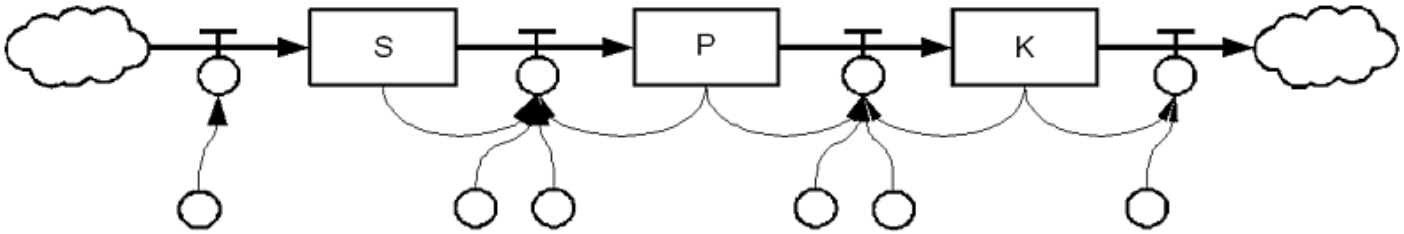
TECHNISCHE UNIVERSITÄT DRESDEN
Faculty of Forest-, Geo- and Hydrosciences, Institute of Hydrobiology

Dynamic simulation models – is R powerful enough?



Thomas.Petzoldt@TU-Dresden.de

useR!

Dynamic models …

$$\frac{dN}{dt} = r \cdot N \cdot \left(1 - \frac{N}{K}\right)$$

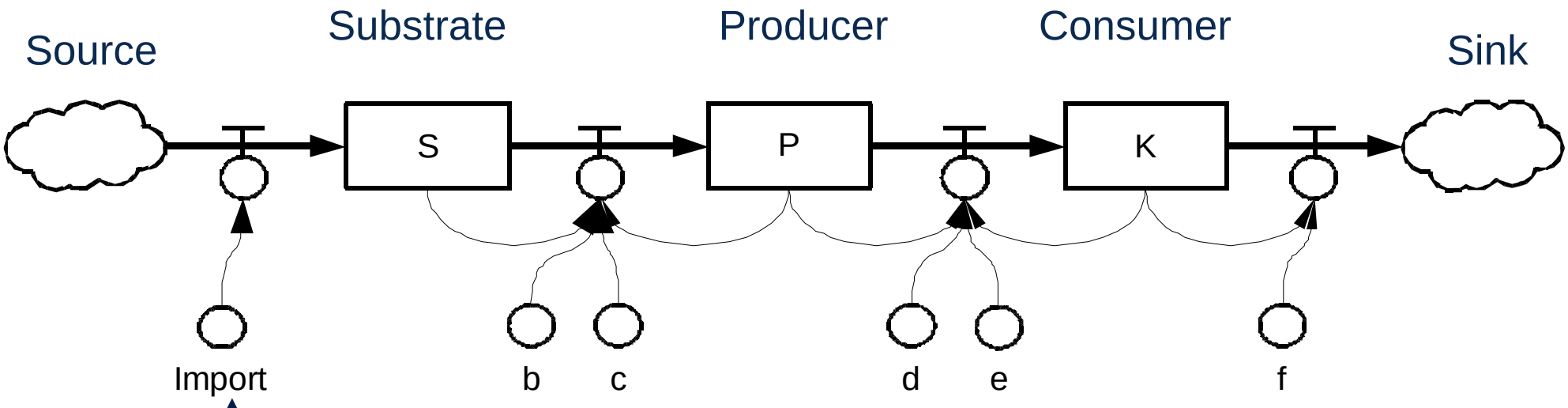
- … models that respect **time explicitly**.

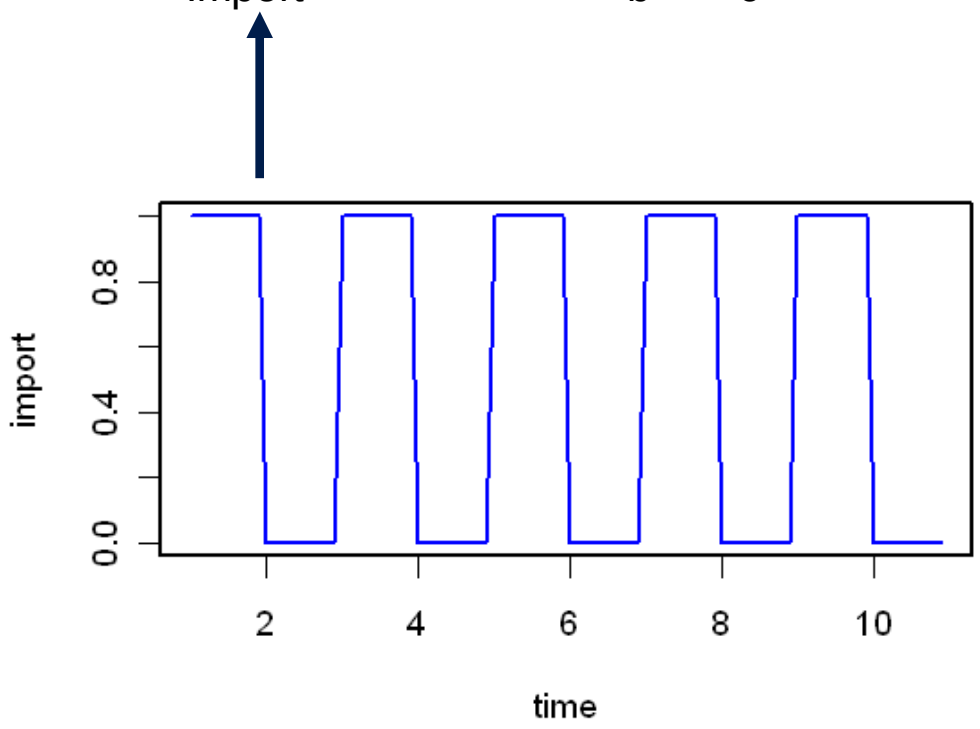
- used in many fields:
  mathematics, physics, chemistry, biology,
  ecology, engineering, economics…

