Streeter-Phelps Model

david.kneis@tu-dresden.de

Version: March 11, 2015

– This version of the PDF is optimized for presentation
– Use the side bar for convenient navigation
– Feedback and bug reports are welcome

Outline

Introduction

Analytical solution

Numerical solution

Model extension: Oxygen limitation

Further possible extensions

Spatially distributed model (1D)

Outlook

Outline

Introduction

Analytical solution

Numerical solution

Model extension: Oxygen limitation

Further possible extensions

Spatially distributed model (1D)

Outlook

Subject

- Organic waste is discharged into a river
- The organic matter is degraded by microorganisms under aerobic conditions
- What is the O_2 concentration downstream of the effluent?

Subject

- Organic waste is discharged into a river
- The organic matter is degraded by microorganisms under aerobic conditions
- What is the O_2 concentration downstream of the effluent?

A pioneering publication:

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.

Practical aspects

- What are common sources of organic waste?

Practical aspects

- What are common sources of organic waste?
  e.g. WWTP, paper and food industry

Practical aspects

- What are common sources of organic waste?
  e.g. WWTP, paper and food industry
- How could a basic reaction equation look line?

Practical aspects

- What are common sources of organic waste?
  e.g. WWTP, paper and food industry

- How could a basic reaction equation look line?
  e.g. Oxidation of Glucose

Practical aspects

- What are common sources of organic waste?
  e.g. WWTP, paper and food industry
- How could a basic reaction equation look line?
  e.g. Oxidation of Glucose
- How to measure general organic pollution?

Practical aspects

- What are common sources of organic waste?
  e.g. WWTP, paper and food industry

- How could a basic reaction equation look line?
  e.g. Oxidation of Glucose

- How to measure general organic pollution?
  z.B. COD, BOD, TOC

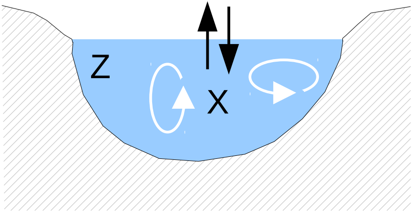
Practical aspects

- What are common sources of organic waste?
  e.g. WWTP, paper and food industry
- How could a basic reaction equation look line?
  e.g. Oxidation of Glucose
- How to measure general organic pollution?
  z.B. COD, BOD, TOC
- What happens if all O_2 has been consumed?

Practical aspects

- What are common sources of organic waste?
  e.g. WWTP, paper and food industry
- How could a basic reaction equation look line?
  e.g. Oxidation of Glucose
- How to measure general organic pollution?
  z.B. COD, BOD, TOC
- What happens if all O_2 has been consumed?
  $\rightarrow$ alternative pathways of mineralization

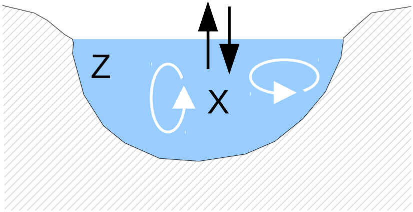
Derivation of equations



We consider a mixed system (no in-/outflow) with the following state variables:

Symbol	Units	Explanation
Z	mg/l	Degradable organic matter
X	mg/l	Dissolved oxygen

Derivation of equations



Considered processes:

- Aerobic decay of organic matter *Z* by bacteria suspended in the water column (1st order process)
- Consumption of oxygen *X* during mineralization of *Z*
- Exchange of oxygen between water and atmosphere

Derivation of equations

Differential eqiations (ODE) and parameters:

Derivation of equations

Differential eqiations (ODE) and parameters:

$$\frac{d}{dt}Z = -k_d \cdot Z \tag{1}$$

$$\frac{d}{dt}X = -k_d \cdot Z \cdot s + k_a \cdot (X_{sat} - X) \tag{2}$$

Symbol	**Units**	**Explanation**
k_d	1/Time	Decay rate
k_a	1/Time	Aeration rate
s	Mass X / mass Z	Stoichiometric factor
X_{sat}	mg/l	O_2 saturation level

Simplifying assumptions

- System does not turn anaerobic; equations only valid if $X \gg 0$
- Degradation happens in water column only (no bacteria attached to surfaces)
- Decay rate is constant (no growth of bacteria)
- Effect of temperature is neglected
- ...

Outline

Introduction

Analytical solution

Numerical solution

Model extension: Oxygen limitation

Further possible extensions

Spatially distributed model (1D)

Outlook

Re-Definition of state variables

Re-definition of state variables leads to simplified ODE:

Old	**New**	**Relation**	**Meaning**
Z	L	$L = Z$	Biochemical O_2 demand for complete degradation of Z
X	D	$D = X_{sat} - X$	O_2 saturation deficit

Re-Definition of state variables

Re-definition of state variables leads to simplified ODE:

Old	**New**	**Relation**	**Meaning**
Z	L	$L = Z$	Biochemical O_2 demand for complete degradation of Z
X	D	$D = X_{sat} - X$	O_2 saturation deficit

- L is usually labeled BOD (biochem. oxygen demand)
- Stoichiometric factor s equals 1 $\rightarrow$ omitted

Analytical solution

Re-Definition of state variables

Re-definition of state variables leads to simplified ODE:

Old	**New**	**Relation**	**Meaning**
Z	L	$L = Z$	Biochemical $\mathrm{O_2}$ demand for complete degradation of Z
X	D	$D = X_{sat} - X$	$\mathrm{O_2}$ saturation deficit

- L is usually labeled BOD (biochem. oxygen demand)
- Stoichiometric factor s equals 1 → omitted

Question: What sign can D take?

Original and new ODE system

Original

$$\frac{d}{dt}Z = -k_d \cdot Z \qquad \text{(1, rep.)}$$

$$\frac{d}{dt}X = -k_d \cdot Z \cdot s + k_a \cdot (X_{sat} - X) \qquad \text{(2, rep.)}$$

Original and new ODE system

Original

$$\frac{d}{dt}Z = -k_d \cdot Z \qquad \text{(1, rep.)}$$

$$\frac{d}{dt}X = -k_d \cdot Z \cdot s + k_a \cdot (X_{sat} - X) \qquad \text{(2, rep.)}$$

New

$$\frac{d}{dt}L = -k_d \cdot L \qquad (3)$$

$$\frac{d}{dt}D = k_d \cdot L - k_a \cdot D \qquad (4)$$

Hint

$$\frac{d}{dt}X = \frac{d}{dt}(X_{sat} - D) = -\frac{d}{dt}D$$

Integration

The first ODE may be solved by *separation of variables* for the initial condition $L(t=0)=L_0$.

