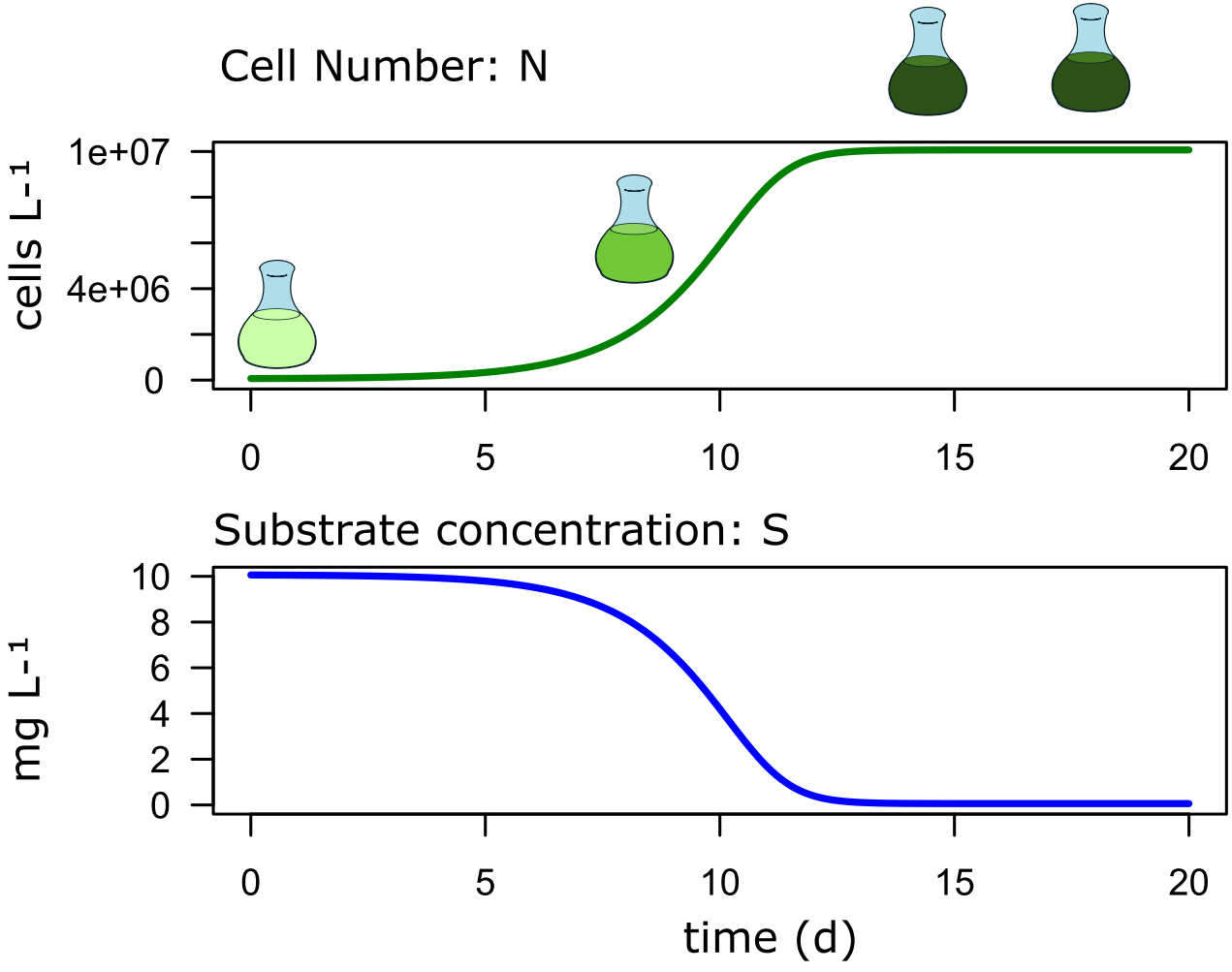
Examples: Laboratory models and field studies

Example 1: Models for laboratory systems

- estimation of physiological parameters (inverse method)
- fitness of populations (seealso: R package growthrates)
- experimental design: sample size to get significant effects
- design of production processes (biomass, pharmaceuticals, bewerages)

- Lab models can be implemented ad-hoc.
- It's so easy!

Substrate dependent growth in a batch



- cells grow until substrate is exhausted.

Substrate limited growth model

Equations

$$f(S) = \frac{r \cdot S}{ks + S}$$

$$\frac{dS}{dt} = -\frac{1}{Y} \cdot f(S) \cdot N$$

$$\frac{dN}{dt} = f(S) \cdot N$$

R Code

```
batch <- function(time, y, parms){
  with(as.list(c(y, parms)), {
    f <- r * S / (ks + S)
    dS <- - 1/Y * f * N
    dN <-         f * N
    return(list(c(dS, dN)))
  })
}
y <- c(S = 10, N = 1e4)
parms <- c(r=1, ks=5, Y=1e6, S0=10)
times <- seq(0, 20, 0.1)
out <- ode(y, times, batch, parms)
plot(out)
```