- "What makes using **system dynamics** different from other approaches to studying complex systems is the use of **feedback loops** and **stocks** and **flows**. These elements help describe how even seemingly simple systems display baffling **nonlinearity**.

http://en.wikipedia.org/wiki/System_dynamics

Example 1: A Lotka-Volterra-type model





$$\frac{dS}{dt} = import - b \cdot S \cdot P$$

$$\frac{dP}{dt} = c \cdot S \cdot P - d \cdot P \cdot K$$

$$\frac{dK}{dt} = e \cdot P \cdot K - f \cdot K$$

The LV-type model in R

```
lvmodel <- function(t, x, parms, input)  {
  with(as.list(c(parms, x)), {
    import <- input(t)
    dS <- import - b*S*P
    dP <- c*S*P  - d*K*P
    dK <- e*P*K  - f*K
    list(c(dS, dP, dK))
  })
}

# parameters, initial values time steps …
# … and some data, e.g. rectangular signal

sigimp <- approxfun(signal$times, signal$import)

out <- lsoda(init, times, lvmodel, parms, input=sigimp)
```

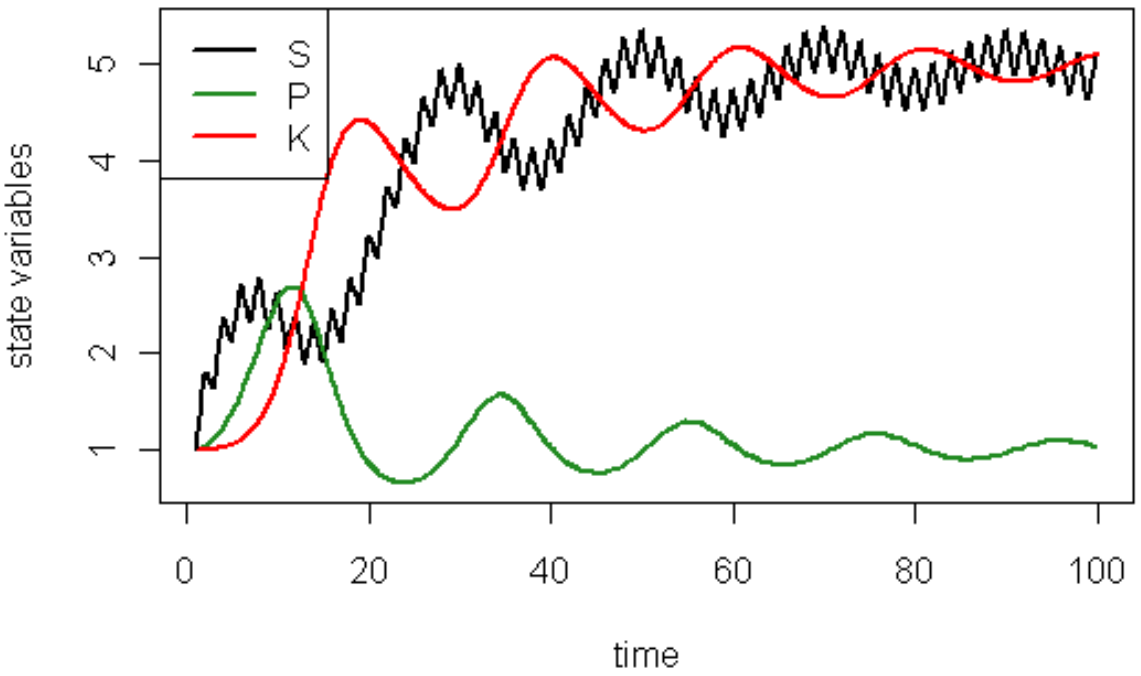
package deSolve (Soetaert, Petzoldt, Setzer)