Analytical solution

Integration

The first ODE may be solved by *separation of variables* for the initial condition $L(t = 0) = L_0$.

$$
\begin{aligned}
\frac{d}{dt} L &= -k_d \cdot L \\
\int \frac{1}{L} \cdot dL &= \int -k_d \cdot dt \\
ln(L) &= -k_d \cdot t + C \\
L &= exp(-k_d \cdot t + C) = exp(-k_d \cdot t) \cdot C_2 \\
L &= L_0 \cdot exp(-k_d \cdot t)
\end{aligned}
$$

Analytical solution

Integration

Integration of the second ODE is much harder.

Insert for L the solution just derived:

$$\begin{aligned}\frac{d}{dt}D &= k_d \cdot L - k_a \cdot D \\ &= k_d \cdot L_0 \cdot exp(-k_d \cdot t) - k_a \cdot D\end{aligned}$$

Variables can’t be separated (t and D appear as a sum). Substitution doesn’t work either.

→ One must use the method of the *integrating factor*

Integration

Re-order terms:

$$\frac{d}{dt}D = k_d \cdot L_0 \cdot exp(-k_d \cdot t) - k_a \cdot D$$

$$\frac{d}{dt}D + k_a \cdot D = k_d \cdot L_0 \cdot exp(-k_d \cdot t)$$

Multiply by the integrating factor. In our case, a suitable factor is $exp(k_a \cdot t)$ yielding:

$$\begin{aligned} e^{k_a \cdot t} \cdot \frac{d}{dt}D + e^{k_a \cdot t} \cdot k_a \cdot D &= e^{k_a \cdot t} \cdot k_d \cdot L_0 \cdot e^{-k_d \cdot t} \\ &= k_d \cdot L_0 \cdot e^{(k_a - k_d) \cdot t} \end{aligned}$$

Analytical solution

Integration

The new expression looks even more complicated. But now we can apply the product rule

$$v \cdot \frac{d}{dt} u + u \cdot \frac{d}{dt} v = \frac{d}{dt}(u \cdot v)$$

to the left hand side. Also making use of the chain rule

$$\frac{d}{dt}(e^{k_a \cdot t}) = k_a \cdot e^{k_a \cdot t}$$

we get:

$$\frac{d}{dt}\left(D \cdot e^{k_a \cdot t}\right) = k_d \cdot L_0 \cdot e^{(k_a - k_d) \cdot t}$$

Analytical solution

Integration

Separation of variables and integration yields:

$$\int d\left(D \cdot e^{k_a \cdot t}\right) = k_d \cdot L_0 \cdot \int e^{(k_a - k_d)\cdot t} \cdot dt$$

$$\begin{aligned} D \cdot e^{k_a \cdot t} &= k_d \cdot L_0 \cdot e^{(k_a - k_d)\cdot t} \cdot \frac{1}{k_a - k_d} + C \\ &= \frac{k_d \cdot L_0}{k_a - k_d} \cdot e^{(k_a - k_d)\cdot t} + C \end{aligned}$$

With the initial condition $D(t = 0) = D_0$ we get:

$$C = D_0 - \frac{k_d \cdot L_0}{k_a - k_d}$$

We can now solve for D.

Integration

The analytical solutions of Eq. 3 and 4 are:

$$L = L_0 \cdot exp(-k_d \cdot t) \tag{5}$$

$$D = \frac{k_d \cdot L_0}{k_a - k_d} \cdot \left(e^{-k_d \cdot t} - e^{-k_a \cdot t} \right) + D_0 \cdot e^{-k_a \cdot t} \tag{6}$$

Integration

The analytical solutions of Eq. 3 and 4 are:

$$L = L_0 \cdot exp(-k_d \cdot t) \tag{5}$$

$$D = \frac{k_d \cdot L_0}{k_a - k_d} \cdot \left(e^{-k_d \cdot t} - e^{-k_a \cdot t} \right) + D_0 \cdot e^{-k_a \cdot t} \tag{6}$$

What is the value of analytical solutions in modern times?

Analytical solution

Integration

The analytical solutions of Eq. 3 and 4 are:

$$L = L_0 \cdot exp(-k_d \cdot t) \tag{5}$$

$$D = \frac{k_d \cdot L_0}{k_a - k_d} \cdot \left(e^{-k_d \cdot t} - e^{-k_a \cdot t}\right) + D_0 \cdot e^{-k_a \cdot t} \tag{6}$$

What is the value of analytical solutions in modern times?

- Exact reference for approximate numerical solutions
- Fast calculations

Predicted dynamics

Excersise

Plot the evolution of L, D and X for a period of 5 days. At $t = 0$ let L=10 mg/l and assume saturation with resp. to $\mathrm{O_2}$. Set the decay rate to 0.5 d^{-1} and the aeration rate to 1.8 d^{-1}. Water temperature is 12 $^\circ$C.

