TECHNISCHE UNIVERSITÄT DRESDEN

A code generator for ODE-based models

–

R-package `rodeo`

david.kneis@tu-dresden.de

Special seminar of the Systems Analysis and Modelling group at EAWAG, 2016-05-03

Outline

Outline

Background

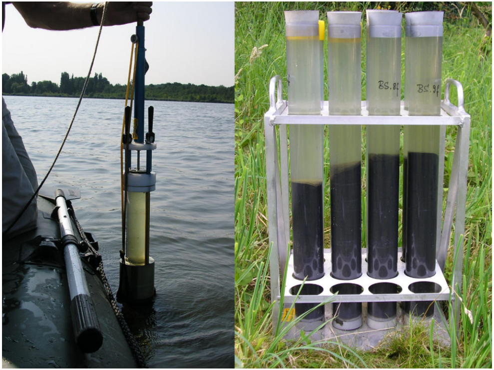
- Lake eutrophication
- Flood management
- Operational runoff forecasting
- Early diagenesis of lake sediments
- Fate of antibiotic resistant bacteria

Background

- Lake eutrophication
- Flood management
- Operational runoff forecasting
- Early diagenesis of lake sediments
- Fate of antibiotic resistant bacteria

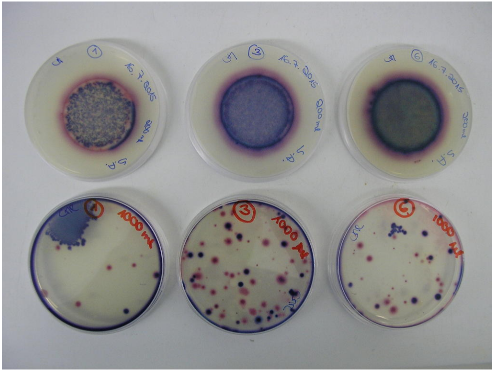
Background

- Lake eutrophication
- Flood management
- Operational runoff forecasting
- Early diagenesis of lake sediments
- Fate of antibiotic resistant bacteria



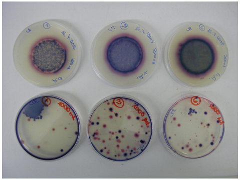
Background

- Lake eutrophication
- Flood management
- Operational runoff forecasting
- Early diagenesis of lake sediments
- Fate of antibiotic resistant bacteria



Background

- Lake eutrophication
- Flood management
- Operational runoff forecasting
- Early diagenesis of lake sediments
- Fate of antibiotic resistant bacteria



→ Several years of model/software development

Re-invention of the wheel?

- Best way to learn modeling is via model development.
- 'Monolithic codes' are hard to extend.
- Rising interest in structural uncertainty
  $\rightarrow$ Need for Re-implementations

Outline

Documentation

- Often incomplete or outdated
- Mistakes in published equations
- Source code alone not sufficient

$\rightarrow$ Embedded / automatic documentation

Portability & life time

- Implementation in specific language / framework
- Impedes collaborative development
- Software undergoes aging

$\rightarrow$ True portability

$\rightarrow$ Equations to be separated from source code

Handling of large arrays

- Access by index: Hard to read / maintain
- Access by name: Slow

$\rightarrow$ Combine the two options

Computational efficiency

- Interpreted code is convenient but relatively slow
- Need for high-performance (Optimization, Uncertainty, ...)

$\rightarrow$ Use compiled code sections

Redundant terms

- Repeated evaluation wastes time
- Code is difficult to maintain

$\rightarrow$ Use proper notation to reduce redundancies

$\rightarrow$ Let the compiler eliminate them

Various model interfaces

- Effort for users
- Individual pre-/post-processors
- Impedes coupling of models

$\rightarrow$ Unified interface

Wish list

- Built-in documentation
- True portability
- Save & fast array access
- Compiled code sections
- Less redundancies
- Unified interface

$\rightarrow$ `rodeo` is one attempt, among others, to achieve this

Scope

- Models built on simultaneous ODE

$$\frac{d}{dt}Y_1 = f(time, Y, parameters)$$

$$\dots$$

$$\frac{d}{dt}Y_n = f(time, Y, parameters)$$

- Numerical integration or steady-state estimation

Outline

Introduction

Objectives

Concepts

Minimalistic example

Specific features & limitations

Applications

Summary

Hands-on part

Two pillars

(1) Use of a table-based standard notation for ODE

- Built-in documentation
- Less redundancies
- Unified interface

Two pillars

(1) Use of a table-based standard notation for ODE

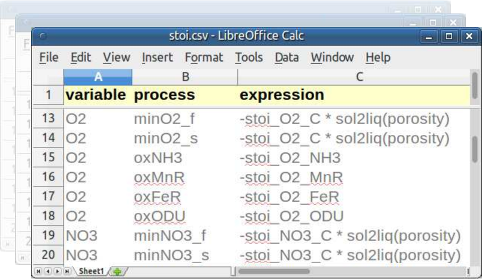
- Built-in documentation
- Less redundancies
- Unified interface