Benchmark

$$\frac{dS}{dt} = import - b \cdot S \cdot P$$

$$\frac{dP}{dt} = c \cdot S \cdot P - d \cdot P \cdot K$$

$$\frac{dK}{dt} = e \cdot P \cdot K - f \cdot K$$



3 equations (ODEs) in R, 1000 (external) timesteps **CPU time**

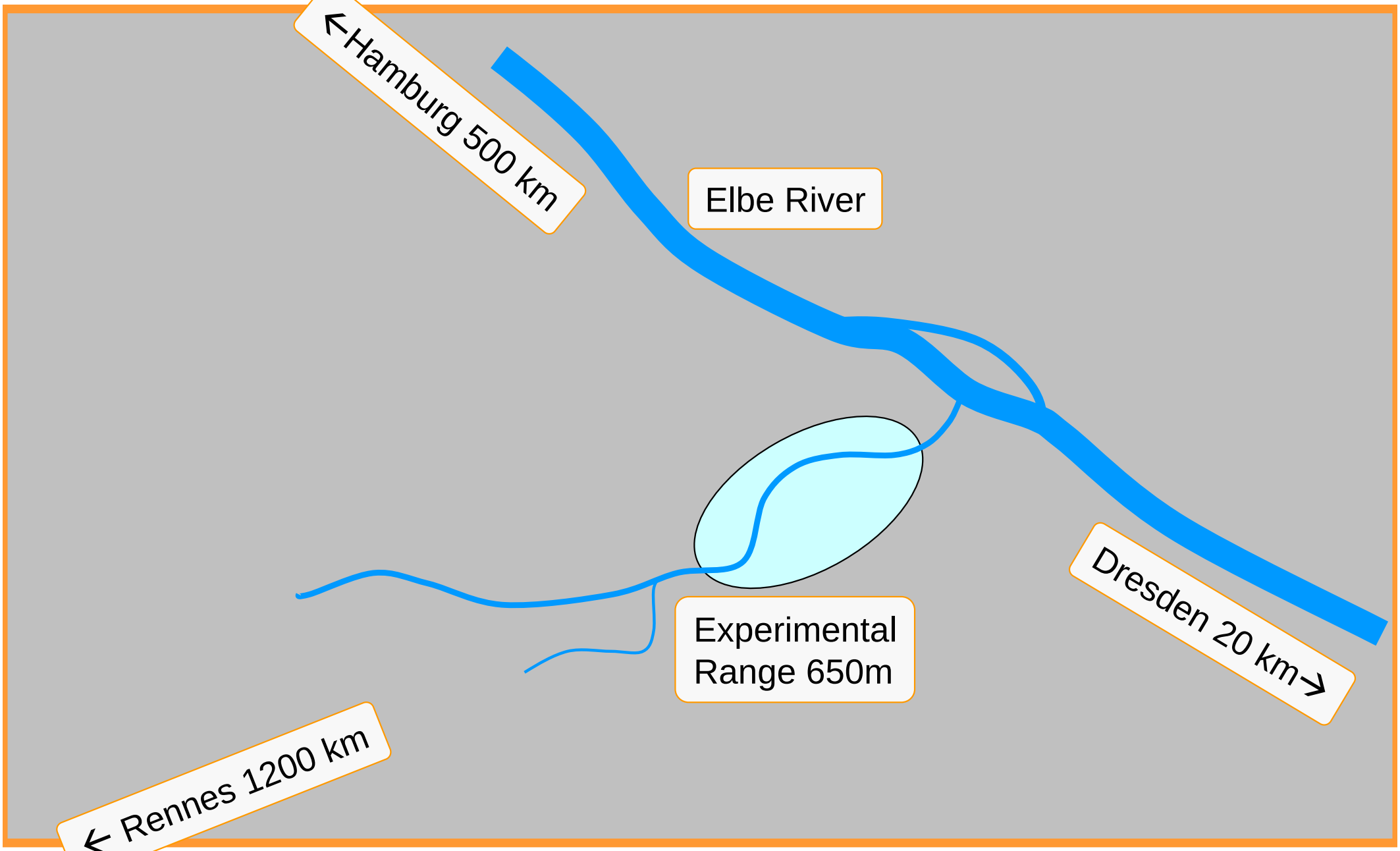
- **case A: interpolated input (approxfun)** **…… 1.4 s**
- **case B: import <- if (trunc(t) %% 2 == 0) 0 else 0.1** **…… 0.8 s**

Time is ok for this toy model, but is R suited for more complex simulations?