Forcing functions, events, and root finding

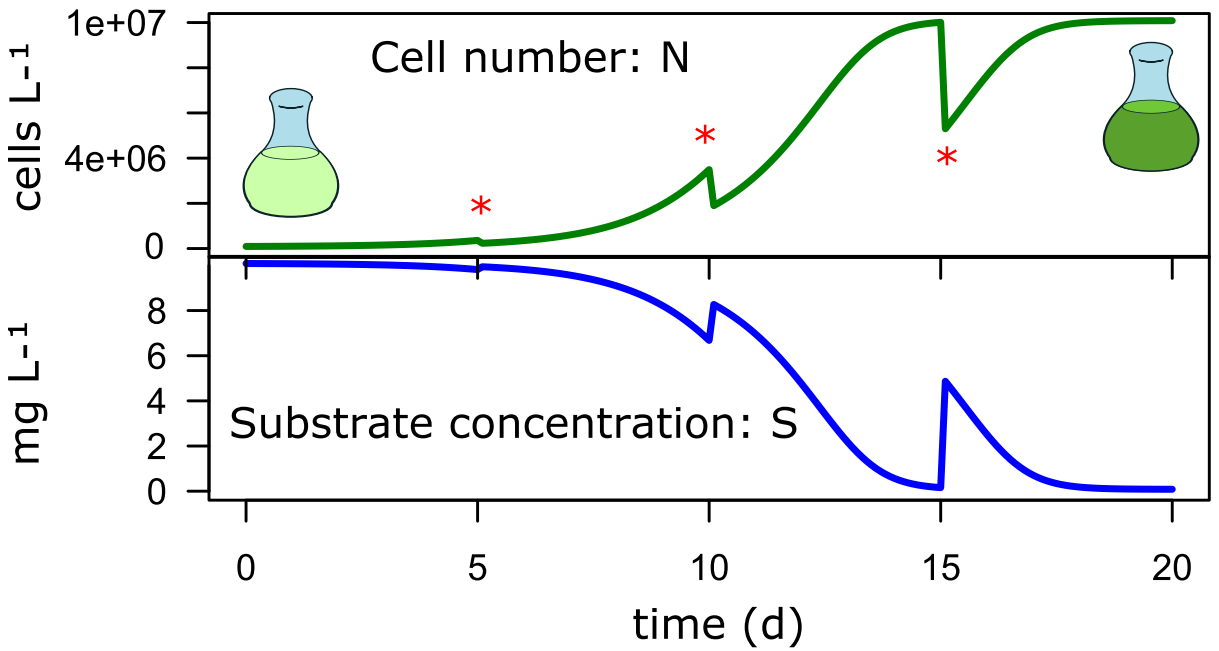
What?

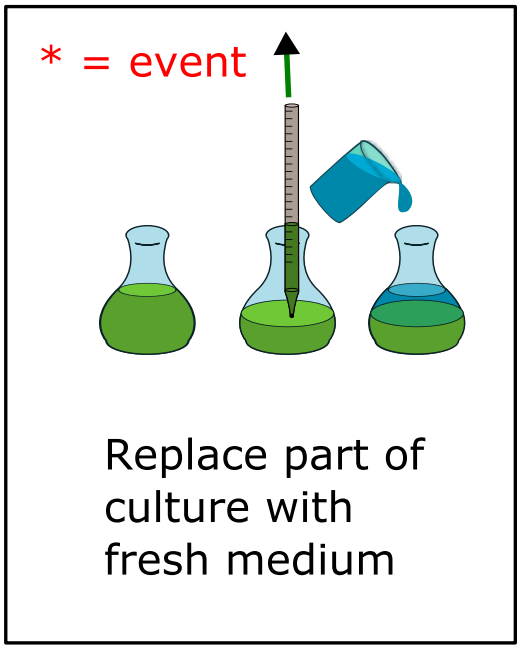
- forcing functions: use of external data.
- event: values of state variables change abruptly.
- root finding: trigger event at certain conditions.

Examples

- semi-continuous cultures
- dosing of chemicals to an experiment
- pharmacokinetics

Events: Semicontinuous culture





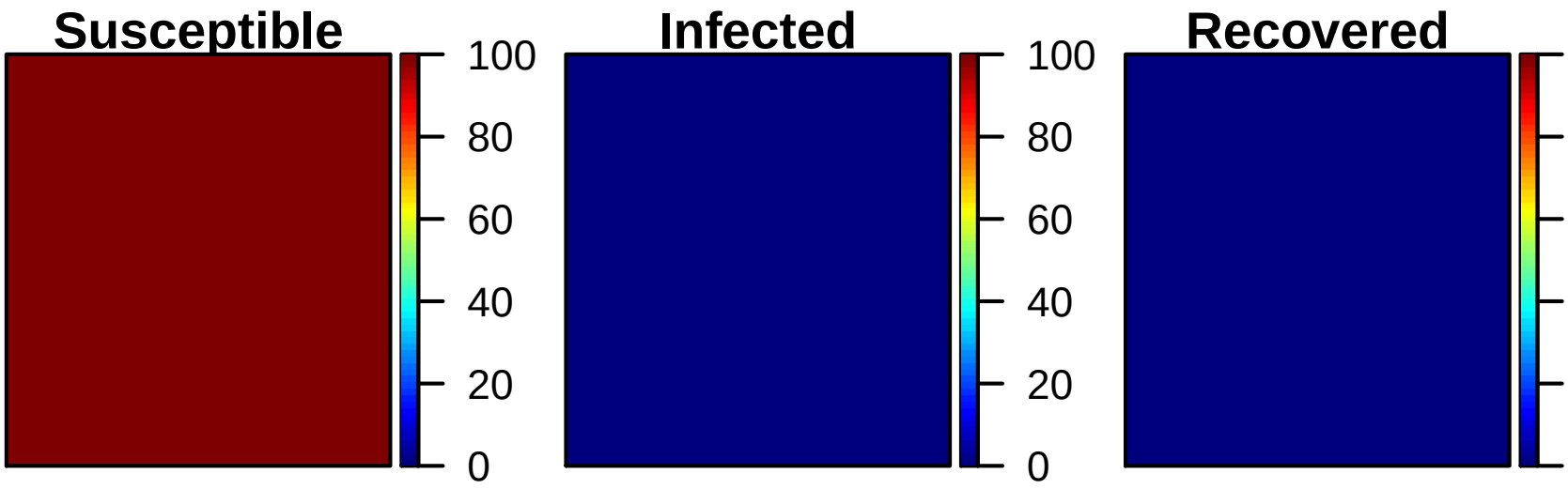
Discontinuous stop-and-go mode

- not trivial in other simulation environments
- R/deSolve supports events directly
- no loops or messy programming required

Partial differential equations (PDE)

- allow systems in 1D, 2D and 3D
- efficient solvers with pre-defined structure of the Jacobian
- suprisingly fast due to R's matrix computations
- core model can be in C or Fortran

An epidemiological SIR model in 2D



- package ReacTran (Soetaert et al. 2012) handles the transport
- 2D grid with 80 x 80 cells (19200 equations)
- about one second for 80 time steps on a standard PC.