(2) Automatic code generation

- Save & fast array access
- Use of compiled code
- Portability

Overview

Table-based model definition



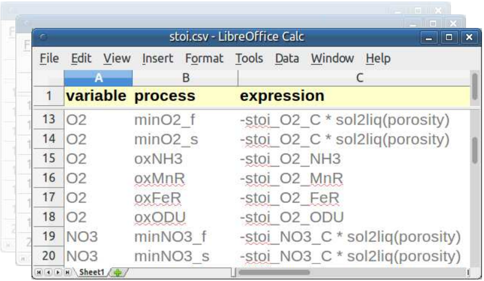
stoi.csv - LibreOffice Calc

	A	B	C
1	**variable**	**process**	**expression**
13	O2	minO2_f	-stoi_O2_C * sol2liq(porosity)
14	O2	minO2_s	-stoi_O2_C * sol2liq(porosity)
15	O2	oxNH3	-stoi_O2_NH3
16	O2	oxMnR	-stoi_O2_MnR
17	O2	oxFeR	-stoi_O2_FeR
18	O2	oxODU	-stoi_O2_ODU
19	NO3	minNO3_f	-stoi_NO3_C * sol2liq(porosity)
20	NO3	minNO3_s	-stoi_NO3_C * sol2liq(porosity)

→ *rodeo* object

Overview

Table-based model definition



	A	B	C
1	**variable**	**process**	**expression**
13	O2	minO2_f	-stoi_O2_C * sol2liq(porosity)
14	O2	minO2_s	-stoi_O2_C * sol2liq(porosity)
15	O2	oxNH3	-stoi_O2_NH3
16	O2	oxMnR	-stoi_O2_MnR
17	O2	oxFeR	-stoi_O2_FeR
18	O2	oxODU	-stoi_O2_ODU
19	NO3	minNO3_f	-stoi_NO3_C * sol2liq(porosity)
20	NO3	minNO3_s	-stoi_NO3_C * sol2liq(porosity)

Components of *tex* or *html* documents

rodeo object

R code

Fortran code

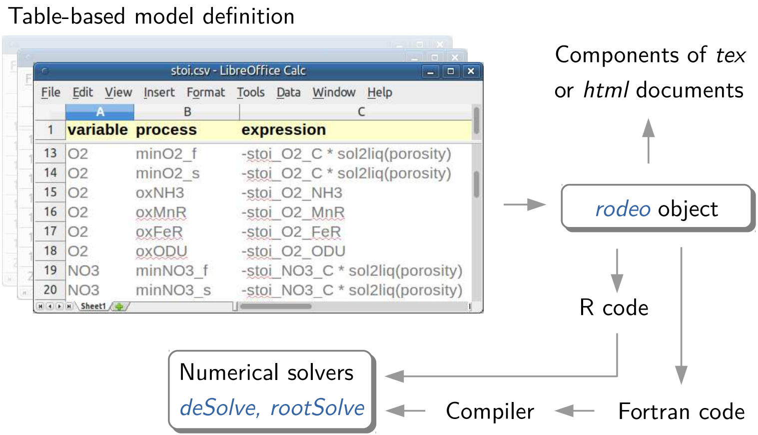


Table-based model definition
stoi.csv - LibreOffice Calc
File Edit View Insert Format Tools Data Window Help
variable process expression
13 O2 minO2_f -stoi_O2_C * sol2liq(porosity)
14 O2 minO2_s -stoi_O2_C * sol2liq(porosity)
15 O2 oxNH3 -stoi_O2_NH3
16 O2 oxMnR -stoi_O2_MnR
17 O2 oxFeR -stoi_O2_FeR
18 O2 oxODU -stoi_O2_ODU
19 NO3 minNO3_f -stoi_NO3_C * sol2liq(porosity)
20 NO3 minNO3_s -stoi_NO3_C * sol2liq(porosity)
Sheet1
Components of tex or html documents
rodeo object
R code
Numerical solvers
deSolve, rootSolve
Compiler
Fortran code

Overview

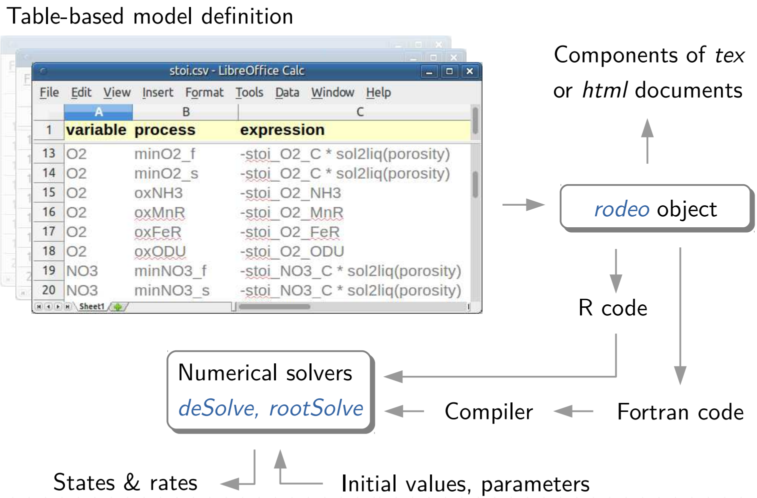
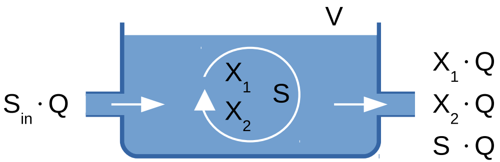


Table-based model definition
stoi.csv - LibreOffice Calc
File Edit View Insert Format Tools Data Window Help
A B C
1 variable process expression
13 O2 minO2_f -stoi_O2_C * sol2liq(porosity)
14 O2 minO2_s -stoi_O2_C * sol2liq(porosity)
15 O2 oxNH3 -stoi_O2_NH3
16 O2 oxMnR -stoi_O2_MnR
17 O2 oxFeR -stoi_O2_FeR
18 O2 oxODU -stoi_O2_ODU
19 NO3 minNO3_f -stoi_NO3_C * sol2liq(porosity)
20 NO3 minNO3_s -stoi_NO3_C * sol2liq(porosity)
Sheet1
Components of *tex*
or *html* documents
rodeo object
R code
Numerical solvers
deSolve, rootSolve
Compiler
Fortran code
States & rates
Initial values, parameters