The $\mathrm{O_2}$ saturation level (mg/l) can be calculated from temperature T ($^\circ$C) using the Elmore & Hayes (1960) formula.

$$X_{sat} = 14.652 - 0.41022 \cdot T + 0.007991 \cdot T^2 - 7.7774\text{e-}05 \cdot T^3$$

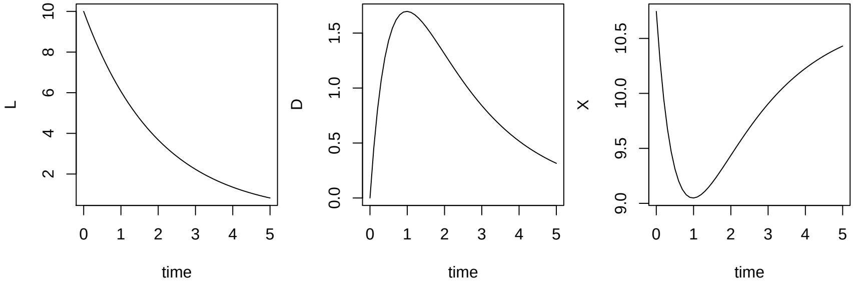
Predicted dynamics

```r
# Analytical solutions
L= function (t, p) {
  with(p,  L0 * exp(-kd * t) )
}
D= function (t, p) {
  with(p,  kd * L0 / (ka - kd) * (exp(-kd * t) -
           exp(-ka * t)) + D0 * exp(-ka * t) )
}
# O2 saturation as a function of temperature
X_sat= function(T) { 14.652 - 0.41022*T + 0.007991*T^2 - 7.7774e-5*T^3 }
```

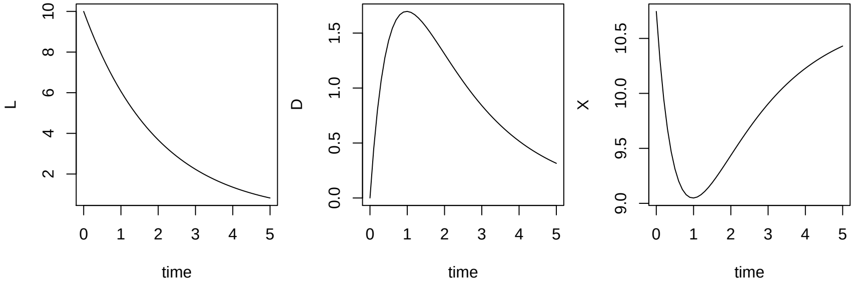
```r
p= list(D0= 0, L0= 10, ka= 1.8, kd= 0.5, temp=12)          # Parameters
time=seq(0, 5, 0.1)                                        # Times of interest

layout(matrix(1:3,ncol=3))
plot(time, L(time, p), type="l", ylab="L")                 # BOD
plot(time, D(time, p), type="l", ylab="D")                 # O2-Deficit
plot(time, X_sat(p$temp)-D(time, p), type="l", ylab="X")   # O2-Conc.
```

Predicted dynamics



Predicted dynamics



- What values take the state variables after infinite time?
- How to interpret the extreme values?
- How to compute the minimum $\mathrm{O_2}$ level?

Finding the minimum of $\mathrm{O_2}$

1. Set $dD/dt = 0$ and solve for time t
   $\rightarrow$ Time where the minimum occurs, t_x
2. Calculate D at $t = t_x$
3. Convert to $\mathrm{O_2}$ using $X_{min} = X_{sat} - D(t_x)$

Hint: Solving $dD/dt = 0$ for t becomes simpler if one divides by $k_a \cdot exp(-k_a \cdot t)$ after differentiation.

Analytical solution

Finding the minimum of $\mathrm{O_2}$

1. Set $dD/dt = 0$ and solve for time t
   $\rightarrow$ Time where the minimum occurs, t_x
2. Calculate D at $t = t_x$
3. Convert to $\mathrm{O_2}$ using $X_{min} = X_{sat} - D(t_x)$

Hint: Solving $dD/dt = 0$ for t becomes simpler if one divides by $k_a \cdot exp(-k_a \cdot t)$ after differentiation.

Time of occurrence of the minimum

$$t_x = \frac{1}{k_a - k_d} \cdot ln\left(\frac{k_a}{k_d} \cdot \left(1 - D_0 \cdot \frac{k_a - k_d}{k_d \cdot L_0}\right)\right) \tag{7}$$

Finding the minimum of O_2

The less elegant alternative

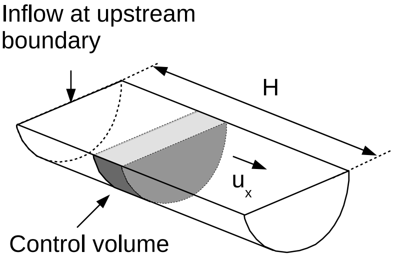
```r
# Function to be maximized with resp. to its first argument
D= function (t, p) {
  with(p,  kd * L0 / (ka - kd) * (exp(-kd * t) -
           exp(-ka * t)) + D0 * exp(-ka * t) )
}

# Parameters
p= list(D0= 0, L0= 10, ka= 1.8, kd= 0.5)

# Numerical optimization in 1 dimension
opt= optimize(f=D, interval=c(0,365), p, maximum=TRUE)

print(paste("Maximum deficit occurs after",round(opt$maximum*24,1),"hours"))
```

Application to a plug-flow system



Assumptions

- Transport controlled by advection
- Control volume is homogeneous internally but it doesn't mix with neighboring volumes (no longitudinal dispersion)
- Volume reaches a station h after travel time $t = h/u_x$ (u_x: velocity)
- Residence time in the reach: $T = H/u_x$

Application to a plug-flow system

The Streeter-Phelps equations (Eq. 5 und 6) are directly applicable to a moving control volume since time and space are related through velocity u_x.

Excercise

Plot the $\mathrm{O_2}$ concentrations for a river stretch of 100 km and find the location of the minimum. Let the BOD level at the discharge location (km 0) be 45 mg/l. Assume that the water is initially clean ($\mathrm{O_2}$ saturated). The rates of decay and aeration are 0.5 d^{-1} and 1.8 d^{-1}, respectively. Temperature is constant at 12 $^\circ$C. The average flow velocity is 0.75 m/s.