3 ODEs = 1 s ➔ spatial system with 10.000 ODEs > 1 hour?

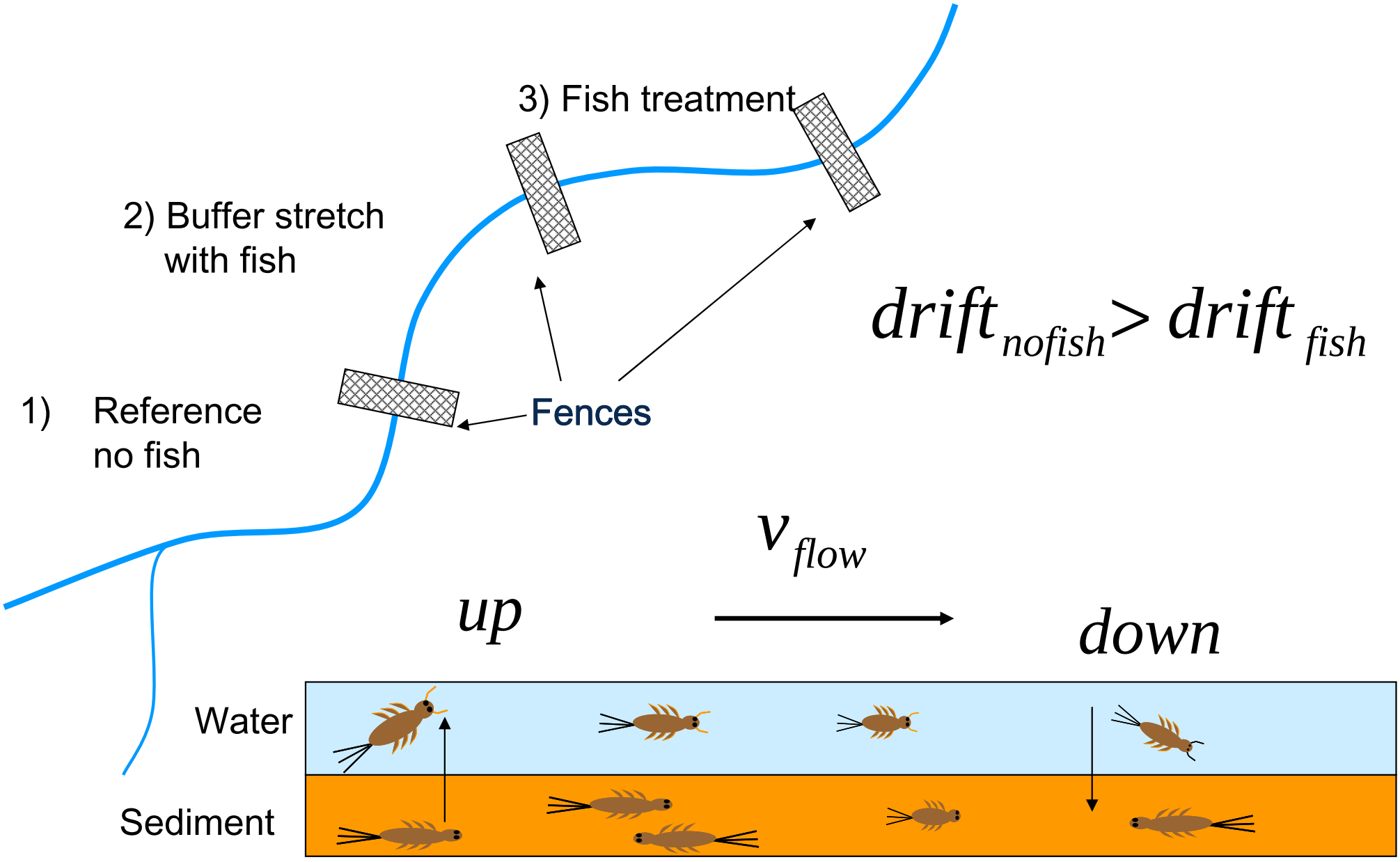
Example 2: A stream model

Experimentally manipulated small stream of our limnological workgroup



Downward drift of water insects in the stream (Mayflies)

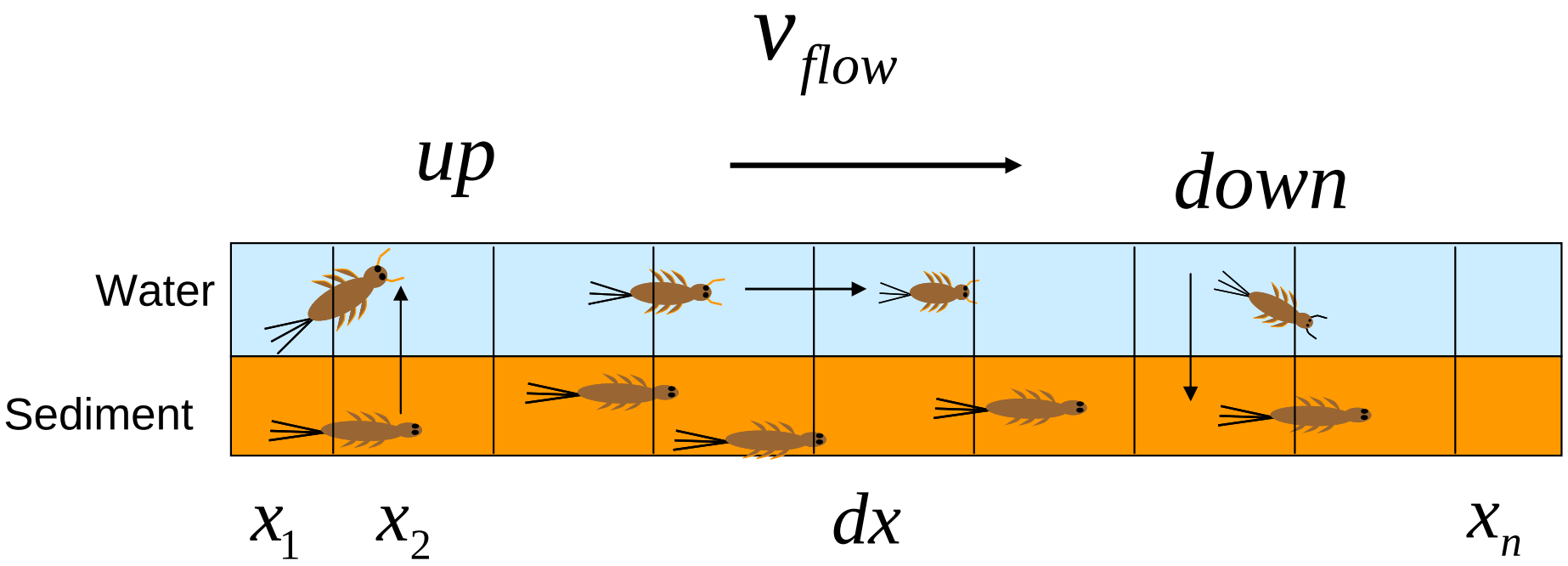
Is buffer stretch sufficiently long?



Can be described by a basic PDE model

Mobile Organisms $$\frac{\partial M}{\partial t} = up \cdot S - down \cdot M - v \cdot \frac{\partial M}{\partial x}$$