Outline

Modeled system

- Mixed reactor with constant volume V and flow rate Q
- Two species (X_1, X_2) competing for dissolved resource S



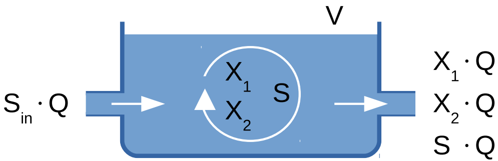
Corresponding ODE

$$\frac{d}{dt}X_1 = r_1 \cdot X_1 \cdot \frac{S}{S+h_1} - X_1 \cdot \frac{Q}{V}$$

$$\frac{d}{dt}X_2 = r_2 \cdot X_2 \cdot \frac{S}{S+h_2} - X_2 \cdot \frac{Q}{V}$$

$$\frac{d}{dt}S = -c_1 \cdot r_1 \cdot X_1 \cdot \frac{S}{S+h_1} - c_2 \cdot r_2 \cdot X_2 \cdot \frac{S}{S+h_2} + (S_{in} - S) \cdot \frac{Q}{V}$$

Growth
Im/Export



Matrix notation

$$\frac{d}{dt}X_1 = r_1 \cdot X_1 \cdot \frac{S}{S+h_1} - X_1 \cdot \frac{Q}{V}$$

$$\frac{d}{dt}X_2 = r_2 \cdot X_2 \cdot \frac{S}{S+h_2} - X_2 \cdot \frac{Q}{V}$$

$$\frac{d}{dt}S = -c_1 \cdot r_1 \cdot X_1 \cdot \frac{S}{S+h_1} - c_2 \cdot r_2 \cdot X_2 \cdot \frac{S}{S+h_2} + (S_{in} - S) \cdot \frac{Q}{V}$$

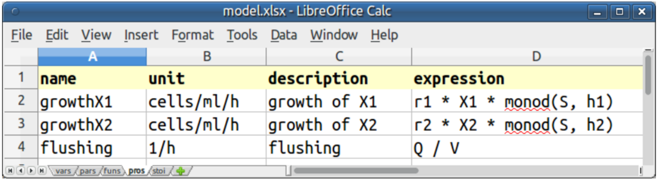
Growth

Im/Export

$$\frac{d}{dt}\begin{bmatrix} X_1 \\ X_2 \\ S \end{bmatrix} = \begin{bmatrix} 1 & 0 & -X_1 \\ 0 & 1 & -X_2 \\ -c_1 & -c_2 & S_{in} - S \end{bmatrix} \cdot \begin{bmatrix} r_1 \cdot X_1 \cdot S/(S+h_1) \\ r_2 \cdot X_2 \cdot S/(S+h_2) \\ Q/V \end{bmatrix}$$

Table of Processes

$$\frac{d}{dt}\begin{bmatrix} X_1 \\ X_2 \\ S \end{bmatrix} = \begin{bmatrix} 1 & 0 & -X_1 \\ 0 & 1 & -X_2 \\ -c_1 & -c_2 & S_{in} - S \end{bmatrix} \cdot \begin{bmatrix} r_1 \cdot X_1 \cdot S/(S + h_1) \\ r_2 \cdot X_2 \cdot S/(S + h_2) \\ Q/V \end{bmatrix}$$



model.xlsx - LibreOffice Calc

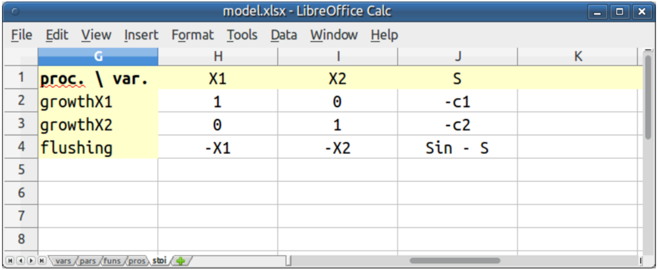
File Edit View Insert Format Tools Data Window Help

name	unit	description	expression
growthX1	cells/ml/h	growth of X1	r1 * X1 * monod(S, h1)
growthX2	cells/ml/h	growth of X2	r2 * X2 * monod(S, h2)
flushing	1/h	flushing	Q / V

vars pars funs pros stoi

Table of stoichiometric factors

$$\frac{d}{dt}\begin{bmatrix} X_1 \\ X_2 \\ S \end{bmatrix} = \begin{bmatrix} 1 & 0 & -X_1 \\ 0 & 1 & -X_2 \\ -c_1 & -c_2 & S_{in} - S \end{bmatrix} \cdot \begin{bmatrix} r_1 \cdot X_1 \cdot S/(S + h_1) \\ r_2 \cdot X_2 \cdot S/(S + h_2) \\ Q/V \end{bmatrix}$$



	G	H	I	J	K
1	proc. \ var.	X1	X2	S	
2	growthX1	1	0	-c1	
3	growthX2	0	1	-c2	
4	flushing	-X1	-X2	Sin - S	
5					
6					
7					
8					