Diffusive transport with package ReacTran

```
library("ReacTran")

SIR2D <- function (t, y, parms)  {
  S  <- statematrix(y, N, 1)
  I  <- statematrix(y, N, 2)
  R  <- statematrix(y, N, 3)

  dS <- -beta * I * S
  dI <- beta * I * S - nu * I  + tran.2D(I, dx = dx, dy = dy, D.x = D, D.y = D)$dC
  dR <- nu * I
  list(c(dS, dI, dR))
}


  out <- ode.2D (y = y, func = SIR2D, t = times, nspec=3, parms = NULL,
                 dim = c(N, N), method = "adams")
```

Even better in matrix notation

Model equations

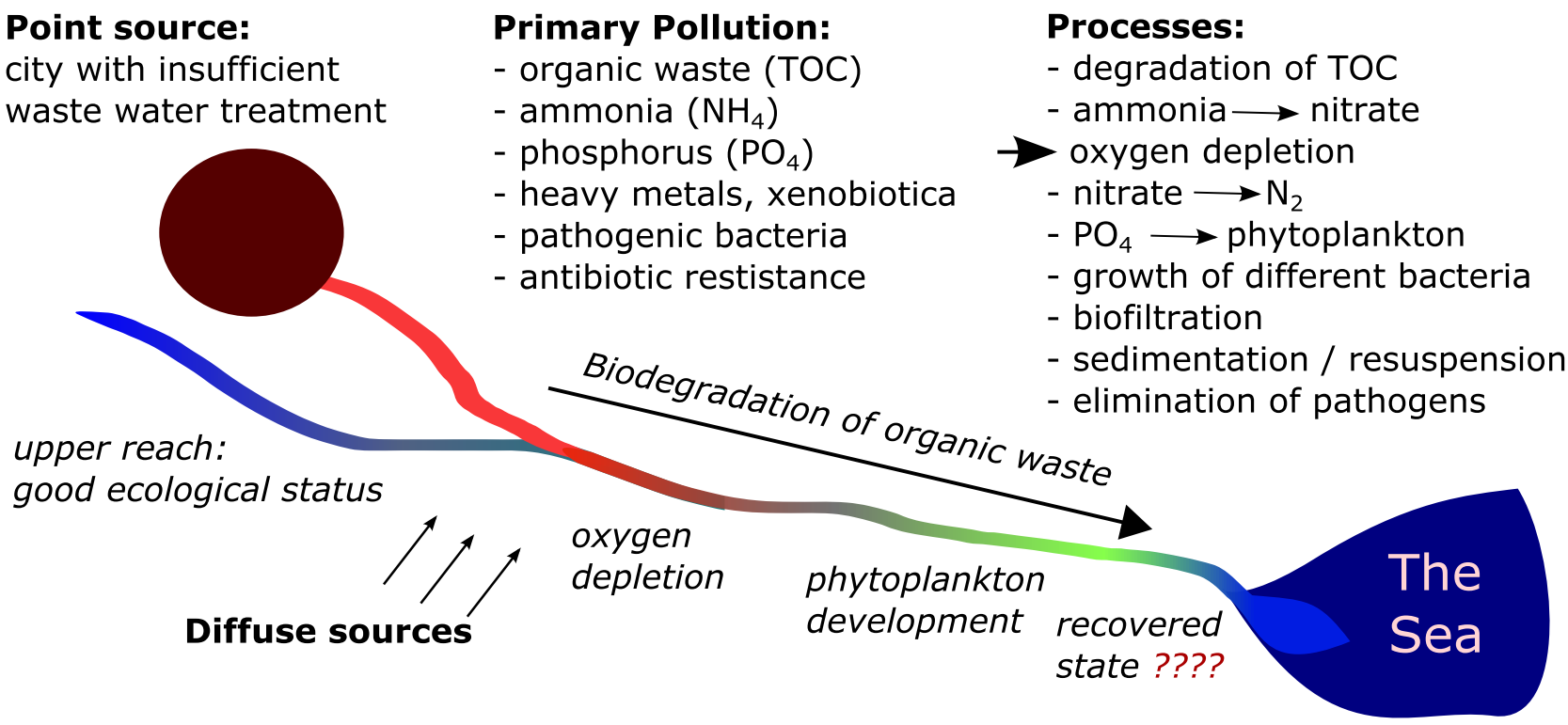
```
SIR2D <- function (t, y, parms)  {
  S  <- statematrix(y, N, 1)
  I  <- statematrix(y, N, 2)
  R  <- statematrix(y, N, 3)

  proc <- matrix(c(
    beta * I * S,                                    # infect
    nu * I,                                          # recover
    tran.2D(I, dx = dx, dy = dy, D.x = D, D.y = D)$dC # transport
  ), ncol=3)

  dy <-  proc %*% interact                           # state equation

  list(dy)
}
```

Interaction matrix

```
interact <- matrix(c(
  -1,  1, 0,  # infect
   0, -1, 1,  # recover
   0,  1, 0   # transport
), ncol=3, byrow=TRUE)
```