Application to a plug-flow system

```r
# Analytical solutions
L= function (t, p) {
  with(p,  L0 * exp(-kd * t) )
}
D= function (t, p) {
  with(p,  kd * L0 / (ka - kd) * (exp(-kd * t) -
           exp(-ka * t)) + D0 * exp(-ka * t) )
}
# O2 saturation as a function of temperature
X_sat= function(T) { 14.652 - 0.41022*T + 0.007991*T^2 - 7.7774e-5*T^3 }
```

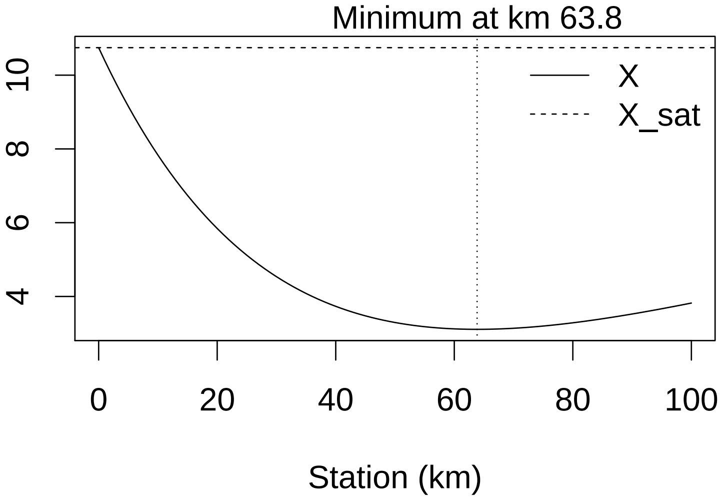
```r
p= list(D0= 0, L0= 45, ka= 1.8, kd= 0.5, temp=12)  # Initial values
h= seq(0, 100, 1)                                   # River stations (km)
u= 0.75 / 1000 * 86400                              # Velocities (m/s -> km/day)

t_x= function(p) {                                  # Timing of DO minimum (days)
  with(p,  1 / (ka - kd) * log( ka/kd * ( 1 - D0 * (ka - kd) / kd / L0)) )
}

plot(h, X_sat(p$temp)-D(t=h/u, p), type="l", xlab="Station (km)", ylab="")
abline(v=u*t_x(p), lty=3)
mtext(side=3, at=u*t_x(p), paste0("Minimum at km ",round(u*t_x(p),1)))
abline(h=X_sat(p$temp), lty=2)
legend("topright",bty="n",lty=c(1,2), legend=c("X","X_sat"))
```

Analytical solution

Application to a plug-flow system



Application to a plug-flow system

Discussion

- Why does the minimum of O_2 occur far downstream of the effluent?
- What happens if there is an initial oxygen deficit?

Analytical solution

Parameter estimation

How to obtain values for k_a and k_d?

Parameter estimation

How to obtain values for k_a and k_d?

- By calibration
  1. Measure at several locations along a river stretch, or
  2. Sample a particular volume at different times (drifting boat)
- There are empirical formulas for k_a (e.g. using depth and velocity as predictors)
- Monitor decay rate k_d under lab conditions (e.g. O_2 consumption)

Outline

Introduction

Analytical solution

Numerical solution

Model extension: Oxygen limitation

Further possible extensions

Spatially distributed model (1D)

Outlook

Why?

- Numerical solutions come with less restrictions
  - time-varying external forcings (temperature, loading)
  - arbitrary initial conditions (e.g. spatially variable)

Why?

- Numerical solutions come with less restrictions
  - time-varying external forcings (temperature, loading)
  - arbitrary initial conditions (e.g. spatially variable)
- For extended models
  - analytical solutions can be hard to find
  - a closed-form solutions might not exist at all

Using Eq. 1 & 2

Estimation of the stoichiometric factor *s*

Numerical solution
Using Eq. 1 & 2

Estimation of the stoichiometric factor *s*

1. Use carbon as the base element for organic matter. Then, a typical unit for Z is mg C_{org}/l.
2. Reaction (e.g. Chen et al. (1996), Marine Chemistry 54, 179-190)

   $(CH_2O)_{106}(NH_3)_{16}(H_3PO_4) + 138\ O_2 \rightarrow$
   $106\ CO_2 + 16\ HNO_3 + H_3PO_4 + 122\ H_2O$
3. Estimate of s in units of (g X / g Z) i.e. (g O_2 / g C_{org}):

   $$s \approx \frac{138 \cdot 32}{106 \cdot 12}$$

   (32 and 12 are molar masses of O_2 and C)

   Real-world values may be lower (z.B. doi:10.1016/j.ecolmodel.2005.04.016)

Direct implementation

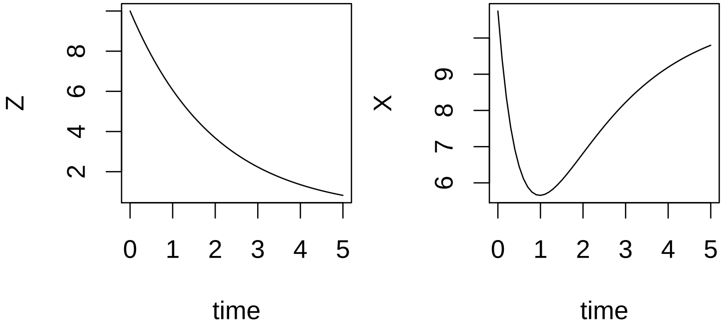
```r
# Returns the vector of derivatives (wrapped into a list)
model= function(t, y, p) {
  list(c(
    Z= -p$kd * y[["Z"]],
    X= -p$kd * y[["Z"]] * p$s + p$ka * (X_sat(p$temp) - y[["X"]])
  ))
}
```

```r
# O2 saturation as a function of temperature
X_sat= function(T) { 14.652 - 0.41022*T + 0.007991*T^2 - 7.7774e-5*T^3 }
# Parameters, initial values, and times of interest
p=     list(ka= 1.8, kd= 0.5, s=3, temp=12)
y0=    c(Z=10, X=X_sat(p$temp))
times= seq(0, 5, 0.1)
```

```r
library(deSolve)
res= lsoda(y=y0, times=times, func=model, parms=p)
if (attr(res,which="istate",exact=TRUE)[1] != 2) stop("Integration failed.")
```

```r
# Plot concentrations
layout(matrix(1:2,ncol=2))
res= as.data.frame(res)
for (item in c("Z","X"))
  plot(res$time, res[,item], type="l", xlab="time", ylab=item)
```

Direct implementation



Implementation in matrix notation

Motivation

$$\frac{d}{dt}Z = -k_d \cdot Z \qquad (1, \text{rep.})$$

$$\frac{d}{dt}X = -k_d \cdot Z \cdot s + k_a \cdot (X_{sat} - X) \qquad (2, \text{rep.})$$