Tables with declarations

Variables

model.xlsx - LibreOffice Calc

name	unit	description	tex	html
X1	cells/ml	abundance species 1	X_1	X₁
X2	cells/ml	abundance species 2	X_2	X₂
S	mg/l	substrate	S	S

vars / pars / funs / pros / stoi

Parameters

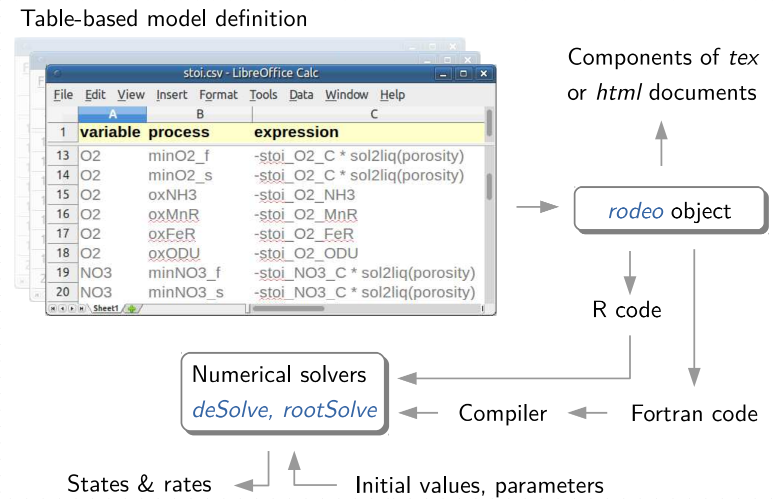
model.xlsx - LibreOffice Calc

name	unit	description	tex	html
r1	1/h	growth rate 1	r_1	r₁
r2	1/h	growth rate 2	r_2	r₂
h1	mg/ml	half sat. spec. 1	h_1	h₁
h2	mg/ml	half sat. spec. 2	h_2	h₂
c1	mg/cell	stoich. of X1	c_1	c₁
c2	mg/cell	stoich. of X2	c_2	c₂
Q	ml/h	flow rate	Q	Q
V	ml	volume	V	V
Sin	mg/ml	substrate in inflow	S_{in}	S_{in}

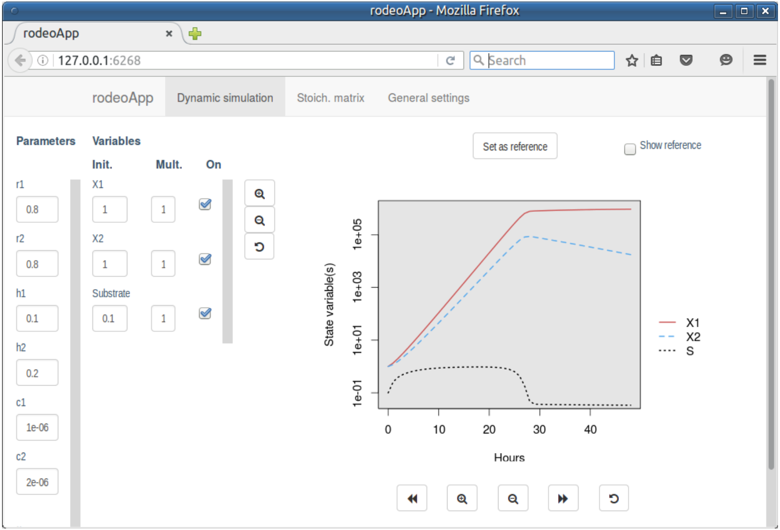
vars / pars / funs / pros / stoi

+ Functions

Code & document generation



Auto-generated GUI



Outline

Support for PDE

$$\frac{\partial c}{\partial t} = \underbrace{D \cdot \frac{\partial^2 c}{\partial x^2}}_{\text{Dispersion}} - \underbrace{u \cdot \frac{\partial c}{\partial x}}_{\text{Advection}} + \underbrace{R}_{\text{Reactions}}$$

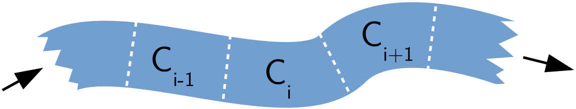
Support for PDE

$$\frac{\partial c}{\partial t} = \underbrace{D \cdot \frac{\partial^2 c}{\partial x^2}}_{\text{Dispersion}} - \underbrace{u \cdot \frac{\partial c}{\partial x}}_{\text{Advection}} + \underbrace{R}_{\text{Reactions}}$$

Method-of-lines

$$\frac{dc_i}{dt} = D \cdot \frac{(c_{i+1} - c_i) - (c_i - c_{i-1})}{\Delta x^2} - u \cdot \frac{c_i - c_{i-1}}{\Delta x} + R_i$$

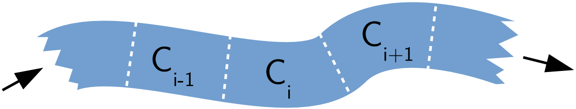
Support for PDE



Method-of-lines

$$\frac{dc_i}{dt} \quad = \quad D \cdot \frac{(c_{i+1} - c_i) - (c_i - c_{i-1})}{\Delta x^2} \quad - \quad u \cdot \frac{c_i - c_{i-1}}{\Delta x} \quad + \quad R_i$$

Support for PDE



Method-of-lines

$$\frac{dc_i}{dt} \quad = \quad D \cdot \frac{(c_{i+1} - c_i) - (c_i - c_{i-1})}{\Delta x^2} \quad - \quad u \cdot \frac{c_i - c_{i-1}}{\Delta x} \quad + \quad R_i$$

Function-like syntax to access adjacent cells, e.g.

```
u / dx * ( c - left(c) )
```

Forcing functions

`foo(time)` can appear in right hand side expressions

Actual functions must be defined

↙ ↘

Analytical

Interpolation

- Use `approxFun` in `R`-based models
- Use `rodeo`-generated Fortran code

Known limitations

- No forced documentation for user-function arguments
- No built-in support for 2D or 3D models
- Generated code uses a Fortran 2008 feature

Known limitations

CRAN Package Check Results for Package rodeo

Last updated on 2016-04-28 06:47:39.