Sessile Organisms $$\frac{dS}{dt} = -up \cdot S + down \cdot M$$



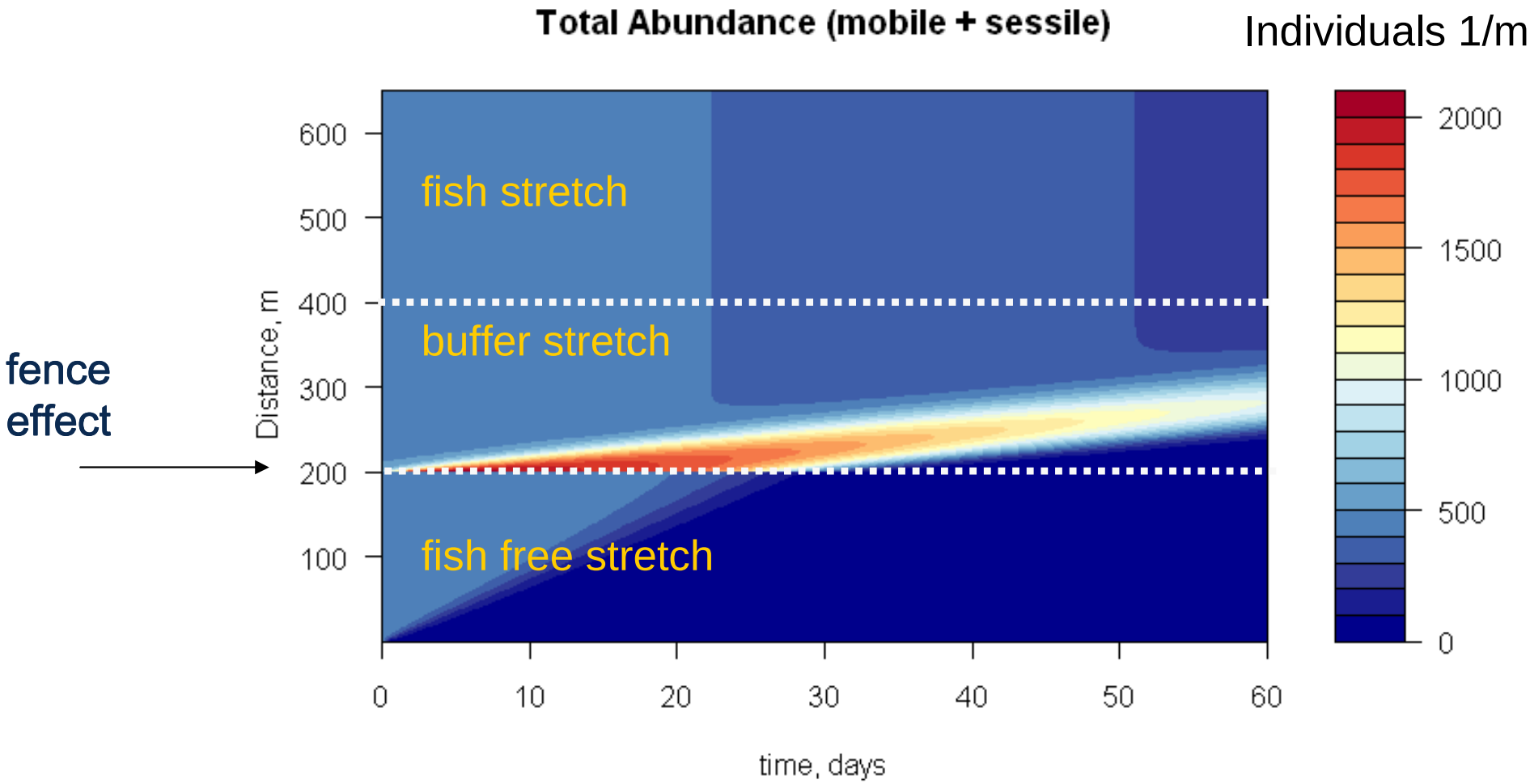
The drift model in R: simple structure but 1300 eqs.

```
drift <- function(time, state, parms) {
  S <- state[1:N]                              # sessile
  M <- state[(N+1):(2*N)]                      # mobile
  dM <- -v * diff(c(0, M))/dx  - down * M  + up * S
  dS <                         + down * M  - up * S - mort * S
  list(c(dS, dM))
}

dx   <- 1                                      # grid size [m]
v    <- 10000                                  # velocity, m/day
x    <- seq(dx/2, 650, by = dx)                # experimental strech 650 m
N    <- length(x)
up   <- c(rep(6.1, 200), rep(1.4, N-200)) # drift rate, 1/d
down <- 8000                                   # settlement rate, 1/d
mort <- 1e-2                                   # mortality rate 1/d

## initial conditions (abundance of mayfly larvae)
state <- c(S=rep(500, N), M=rep(0, N))
times <- seq(0, 60, length=101)                # two months

out   <- ode.1D(y = state, times, drift, parms = NULL, nspec = 2)
```



- 2 simulated months, 100 external time steps
- model with 2 x 650 = 1300 equations; computation time of ode.1D: ……. **0.6s**