- Write identical terms (*process rates*) only once
  → improved readability
  → faster computation
  → code is easier to maintain
- Better performance due to vectorization (eventually)

Implementation in matrix notation

$$\frac{d}{dt}Z = -k_d \cdot Z \qquad \text{(1, rep.)}$$

$$\frac{d}{dt}X = -k_d \cdot Z \cdot s + k_a \cdot (X_{sat} - X) \qquad \text{(2, rep.)}$$

Derivatives

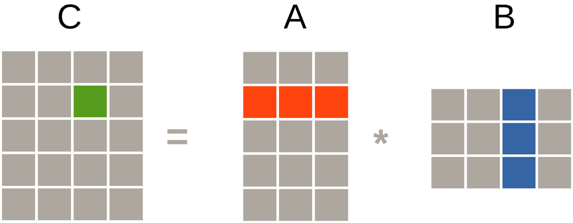
Stoichiometry matrix (transp.)

Process rates

$$\begin{bmatrix} \frac{d}{dt}Z \\ \frac{d}{dt}X \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ -s & 1 \end{bmatrix} \cdot \begin{bmatrix} k_d \cdot Z \\ k_a \cdot (X_{sat} - X) \end{bmatrix}$$

Implementation in matrix notation

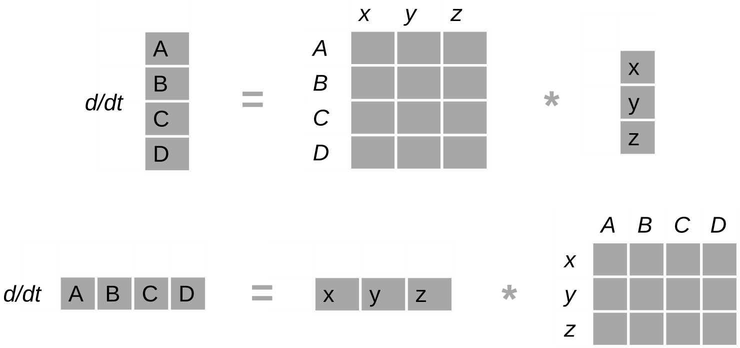
Matrix multiplication



$$C\,[i,k] = \sum\left(A\,[i,]\cdot B\,[,k]\right)$$

Implementation in matrix notation

Two equivalent ways of multiplication



- State variables: *A*, *B*, *C*, *D*; Process rates: *x*, *y*, *z*
- Note the different layout of the stoichiometry matrix!

Implementation in matrix notation

```r
# Vector of process rates
rates= function(y, p) {
  c( decay= p$kd * y[["Z"]],
     aerat= p$ka * (X_sat(p$temp) - y[["X"]])
  )
}
# Stoichiometry matrix
# columns: Z, X
stoix= function(y, p) {
  rbind(
    decay=c( -1, -p$s),
    aerat=c(  0,    1)
  )
}
# Vectors of derivatives and process rates (wrapped into a list)
model= function(t, y, p) {
  return(list( derivs= t(stoix(y,p)) %*% rates(y,p), rates= rates(y,p) ))
}
```

```r
# O2 saturation as a function of temperature
X_sat= function(T) { 14.652 - 0.41022*T + 0.007991*T^2 - 7.7774e-5*T^3 }
# Parameters, initial values, and times of interest
p=     list(ka= 1.8, kd= 0.5, s=3, temp=12)
y0=    c(Z=10, X=X_sat(p$temp))
times= seq(0, 5, 0.1)
```

Implementation in matrix notation

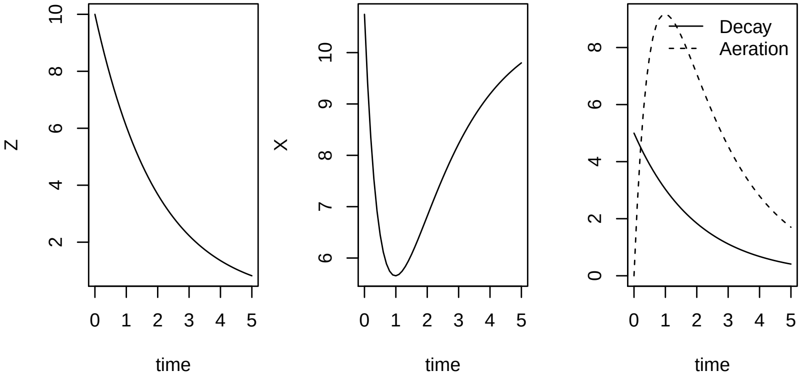
... Continued ...

```r
library(deSolve)
res= lsoda(y=y0, times=times, func=model, parms=p)
if (attr(res,which="istate",exact=TRUE)[1] != 2) stop("Integration failed.")
```

```r
# Plot concentrations
layout(matrix(1:3,ncol=3))
res= as.data.frame(res)
for (item in c("Z","X"))
  plot(res$time, res[,item], type="l", xlab="time", ylab=item)
# Also plot process rates
plot(range(res$time), range(c(res$rates.decay, res$rates.aerat)), type="n",
  xlab="time",ylab="")
lines(res$time, res$rates.decay, lty=1)
lines(res$time, res$rates.aerat, lty=2)
legend("topright", bty="n", lty=c(1,2), legend=c("Decay","Aeration"))
```

Numerical solution

Implementation in matrix notation



Outline

Introduction

Analytical solution

Numerical solution

Model extension: Oxygen limitation

Further possible extensions

Spatially distributed model (1D)

Outlook

Results for increased loading

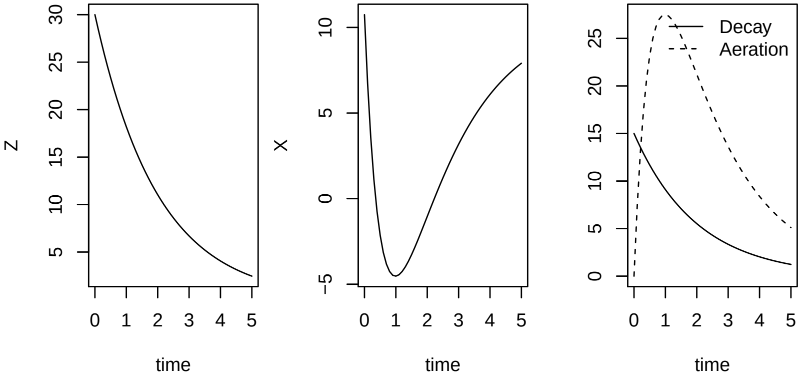
How does the O_2 level respond to increased organic loading?