Flavor	Version	$T_{install}$	T_{check}	T_{total}	Status	Flags
r-devel-linux-x86_64-debian-gcc	0.3	1.18	18.93	20.11	OK	
r-devel-linux-x86_64-fedora-clang	0.3			34.64	OK	
r-devel-linux-x86_64-fedora-gcc	0.3			22.28	OK	
r-devel-osx-x86_64-clang	0.3			37.30	OK	
r-devel-windows-ix86+x86_64	0.3	5.00	59.00	64.00	OK	
r-patched-linux-x86_64	0.3	1.19	18.58	19.77	OK	
r-patched-solaris-sparc	0.3			184.30	WARN	
r-patched-solaris-x86	0.3			41.20	WARN	
r-release-linux-x86_64	0.3	1.36	21.82	23.18	OK	
r-release-osx-x86_64-mavericks	0.3				OK	
r-release-windows-ix86+x86_64	0.3	5.00	72.00	77.00	OK	
r-oldrel-windows-ix86+x86_64	0.3	5.00	82.00	87.00	OK	

WARN: Compiler doesn't implement pointer initialization yet

Outline

`rodeo`-based projects

Existing

- Lake ecology (0D)
- Sediment diagenesis
- Dynamics of E. coli
- Prey-predator systems

`rodeo`-based projects

Existing

- Lake ecology (0D)
- Sediment diagenesis
- Dynamics of E. coli
- Prey-predator systems

Planned

- Lake ecology (1D)
- Activated sludge model

`rodeo`-based projects

Existing

- Lake ecology (0D)
- Sediment diagenesis
- Dynamics of E. coli
- Prey-predator systems

Planned

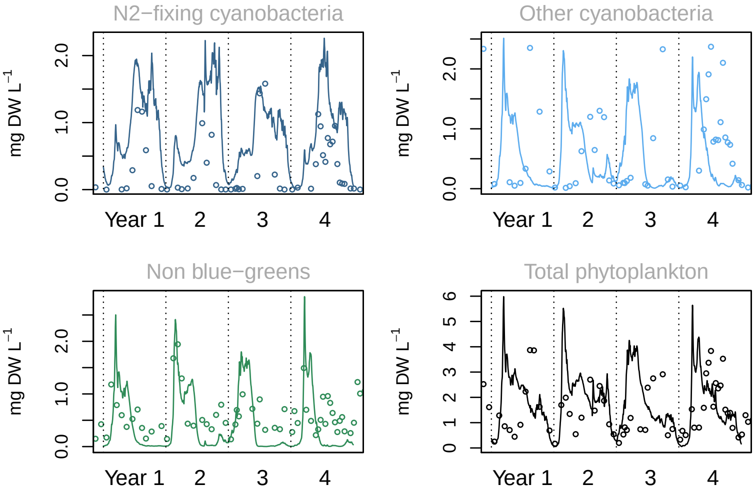
- Lake ecology (1D)
- Activated sludge model

Ecological lake model

- Heavily based on BELAMO
- Applied to a shallow lake, 1.3 km^2, z_{mean} 2.1 m

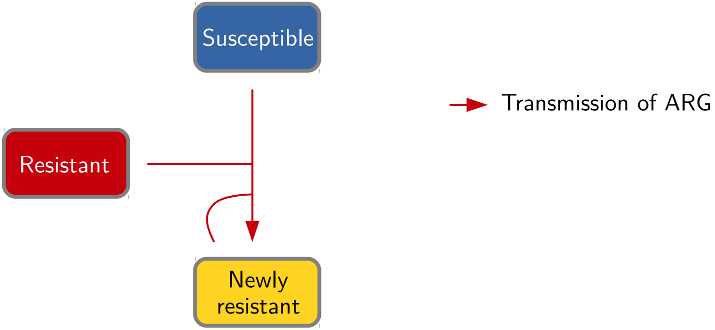
Contribution of N_2-fixation to Nitrogen balance?

Ecological lake model

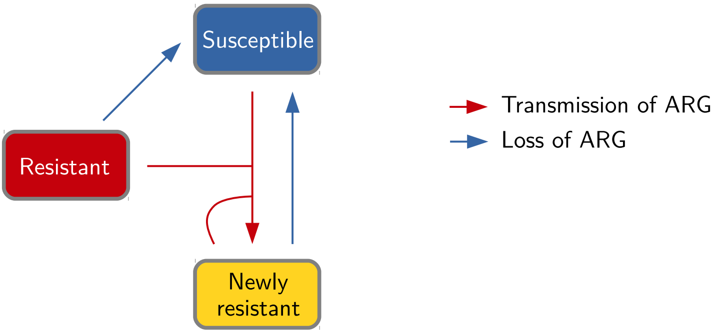


Data: BTU & WSA, Model: Omlin et al. (2001) modified by J. Feldbauer, M. Nisotaki, Y. Zhao