Example 2: Matter turnover in a polluted river

Matter turnover in a polluted river



Let's consider 3 variables and 2 processes:

- ammonia + oxygen ⟶ nitrate
  - ▶ oxygen consumption by biological ammonia oxidation (nitrification)
  - ▶ oxygen exchange between atmosphere and water (re-aeration)

The reactive transport equation

$$\text{change} = \text{transport} + \text{processes} \cdot \text{stoichiometry}$$

$$Y' = T + P \cdot V$$

$$\begin{pmatrix} y'_{1,1} & \cdots & y'_{1,n} \\ y'_{2,1} & \cdots & y'_{2,n} \\ & \cdots\cdots & \\ y'_{x,1} & \cdots & y'_{x,n} \end{pmatrix} = \begin{pmatrix} t_{1,1} & \cdots & t_{1,n} \\ t_{2,1} & \cdots & t_{2,n} \\ & \cdots\cdots & \\ t_{x,1} & \cdots & t_{x,n} \end{pmatrix} + \begin{pmatrix} p_{1,1} & \cdots & p_{1,k} \\ p_{2,1} & \cdots & p_{2,k} \\ & \cdots\cdots & \\ p_{x,1} & \cdots & p_{x,k} \end{pmatrix} \cdot \begin{pmatrix} v_{1,1} & \cdots & v_{1,n} \\ v_{2,1} & \cdots & v_{2,n} \\ & \cdots\cdots & \\ v_{k,1} & \cdots & v_{k,n} \end{pmatrix}$$

with:

n = number of state variables (e.g. chemical species)
k = number of processes
x = space coordinate (here: river kilometers in 1D)

Ammonia, Nitrate, Oxygen

Transport (package ReacTran)

```
tran <- cbind(
  tran.1D(C = NH4, D = D, v = v, C.up = NH4up, C.down = NH4dwn, A = A, dx = Grid)$dC,
  tran.1D(C = NO3, D = D, v = v, C.up = NO3up, C.down = NO3dwn, A = A, dx = Grid)$dC,
  tran.1D(C = O2 , D = D, v = v, C.up = O2up,  C.down = O2dwn,  A = A, dx = Grid)$dC
)
```

Stoichiometry matrix

```
stoich <- matrix(c(
  # NH4  NO3  O2
    0,   0,   1,     # reaeration
   -1,  +1,  -4.57   # nitrification
  ),  nrow = 2, byrow = TRUE)
```

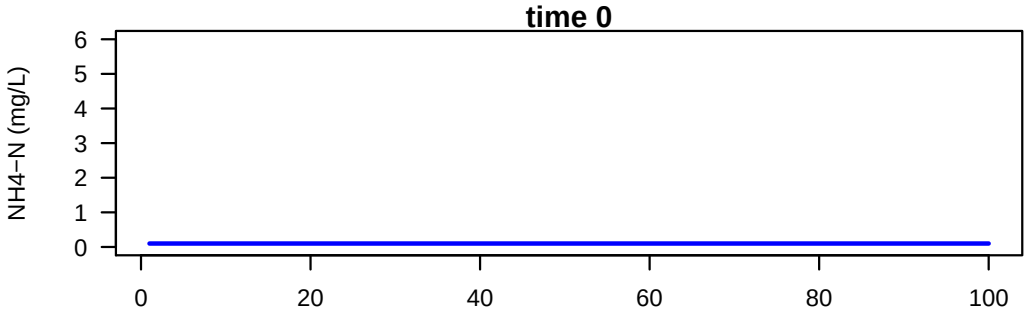
Process equations

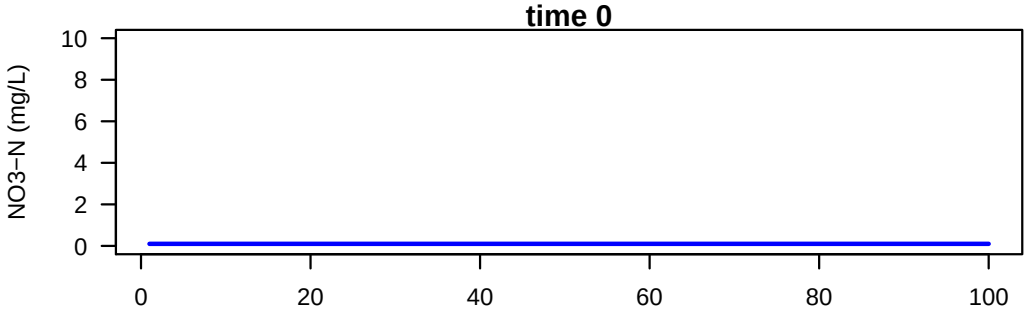
```
proc <- cbind(
  k2 * (O2sat - O2),              # re-aeration
  rMax * O2/(O2 + kO2) * NH4      # nitrification
)
```