Modify the initial concentrations as follows:

```
# Increased organic load
y0=   c(Z=30, X=X_sat(p$temp))
```

Model extension: Oxygen limitation

Results for increased loading



O_2 limited degradation

Michaelis-Menten model

$$v = v_{max} \cdot \frac{S}{h + S}$$

v	Reaction velocity
v_{max}	Maximum (unlimited) v
S	Concentration of substrate
h	Half-saturation constant

O_2 limited degradation

Michaelis-Menten model

$$v = v_{max} \cdot \frac{S}{h + S}$$

v	Reaction velocity
v_{max}	Maximum (unlimited) v
S	Concentration of substrate
h	Half-saturation constant

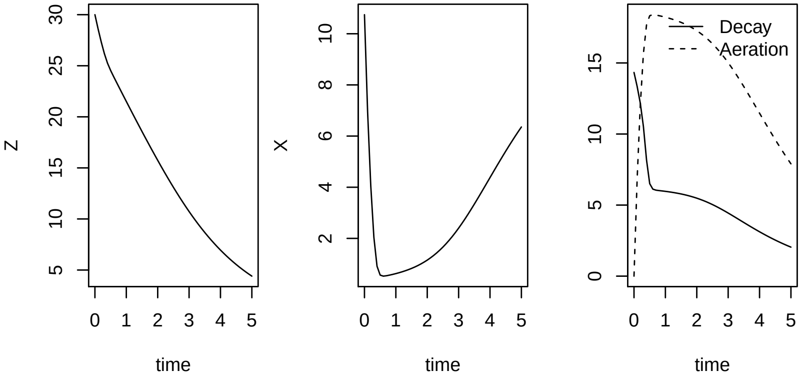
Required adaption to process rates and parameters:

```
# New parameter: Half-saturation constant
p$hx= 0.5

# Returns the vector of process rates
rates= function(y, p) {
  c( decay= p$kd * y[["Z"]] * y[["X"]] / (y[["X"]] + p$hx),
     aerat= p$ka * (X_sat(p$temp) - y[["X"]])
  )
}
```

Model extension: Oxygen limitation

O_2 limited degradation



Outline

Introduction

Analytical solution

Numerical solution

Model extension: Oxygen limitation

Further possible extensions

Spatially distributed model (1D)

Outlook

Wish-list

- Inclusion of anaerobic processes

Further possible extensions

Wish-list

- Inclusion of anaerobic processes
- Distinction between O_2 consumption from oxidation of carbon (CBOD) and ammonium (NBOD)

Wish-list

- Inclusion of anaerobic processes
- Distinction between O_2 consumption from oxidation of carbon (CBOD) and ammonium (NBOD)
- O_2 dynamics due to algal production / respiration

Wish-list

- Inclusion of anaerobic processes
- Distinction between $\mathrm{O_2}$ consumption from oxidation of carbon (CBOD) and ammonium (NBOD)
- $\mathrm{O_2}$ dynamics due to algal production / respiration
- Sedimentation of organic matter; Processes at sediment-water interface

Further possible extensions

Wish-list

- Inclusion of anaerobic processes
- Distinction between O_2 consumption from oxidation of carbon (CBOD) and ammonium (NBOD)
- O_2 dynamics due to algal production / respiration
- Sedimentation of organic matter; Processes at sediment-water interface
- Distinction between dissolved and particulate matter

Wish-list

- Inclusion of anaerobic processes
- Distinction between O_2 consumption from oxidation of carbon (CBOD) and ammonium (NBOD)
- O_2 dynamics due to algal production / respiration
- Sedimentation of organic matter; Processes at sediment-water interface
- Distinction between dissolved and particulate matter
- Bacteria biomass as a state variable

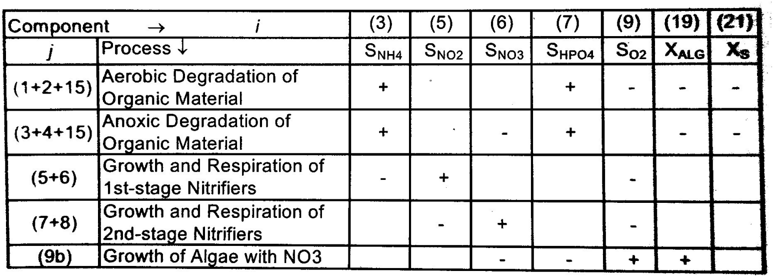
Further possible extensions

Wish-list

- Inclusion of anaerobic processes
- Distinction between O_2 consumption from oxidation of carbon (CBOD) and ammonium (NBOD)
- O_2 dynamics due to algal production / respiration
- Sedimentation of organic matter; Processes at sediment-water interface
- Distinction between dissolved and particulate matter
- Bacteria biomass as a state variable
- Improved transport model (dispersion, non-uniform or unsteady flow)
- ...