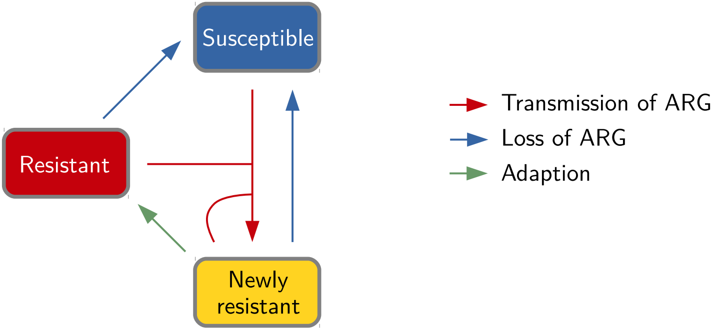
Antibiotic resistance



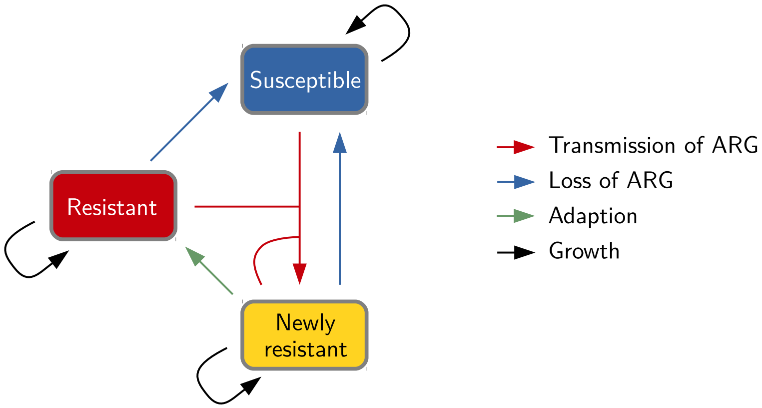
Antibiotic resistance



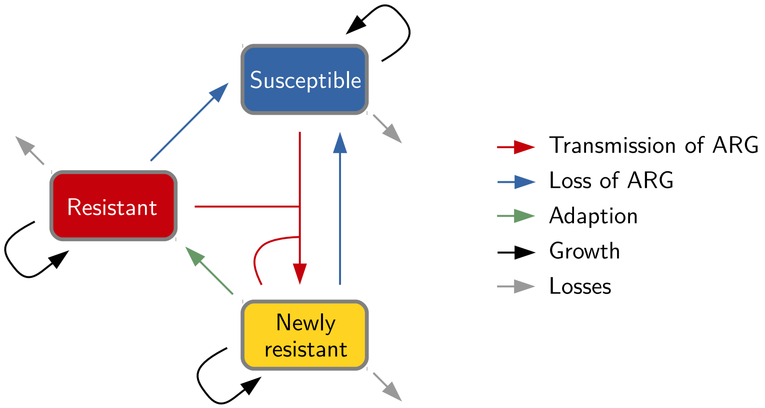
Antibiotic resistance



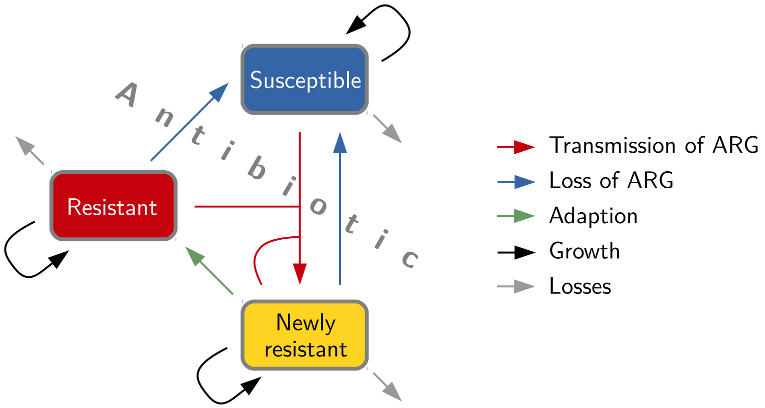
Antibiotic resistance



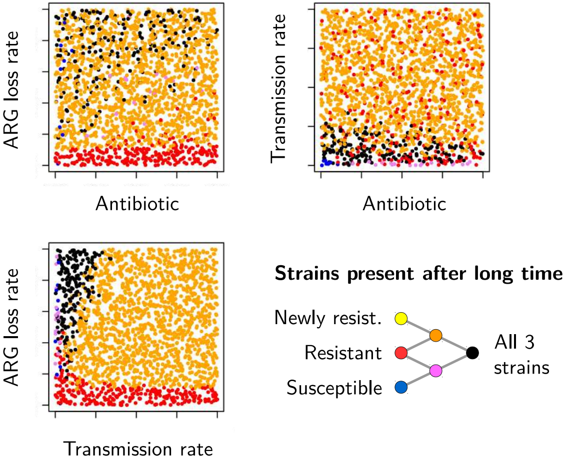
Antibiotic resistance



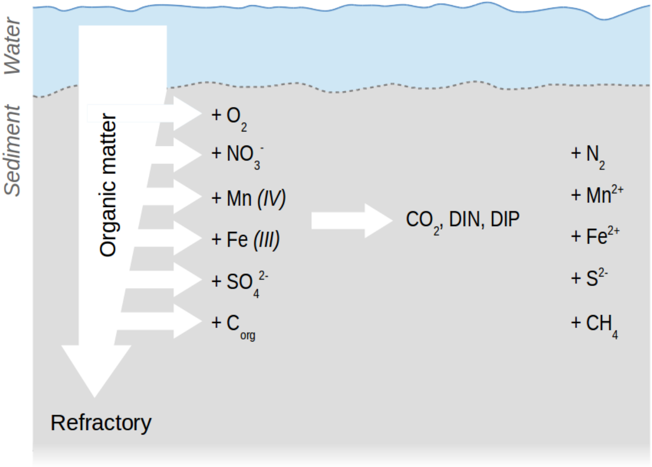
Antibiotic resistance



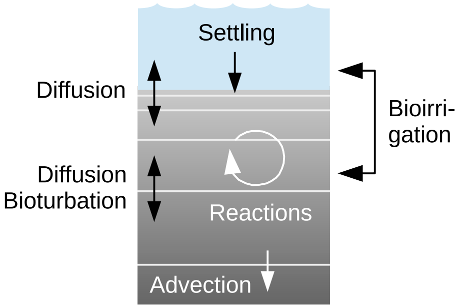
Antibiotic resistance



Early diagenesis



Early diagenesis



Early diagenesis

	C_{org}	DIP	IMP	NH_4^+	sorb. NH_4^+	NO_3^-	O_2	ODU
Settling	▲							
Mineral. (O_2)	▼	▲		▲			▼	
Mineral. (NO_3^-)	▼	▲		▲		▼		
Mineral. (anox.)	▼	▲		▲				▲
Nitrification				▼		▲	▼	
Oxidation w. O_2							▼	▼
Oxidation w. NO_3^-						▼		▼
P-Immobilization		▼	▲					
P-Remobilization		▲	▼					
NH_4^+-Sorption				▼	▲			
Advection ↓	●	●	●	●	●	●	●	●
Advection ↑		●		●		●	●	●
Diffus. transport	●	●	●	●	●	●	●	●
Diffus. at surface		●		●		●	●	●
Bioirrigation		●		●		●	●	●

Concentration

▲ increases

▼ decreases

● goes up or down

DIP: Dissolved inorg. P

IMP: Immobile inorg. P

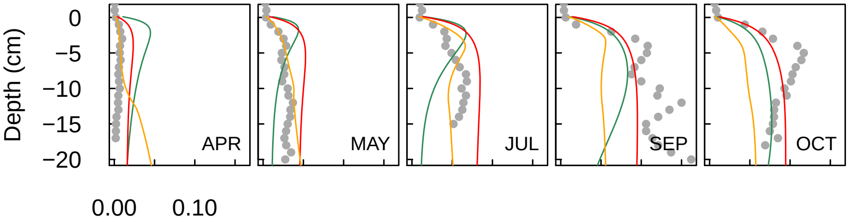
ODU: Mn^{2+}, Fe^{2+}, HS^-, ...

Basic concepts borrowed from Soetaert et al. (1996)

Early diagenesis

● Phosphorus in pore water, observed

/// Simulated with different model structures



mmol L^{-1}

Obs. data: Björn Grüneberg, BTU

Outline

Summary

Scope	Implementation of ODE models (+ 1D PDE)
Concepts	Table-based notation & code generation
Benefit	Simplicity and performance
Uses	Project work & teaching

Package	`https://cran.r-project.org/package=rodeo`
	`https://github.com/dkneis/rodeo`
Examples	`http://dkneis.github.io`
	`http://limno-live.hydro.tu-dresden.de/`

Thanks !