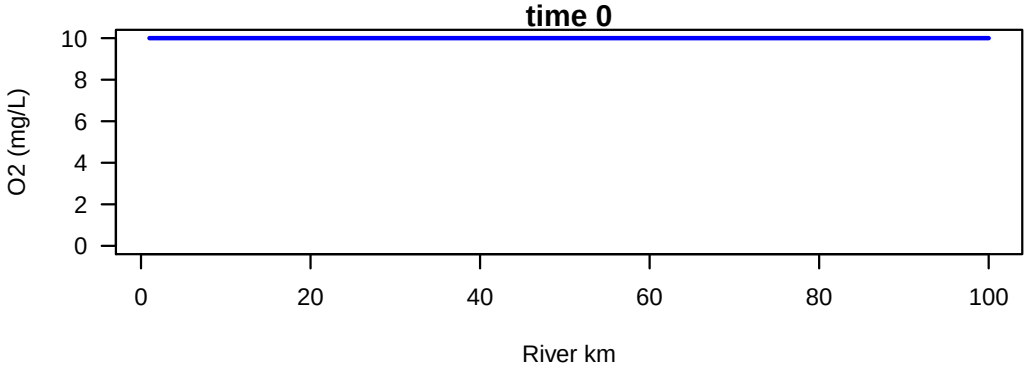
State equation

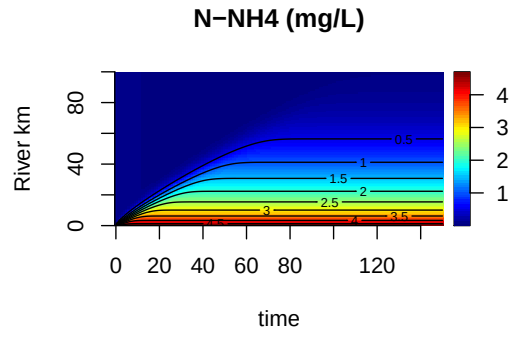
```
dY   <- tran + proc %*% stoich
```

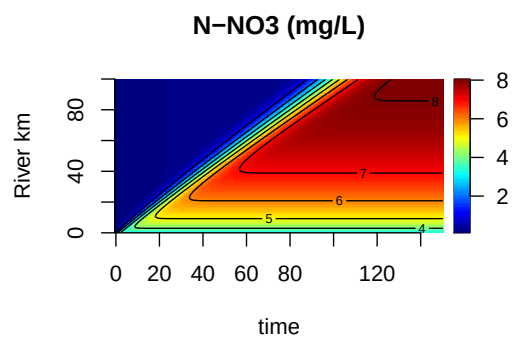
A pollution wave travelling down the river

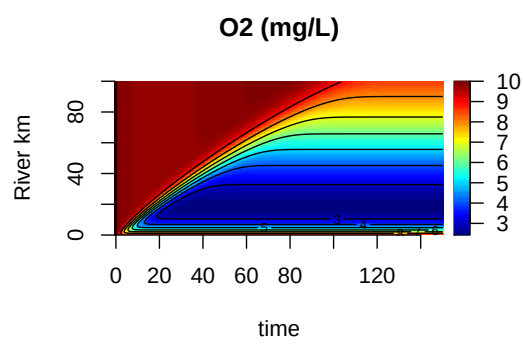












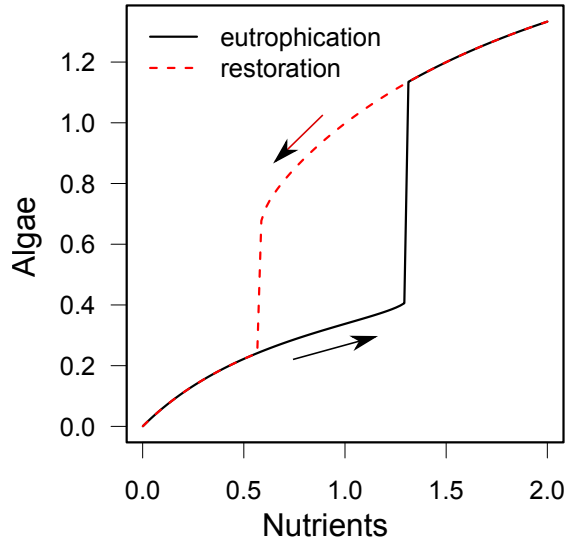
Example 3: Alternative stable states in lakes

Alternative stable states in shallow lakes

Turbid state: many microscopic algae – no macroscopic water plants

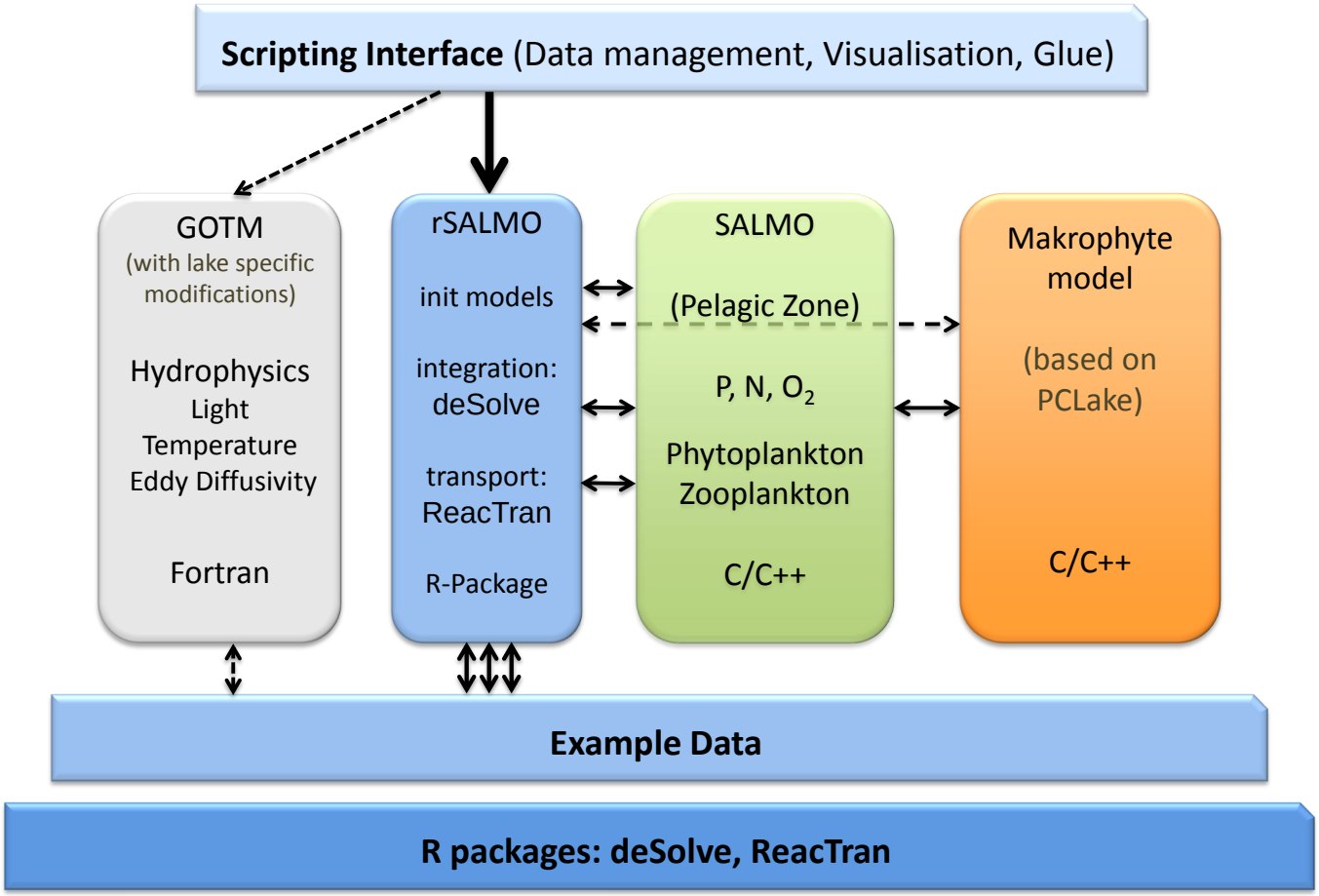
Clear water state: water plants inhibit algae