Further possible extensions

Example of an enhanced version

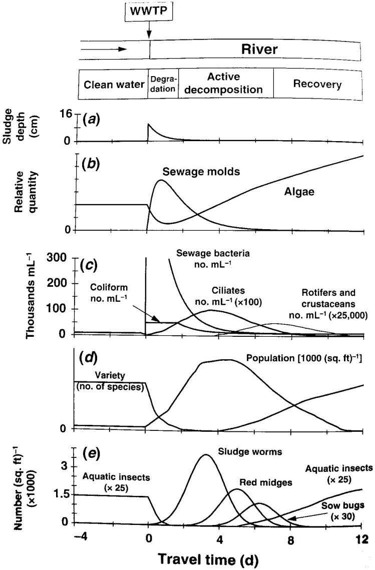


Component →	*i*	(3)	(5)	(6)	(7)	(9)	(19)	(21)
j	Process ↓	S_{NH4}	S_{NO2}	S_{NO3}	S_{HPO4}	S_{O2}	X_{ALG}	X_S
(1+2+15)	Aerobic Degradation of Organic Material	+			+	-	-	-
(3+4+15)	Anoxic Degradation of Organic Material	+		-	+		-	-
(5+6)	Growth and Respiration of 1st-stage Nitrifiers	-	+			-		
(7+8)	Growth and Respiration of 2nd-stage Nitrifiers		-	+		-		
(9b)	Growth of Algae with NO3			-	-	+	+	

From Reichert et al. (2001): River water quality model No. 1, IWA publishing

Further possible extensions

Actual complexity



From Chapra, S. (1997): Surface Water Quality Modeling, McGraw-Hill

Outline

Introduction

Analytical solution

Numerical solution

Model extension: Oxygen limitation

Further possible extensions

Spatially distributed model (1D)

Outlook

Generic PDE for a mobile species

$$\frac{\partial c}{\partial t} = \underbrace{D_x \cdot \frac{\partial^2 c}{\partial x^2}}_{\text{Dispers.}} - \underbrace{u_x \cdot \frac{\partial c}{\partial x}}_{\text{Advect.}} + \underbrace{R}_{\text{React.}}$$

c	Concentration (M/L^3)
x	Spatial coordinate (L)
u_x	Average velocity (L/T)
D_x	Longitudinal dispersions coeff. (L^2/T)

Spatially distributed model (1D)

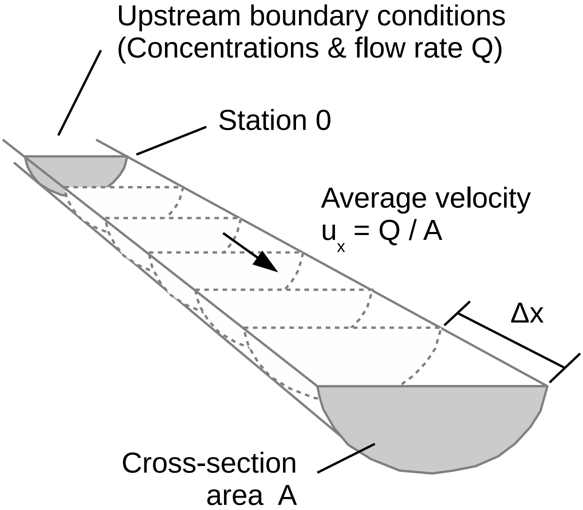
Method-of-lines approach

$$\frac{\partial c}{\partial t} = \underbrace{D_x \cdot \frac{\partial^2 c}{\partial x^2}}_{\text{Dispers.}} - \underbrace{u_x \cdot \frac{\partial c}{\partial x}}_{\text{Advect.}} + \underbrace{R}_{\text{React.}}$$

- Discretize the x-axis (but not the time axis)
  → Sub-divide reach into boxes
- Replace spatial derivatives by finite differences
  → Turns PDE problem into ODE problem
- Select suitable ODE solver + settings
  (e.g. structure of Jacobian)

Spatially distributed model (1D)

Method-of-lines approach



Upstream boundary conditions
(Concentrations & flow rate Q)

Station 0

Average velocity
$u_x = Q / A$

Δx

Cross-section
area A

Spatially distributed model (1D)

Method-of-lines approach

$$\frac{\partial c}{\partial t} \quad = \quad D_x \cdot \frac{\partial^2 c}{\partial x^2} \quad - \quad u_x \cdot \frac{\partial c}{\partial x} \quad + \quad R$$

Index of the box indicated by subscript *i*

$$\frac{dc_i}{dt} \quad = \quad D_x \cdot \frac{\Delta}{\Delta x}\left(\frac{\Delta c_i}{\Delta x}\right) \quad - \quad u_x \cdot \frac{\Delta c_i}{\Delta x} \quad + \quad R_i$$

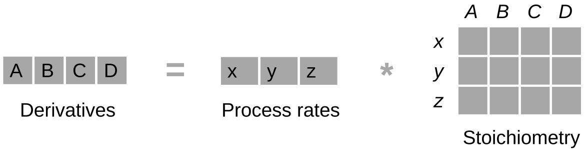
Expanded terms ($i - 1$: upstream box, $i + 1$: downstream box)

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

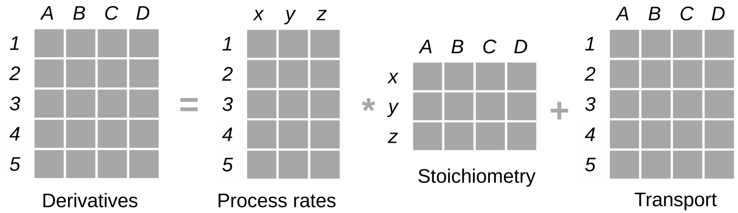
Spatially distributed model (1D)

Model in matrix notation

0-dimensional case (Species: *A*–*D*; Process rates: *x*–*z*)



1-dimensional case (boxes 1...5; invariant stoichiometry)



Simplistic river quality model

Considered processes

- Aerobic degradation in water
  → Process 1: Carbon oxidation
  → Process 2: Nitrification
- Aeration
- Advective transport only
  (to demonstrate numerical diffusion)

Spatially distributed model (1D)

Simplistic river quality model

Considered processes

- Aerobic degradation in water
  → Process 1: Carbon oxidation
  → Process 2: Nitrification
- Aeration
- Advective transport only
  (to demonstrate numerical diffusion)

Simulated species

- Organic carbon (OC), Oxygen (O2), Ammonium-N (NH4), Nitrate-N (NO3)
- Molar concentrations
  → simpler stoichiometry
  → unambiguous (e.g. NH4 == NH4-N)