Sponsored by the Federal Ministry of Education and Research, Germany, Grant no. 0033W015EN, 02WU1351A

Outline

Required software

- Recent R version
- Developer tools (Rtools on Windows)
- R packages
  - `install.packages('deSolve')`
  - `install.packages('readxl')`
  - `install.packages('rodeo')`

Link to instructions on http://dkneis.github.io

`rodeo` objects

https://cran.r-project.org/package=rodeo

- `rodeo` class is a 'reference class'
- Creation: `object <- new('rodeo', <data>)`
- Usage: `object$method()`

`rodeo` objects

https://cran.r-project.org/package=rodeo

- `rodeo` class is a 'reference class'
- Creation: `object <- new('rodeo', <data>)`
- Usage: `object$method()`

```
install.packages('rodeo')   # done this already?
library('rodeo')
?rodeo
vignette('rodeo')
```

`deSolve` integrators

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

- Switch between stiff and non-stiff methods
- Structure of Jacobian can be specified
- Works with compiled code in shared library

deSolve integrators

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

- Switch between stiff and non-stiff methods
- Structure of Jacobian can be specified
- Works with compiled code in shared library

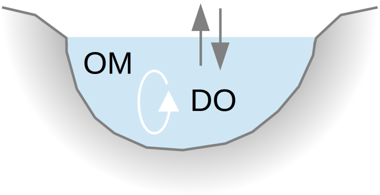
```
install.packages('deSolve')   # done this already?
library('deSolve')
?lsoda
?ode
```

Demo examples

- See links on http://dkneis.github.io
- Available in latest `rodeo` package (not on CRAN yet)

Streeter-Phelps

OM	Organic matter (mg/L)
DO	Dissolved oxygen (mg/L)



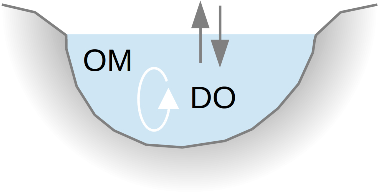
Streeter, W. H. and Phelps, W. B. (1925): A study of the pollution and natural purification of the Ohio River. Public Health Bull. 146, US Public Health Service, Washington DC.