- Algae (A) and macrophytic vegetation (V) depend on nutrients N and initial state A_0
- Simple theoretical model (Scheffer et al. 1990, 1993):



- Well known only for shallow lakes.
- More complex model needed for deep lakes.

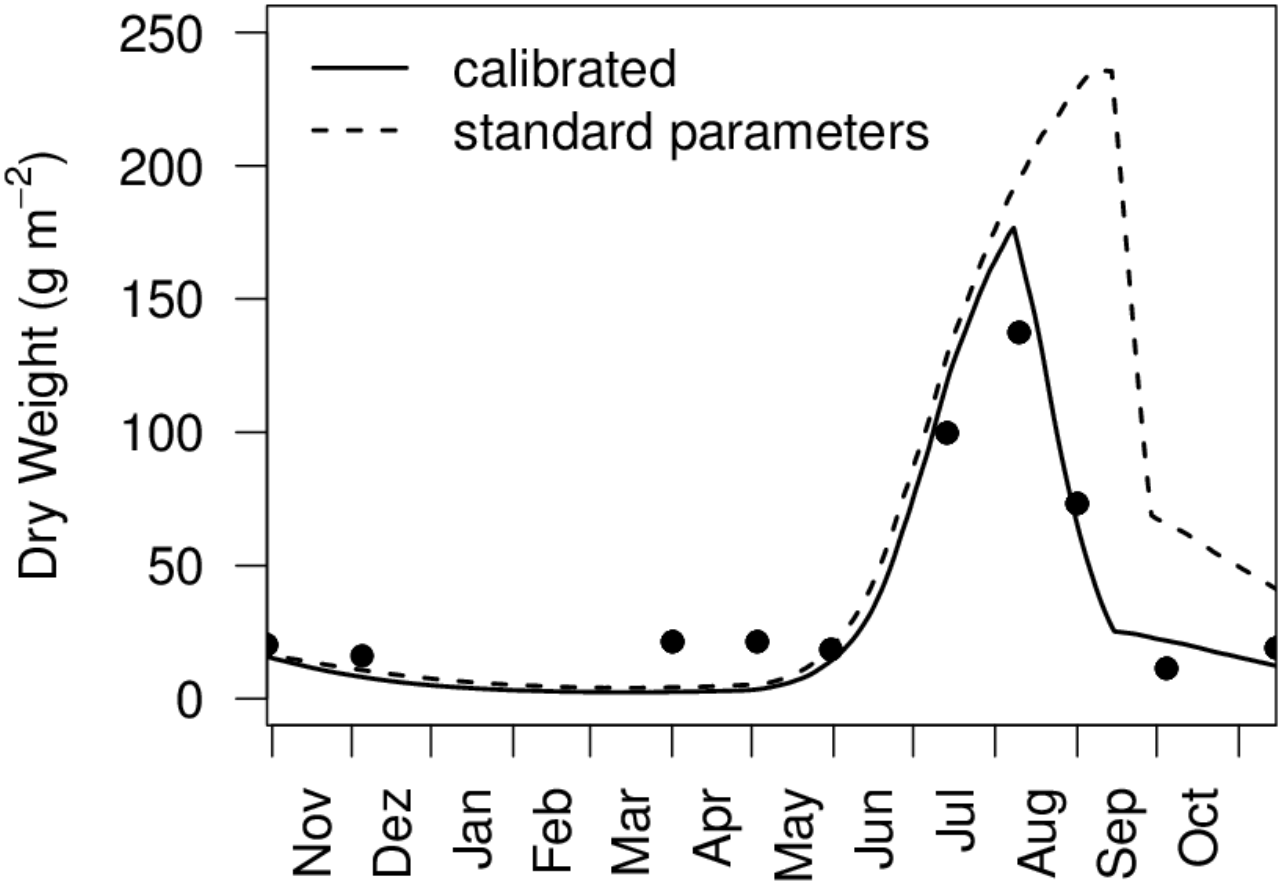
Model coupling



Challenge: Coupling of complete models is a mess, consumes time and human resources.