  AMD Athlon AM2 X2 6000+, 3000 MHz, 2MB RAM, Windows XP, R-2.10-devel

➔ **buffer stretch long enough**

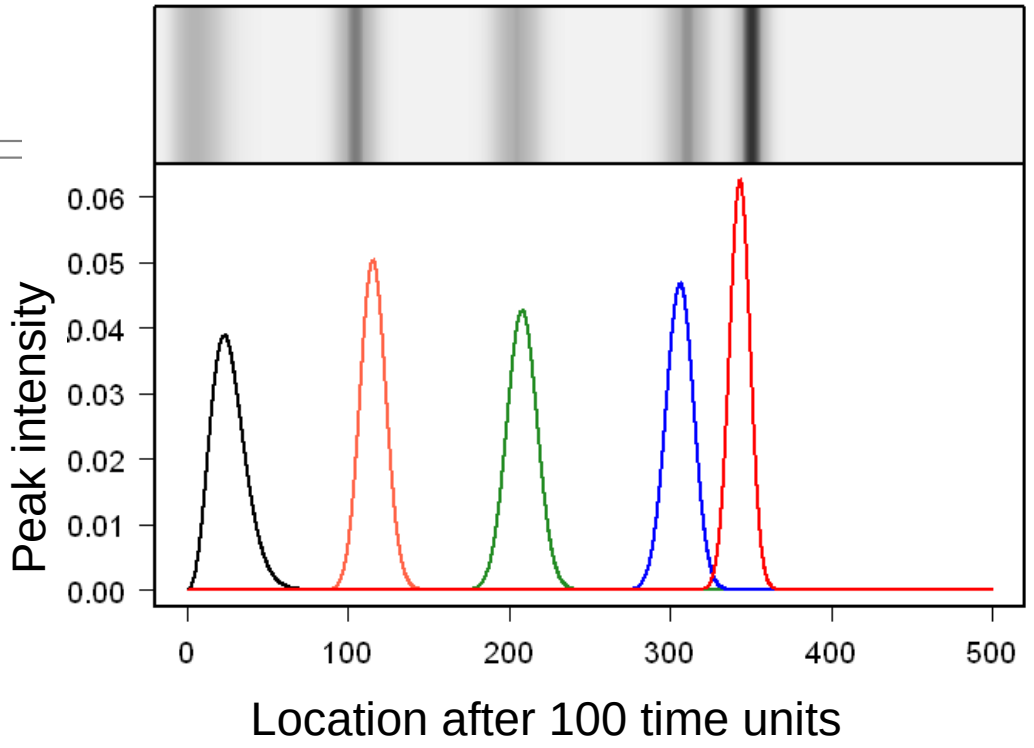
➔ **R much faster than expected**

Chromatography model

Similar approach like insect drift
fixed phase, mobile phase

Example:

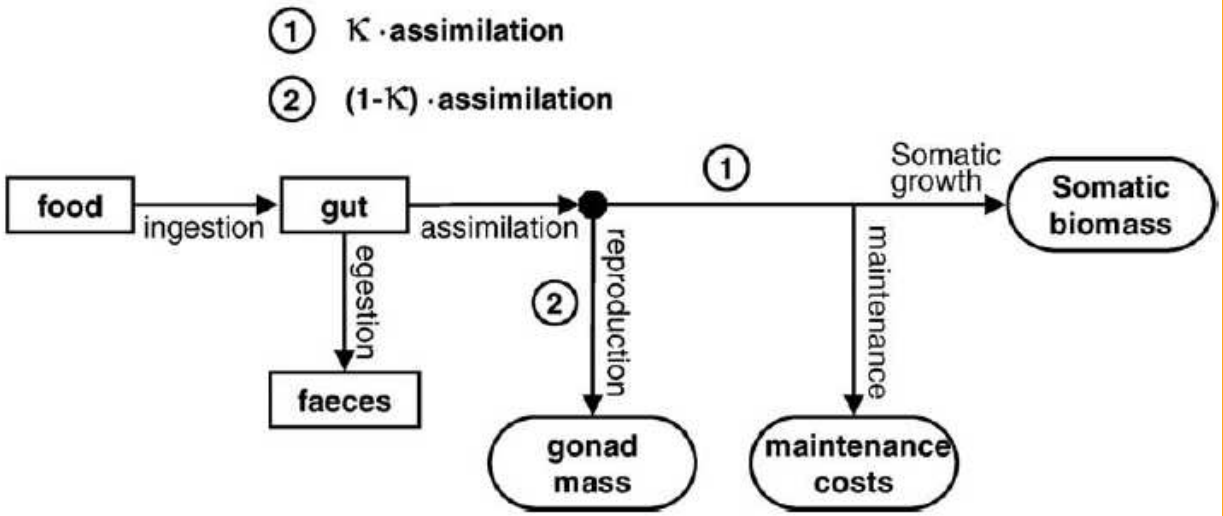
- 5 chemical species,
- 500...5000 grid cells
- 100 external time steps



Computational Effort:

Grid cells	Equations	model in C	model in R
500	500 * 2 * 5 = 5,000	0.8 s	1.95 s
5000	5,000 * 2 * 5 = 50,000	59 s	66 s

Example 3: Population dynamics of Daphnia (water flea)