→ Essential extensions developed in past 90 years

Streeter-Phelps

$$\frac{d}{dt}OM = -\, k_d \cdot OM$$

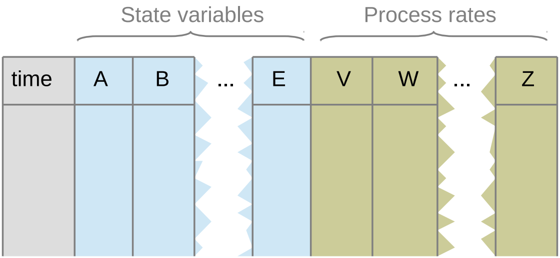
$$\frac{d}{dt}DO = -\, k_d \cdot OM \cdot s + k_a \cdot (DO_{sat} - DO)$$



Symbol	Units	Descr.
k_d	d^{-1}	Decay rate
k_a	d^{-1}	Aeration rate
s	Mass ratio	DO consumed per degraded OM
DO_{sat}	mg/L	O_2 saturation level

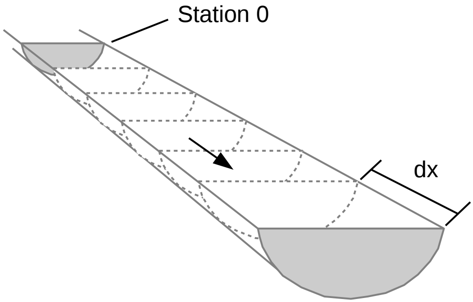
Streeter-Phelps

`deSolve` output for 0D `rodeo` models

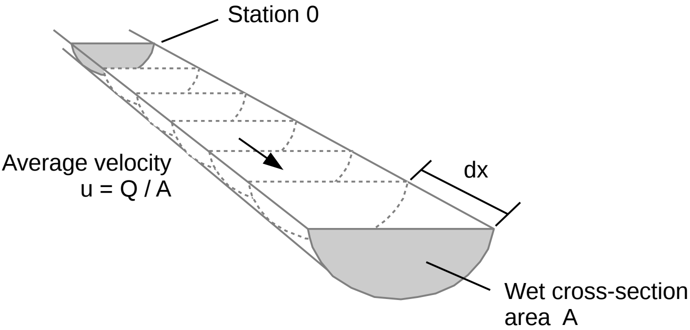



```
is.matrix(out)   # TRUE
ncol(out) == 1 + m$lenVars() + m$lenPros()   # m: model
colnames(out) == c('time', m$namesVars(), m$namesPros())
```

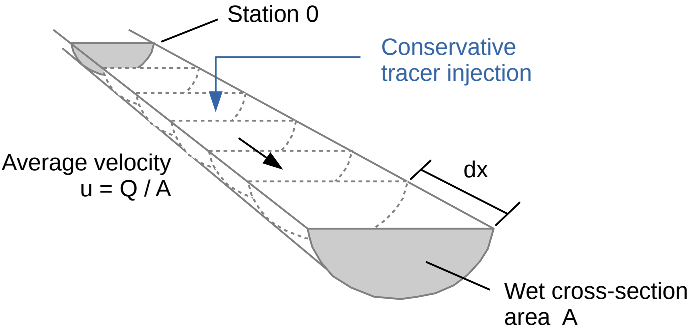
Advection-dispersion



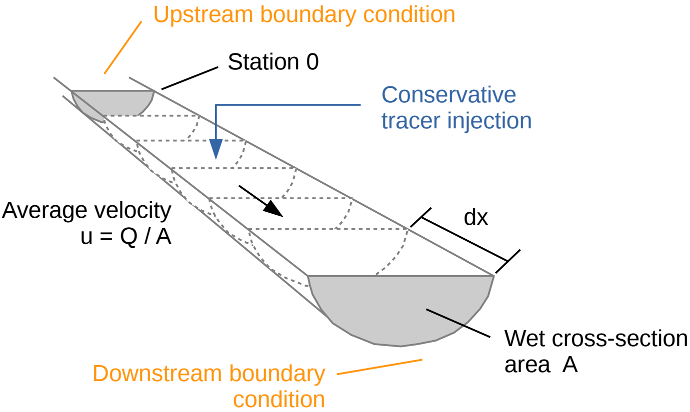
Advection-dispersion



Advection-dispersion

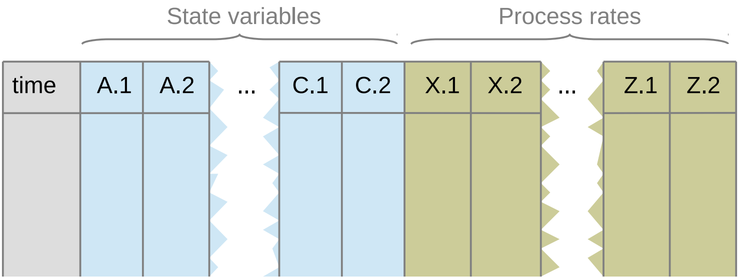


Advection-dispersion



Advection-dispersion

`deSolve` output for 1D `rodeo` models




```
ncol(out) == 1 + m$lenVars() * nBox + m$lenPros() * nBox
```

Advection-dispersion

`deSolve` output for 1D `rodeo` models

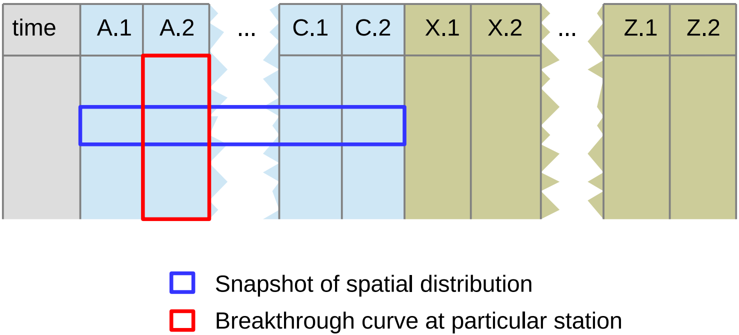


Table file formats

Delimited text	▸ Powerful editors (regular exp., syntax highlight) ▸ Version control ▸ Many processing options (LaTeX, data base, ...) ▸ Portable (but newline & encoding issues)
Spreadsheet	▸ Tabular view ▸ All tables kept in a single file ▸ Portable (different issues)

$\rightarrow$ Best used in combination