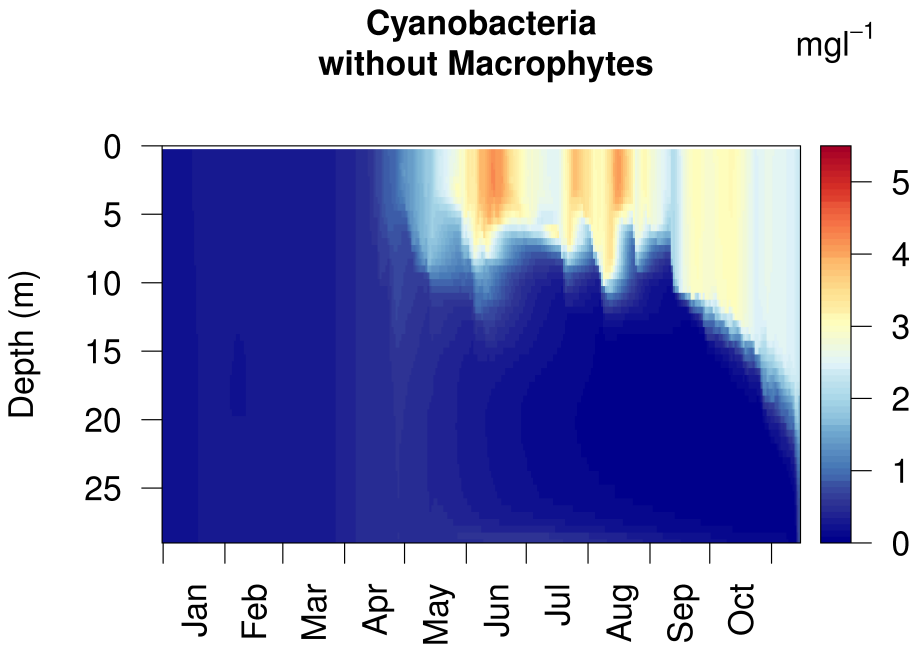
Approach: Strip models down to the processes, close to mathematical notation.

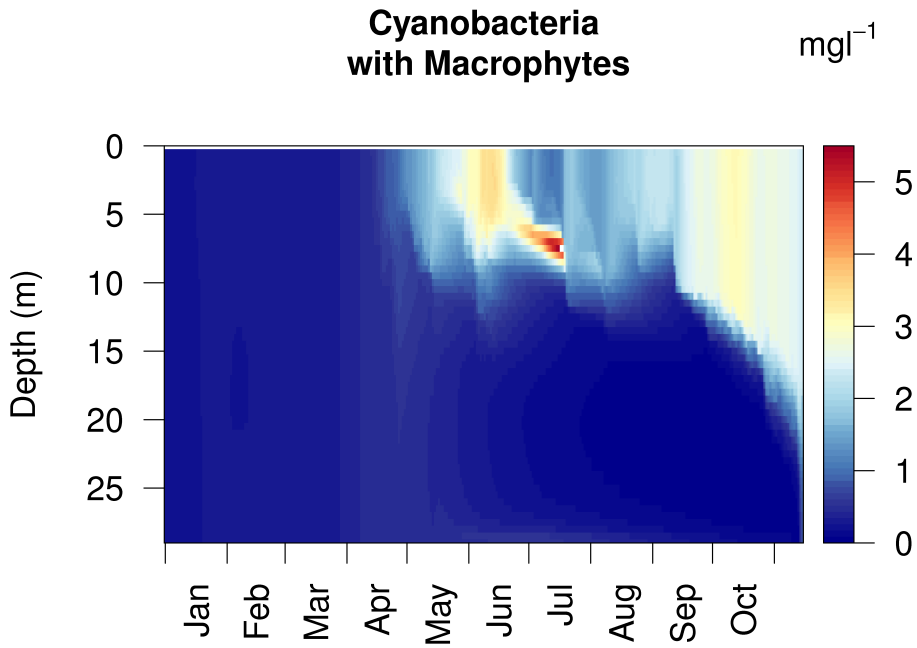
Verification of water plant coverage



Details in Sachse, Petzoldt, Blumstock, Moreira, Pätzig, Rücker, Janse, Mooij and Hilt (2014) http://dx.doi.org/10.1016/j.envsoft.2014.05.023

Less blue-green algae if water plants are present





Details in Sachse, Petzoldt, Blumstock, Moreira, Pätzig, Rücker, Janse, Mooij and Hilt (2014) http://dx.doi.org/10.1016/j.envsoft.2014.05.023

Results

Scientific

- water plants can improve clarity of deep lakes:
  - especially at low and medium nutrient loads
  - strongest impact in summer

Technical

- rapid prototyping of research tools
  - data handling and graphics in R
  - lightweight core models, equations only
  - re-usable numerical packages (deSolve, ReacTran)

But: Model coupling is not student-friendly

- increasing complexity
- need for high-performance
- monolithic codes are hard to extend
- consistency of code and interfaces
- effort of re-implementations
- difficulties in collaborative development

Approach: R ODE Objects (rodeo)

(1) **Table-based notation** of equations

- stoichiometry matrix approach
- unified interface, less redundancies

(2) **Automatic code generation**

- consistency checks
- compiled Fortran code
- portable, easier to extend

Kneis, D., Petzoldt, T., & Berendonk, T. U. (2017) An R-package to boost fitness and life expectancy of environmental models. Environmental Modelling & Software, 96, 123-127. https://doi.org/10.1016/j.envsoft.2017.06.036