Simplistic river quality model

```
# Molar masses of the species’ reference elements (i.e. C, O2, N)
molm= list(C=12, O2=32, N=14)

# Definition of parameters
p= list(
  kd=2/86400,             # Rate of decay (1/s)
  kn=0.1/86400,           # Rate of nitrification (1/s)
  ka=2/86400,             # Rate of aeration (1/s)
  s_O2_C=1/1,             # O2 consumed in oxidation of C (mol/mol)
  s_O2_N=2/1,             # O2 consumed in oxidation of NH4-N (mol/mol)
  s_N_C=16/106,           # N:C ratio for OC (mol/mol)
  hd=2/molm$O2,           # Half-saturation conc. of O2 (mmol/L) for decay
  hn=5/molm$O2,           # Half-saturation conc. of O2 (mmol/L) for nitrification
  O2sat=10/molm$O2,       # Saturation level of O2 at fixed temp. (mmol/L)
  nx=1000,                # Number of boxes
  dx=100,                 # Length of single box (m)
  u=0.5)                  # Velocity (x-section average; m/s)

# Concentrations of all species will be stored in a single vector
# --> A named list of indices allows for convenient and fast access
ispec= list(OC= 1:p$nx, O2= p$nx+(1:p$nx),
  NH4= 2*p$nx+(1:p$nx), NO3= 3*p$nx+(1:p$nx))

# Initialization of concentrations (all zero, except for O2)
y0= double(length(ispec)*p$nx)
y0[ispec$O2]= p$O2sat
```

Spatially distributed model (1D)

Simplistic river quality model

... Continued ...

```r
# Times of interest
# Note: Need to consider the Courant number when setting the time step
times=seq(from=0, to=7*86400, by=p$dx/p$u)

# Definition of boundary conditions (upstream concentrations)
# Assumption: Waste-water is discharged into stream on day 2
bcond= list(
  OC= function(t) {ifelse(t>=86400 && t<=2*86400, 10/molm$C, 0)},
  O2= function(t) {p$O2sat},
  NH4= function(t) { 0 },
  NO3= function(t) { 0 }
)
```

Spatially distributed model (1D)

Simplistic river quality model

... Continued ...

```r
  # Definition of the ODE model
model= function(t, y, p, ispec) {
  # Matrix of processes (boxes x processes)
  rates= cbind(
    degra= p$kd * y[ispec$OC] * y[ispec$O2]/(y[ispec$O2]+p$hd),
    nitri= p$kn * y[ispec$NH4] * y[ispec$O2]/(y[ispec$O2]+p$hn),
    aerat= p$ka * (p$O2sat - y[ispec$O2])
  )
  # Stoichiometry matrix (processes x species)
  stoix= rbind(
    #        OC           O2        NH4  NO3
    degra= c(-1,  -p$s_O2_C,  p$s_N_C,   0),
    nitri= c( 0,  -p$s_O2_N,       -1,   1),
    aerat= c( 0,          1,        0,   0)
  )
  # Matrix of advection terms (boxes x species)
  tran= cbind(
    -p$u * diff(c(bcond$OC(t),y[ispec$OC])) / p$dx,
    -p$u * diff(c(bcond$O2(t),y[ispec$O2])) / p$dx,
    -p$u * diff(c(bcond$NH4(t),y[ispec$NH4])) / p$dx,
    -p$u * diff(c(bcond$NO3(t),y[ispec$NO3])) / p$dx
  )
  # Matrix of derivatives (boxes x species)
  return( list(rates %*% stoix + tran) )
}
```

Simplistic river quality model

... Continued ...

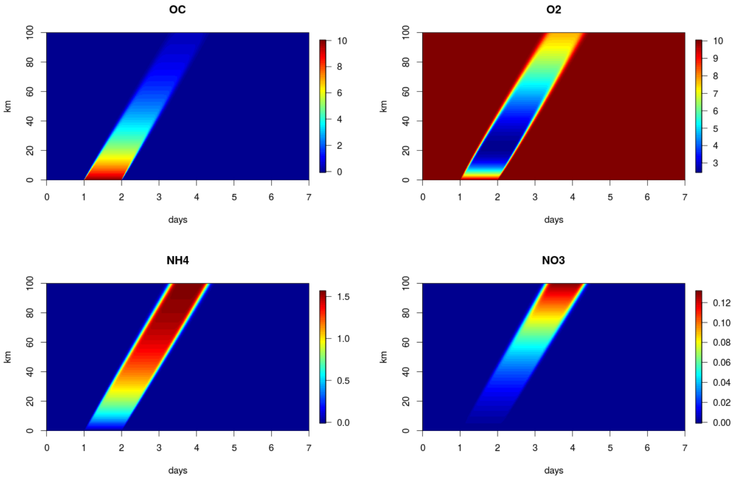
```r
# Integration (Note: algorithm should account for banded Jacobian matrix)
library(deSolve)
out= ode.1D(y=y0, times=times, func=model, parms=p, nspec=length(ispec),
  dimens=p$nbox, ispec=ispec)
# Pragmatic handling of numerical artefacts
out[out < 1e-12]= 0

# Prepare data for plotting (concentrations stored as list of matrices,
# units converted from mmol/L to mg/L)
days= out[,1] / 86400
km= seq(from=0.5*p$dx, by=p$dx, length.out=p$nx) / 1000
conc= list(OC= out[,1+ispec$OC]*molm$C, O2= out[,1+ispec$O2]*molm$O2,
  NH4= out[,1+ispec$NH4]*molm$N, NO3= out[,1+ispec$NO3]*molm$N)

# Plot all concentrations  (fields::image.plot is suitable because it
# doesn’t interpolate, has a legend included, and it works with layout)
library(fields)
layout(matrix(1:(ceiling(length(ispec)/2)*2), ncol=2, byrow=TRUE))
for (n in names(ispec))
  image.plot(x=days, y=km, z=conc[[n]], main=n, useRaster=TRUE, legend.mar=10)
```

Spatially distributed model (1D)

Simplistic river quality model



Outline

Introduction

Analytical solution

Numerical solution

Model extension: Oxygen limitation

Further possible extensions

Spatially distributed model (1D)

Outlook

Outlook

When models become complex...

- Direct coding of the stoichiometry matrix is ugly
  → Hard to write/read/debug
- Computation times are often critical
  → Fortran or C
- Quick-and-dirty hacks are impossible to maintain
  → Modularity, encapsulation, documentation, ...

→ Automatic **code generation** becomes attractive