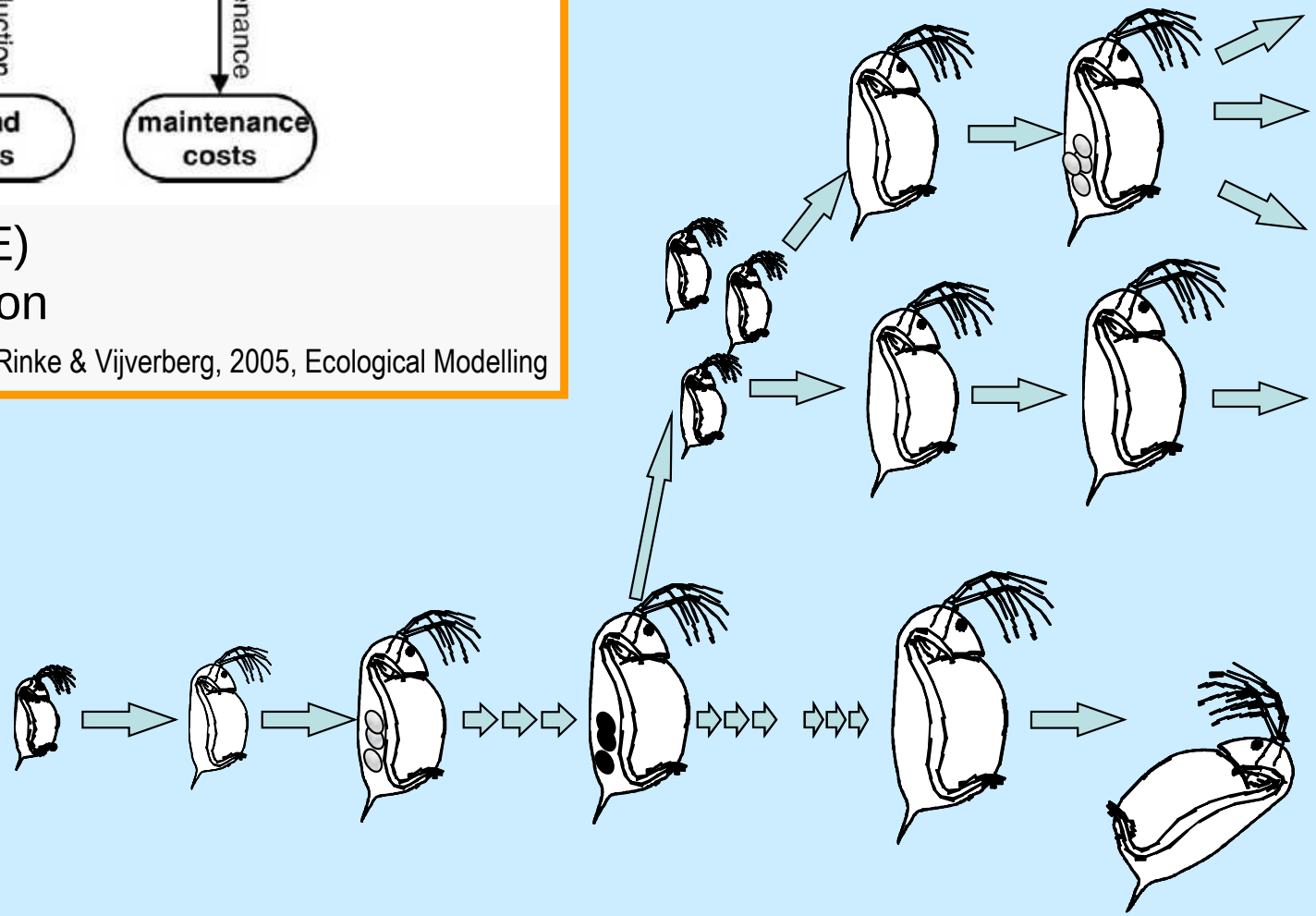


Bioenergetic model (ODE)
→ growth and reproduction

Rinke & Vijverberg, 2005, Ecological Modelling

Agent-based simulation

e.g.
sample of 1000 ... 2000 individuals
→ Population dynamics



Agent-based Daphnia simulation (ABM) with bioenergetic growth model (ODE)

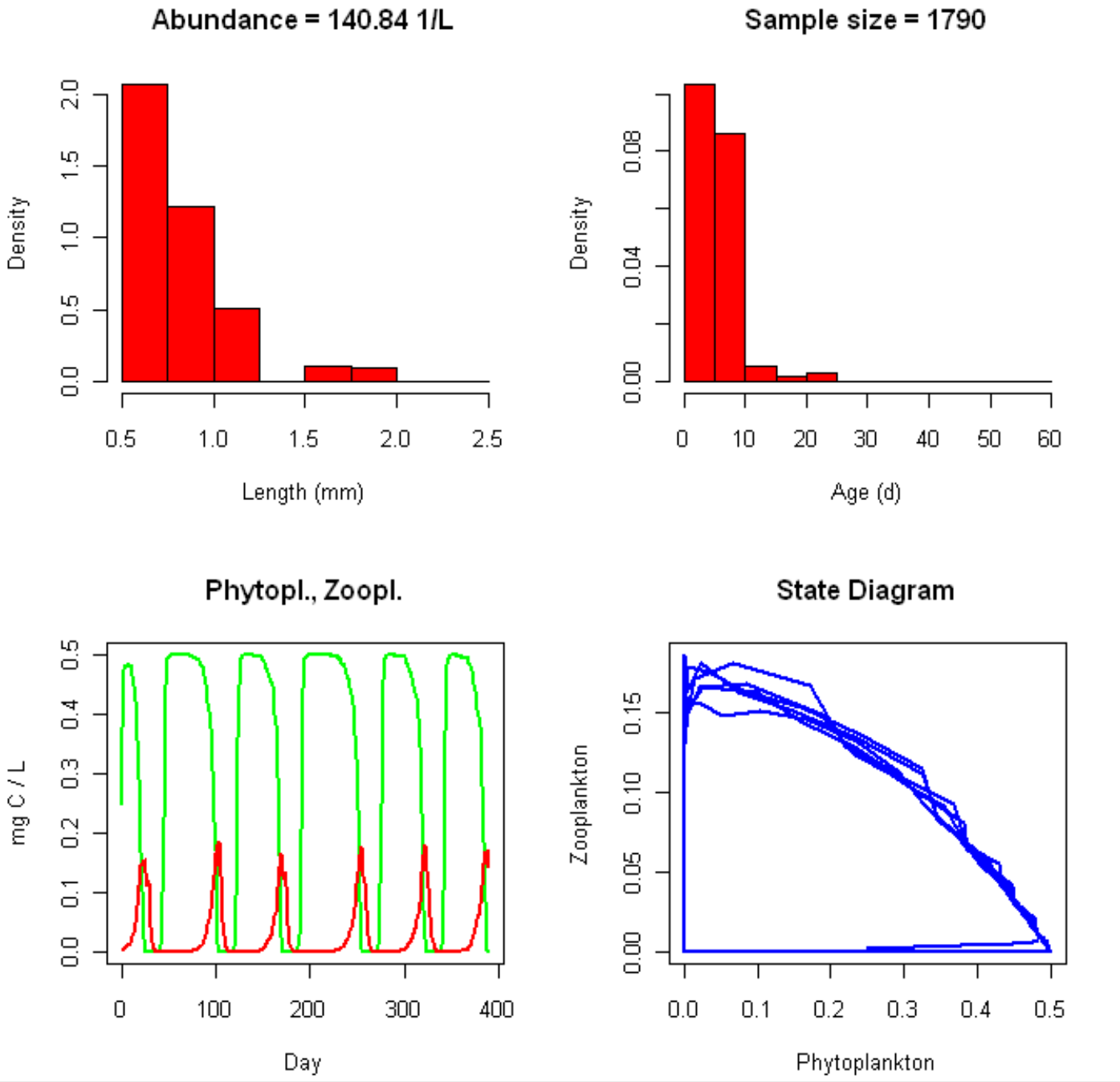
Effort needed per 100 simulation days:

100 ABM time steps
1000 ODE time steps

1000 …2000 individuals
→ 4000 … 8000 equations

Performance

pure R:	…..136 s
ABM in R, ODEs in C:	……..65s



The Lotka-Volterra-type model revisited

$$\frac{dS}{dt} = import - b \cdot S \cdot P$$
$$\frac{dP}{dt} = c \cdot S \cdot P - d \cdot P \cdot K$$
$$\frac{dK}{dt} = e \cdot P \cdot K - f \cdot K$$

Calling a small model many times

Remember:

3 equations (resource, producer, consumer)

Rectangular external signal (import of resource)

Now 10,000 time steps

Model in: R signal with if … else in model function …… 8 s

R with approxfun (10,000 rows in data table) …… 65 s

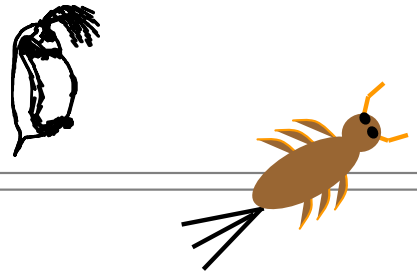
C, bisectioning, similar to approxfun ……..…… 0.16 s

C, sequential search in ordered forcings ..…… 0.03 s

The integrator (lsoda) and the model are able to communicate directly at the machine code level.

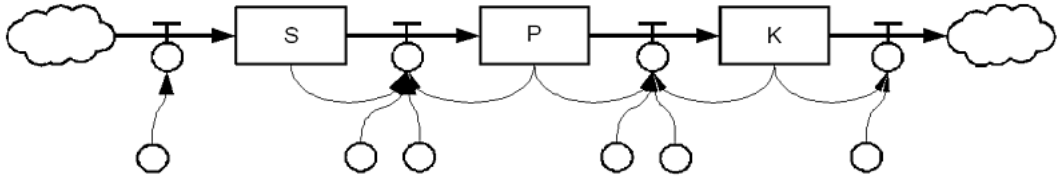
➔ Simulation without "friction"

Efficiency is more than CPU time

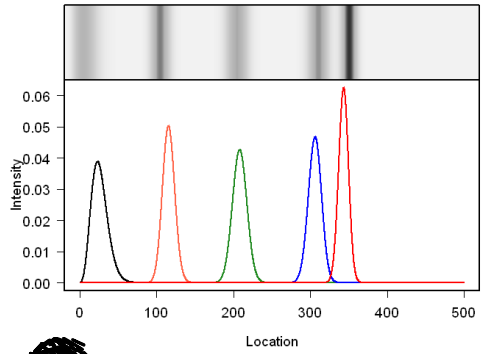


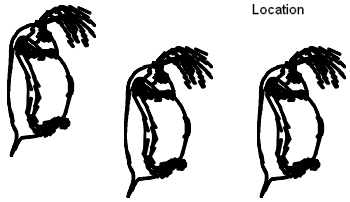
- Programming in R:
  - more convenient than C or Fortran
  - ... more interactive, more compact code, …
- High-level statistical algorithms and graphics
  - I can do almost everything in one system
  - no need to export / import data to other software
  - stats, … deSolve, FME, … Sweave
  - → support data analysis and report writing
- Open Source
  - Allows to work with talented people on a global scale
  - Enables me to share my code with others (and use theirs)
    - → model collection package **simecolModels**

Conclusion: R is powerful for system dynamics

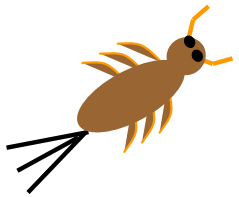


A few rules:





- **Vectorization! Matrix algebra!**
  - Large models with identical equations = fast in pure R
  - ABMs are efficient with data frames and subset()
- **Avoid unnecessary copying** of large objects.
  - Sometimes it helps to prefer matrices over data frames.
  - Avoid interpolation (i.e. approx),
  - If approx is unavoidable, minimize the tables.
- Complex systems of equations or frequent calls to small models:
  - considerable performance gain if **core functions in C or Fortran**
  - **consider direct communication** between deSolve and compiled code
- **R is a good investment** even in that cases:
  - It handles my input and output data
  - so I can concentrate on the equations.



TECHNISCHE UNIVERSITÄT DRESDEN

Thomas.Petzoldt@tu-dresden.de
http://tu-dresden.de/Members/thomas.petzoldt

Thanks to useR!

Karline Soetaert, Woodrow Setzer, Carola Winkelmann

and

Thank You!

package **deSolve** (Soetaert, Petzoldt, Setzer):
http://cran.r-project.org/web/packages/deSolve/

packages **simecol** and **simecolModels**
http://www.simecol.de
http://r-forge.r-project.org/projects/simecol/