Developer

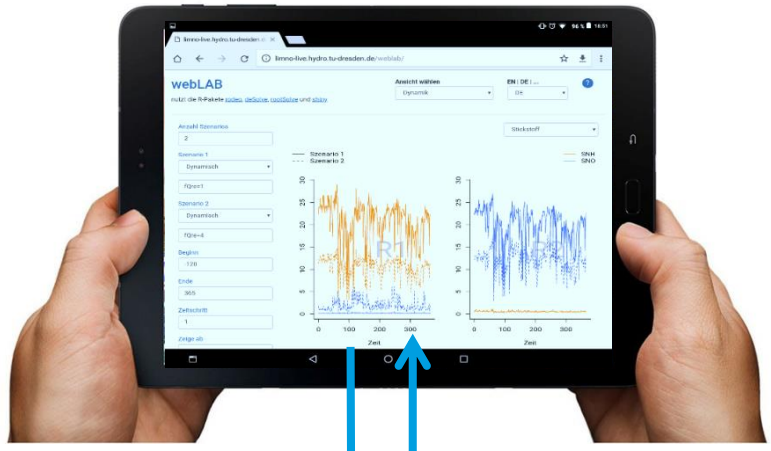
Generic model description (Excel Tables)

Code-Generator

rodeo

docs

User



Linux cloud server

shiny

Numerical solver

Web server, JavaScript-Widgets

deSolve

Fortran

efficient code

docs

Example 4: Nitrogen emissions into lakes

- Is it effective, or do blue-green algae develop?
- Equations adapted from published lake model (BELAMO, ETH Zürich)

Re-implementation: Johannes Feldbauer and David Kneis, Data from BTU Cottbus

Live presentation

`http://limno-live.hydro.tu-dresden.de/models/shallowlake`

Summary and Conclusion

Dynamic models in R:

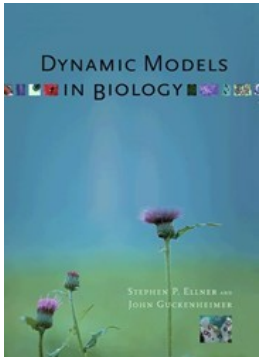
- **efficient algorithms:** basic and advanced applications
- **packages for** model implementation, calibration, visualisation
- **documentation:** manuals, publications, tutorials, books
- **open:** source code + examples

My personal experience:

- fast
- flexible
- highly productive

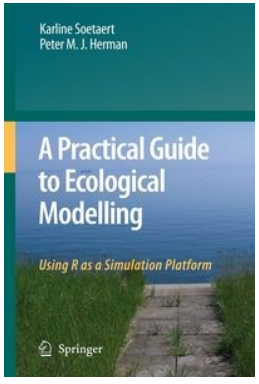
Further reading

2006

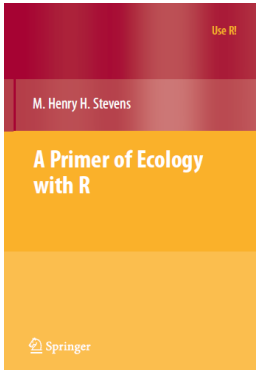


Elner & Guckenheimer

2009

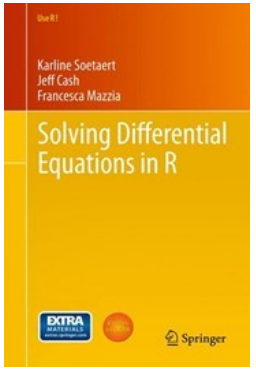


Soetaert & Herman



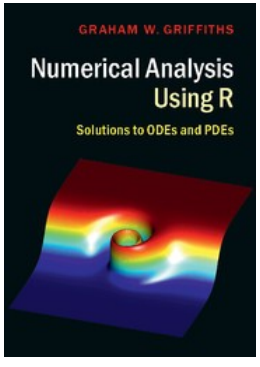
Stevens

2012

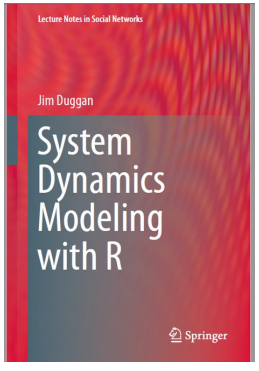


Soetaert, Cash & Mazzia

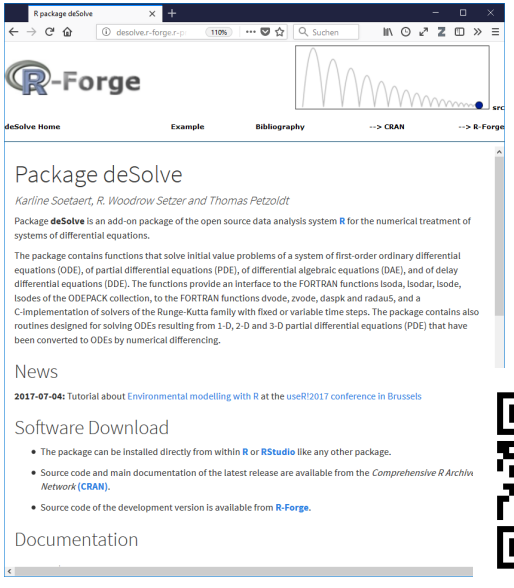
2016



Griffiths



Duggan



Mailing list: r-sig-dynamic-models@r-project.org

Special interest group for dynamic simulation models in R.

http://desolve.r-forge.r-project.org/

Many thanks …

… to Karline Soetaert, Woodrow Setzer, Johannes Feldbauer, David Kneis and Thomas Berendonk

… to our students who tested our software

… to all the people on the internet who gave us feedback

… and to **you** for listening.

Parts of this study were supported by the German Federal Ministry of Education and Research, Grant no. 02WM1028, 0033W015EN, 02WU1351A, 033L041A, 033L041E, 01LR0803D, 01LR0803G

(IWAS, Nitrolimit and INKA-BB)

NITROLIMIT

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

FONA Nachhaltiges Wassermanagement BMBF