

Estimating pH in a 1D Reaction Transport Model in R

Reader Accompanying the Course Reaction Transport Modelling in the Hydrosphere

Karline Soetaert and Lubos Polerecky, Utrecht University

April 2021

Abstract

In two other readers, we have shown how to implement in R (i) a model that can predict the variation of pH in time due to reactions that add/remove chemical species capable of accepting/donating protons, and (ii) a reaction-transport model that can predict the variation of ammonia, nitrate and oxygen along an estuary. Here we combine and expand the two models to illustrate how to implement in R a reaction-transport model that can predict the spatial variation of pH. The overall approach involves two steps. First, the **ReactTran** package is used to calculate the steady-state distribution of the lump-sum species that include total ammonia (ΣNH_4), total nitrate (ΣHNO_3), total dissolved inorganic carbon (ΣCO_2), and (titration) proton alkalinity (TA). Then, based on these profiles, the **AquaEnv** package is used to calculate the steady-state distribution of pH.

Before you start reading

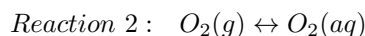
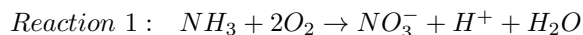
The basic principles that we are going to expand upon here are explained in the following readers:

- Local Equilibrium Chemistry in R — part III: Mixed Equilibrium – Irreversible Reactions
- Anoxia in the Scheldt estuary

It is important that you have fully understood the material covered in those readers before you start reading the following text. Type `RTMexercise("equilibriumOMD")` and `RTMexercise("estuaryAnoxia")` in the R-console to view the readers.

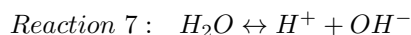
Modelled chemical reactions

In the simplified model of the Scheldt estuary, we assumed that changes in water chemistry are only driven by two processes: nitrification (Reaction 1) and the air-water exchange of oxygen (Reaction 2).



We retain this assumption here to keep the model simple but illustrative.

In addition to the above processes, which are assumed to be relatively slow, we assume the following fast reversible reactions:



These reactions need to be considered for two reasons: (i) Reactions 3 and 4 are affected by Reaction 1, and (ii) Reactions 5–7 involve chemical species with the most important contributions (in terms of concentrations) to alkalinity, which is a central lump-sum species in models involving pH. The reasoning behind the second point is provided further down.

Lump-sum species

As a first step in solving for pH, we need to identify lump-sum species that are not affected by any of the fast reversible reactions 3–7. In class, you saw that they include the total dissolved inorganic carbon

$$\Sigma CO_2 = H_2CO_3 + HCO_3^- + CO_3^{2-},$$

and the total ammonia

$$\Sigma NH_4 = NH_3 + NH_4^+.$$

Additionally, they include the total nitrate

$$\Sigma HNO_3 = HNO_3 + NO_3^-,$$

and the total alkalinity (TA).

With respect to TA , there are various possible definitions.¹ Here we use the proton alkalinity:

$$TA = 2[CO_3^{2-}] + [HCO_3^-] + [NH_3] + [OH^-] - [HNO_3] - [H^+].$$

This definition is very close to the titration alkalinity measured by chemists, and it is also used within the R-package **AquaEnv** that we will use below to calculate pH. We recommend that, before you continue reading, you verify that the lump-sum species ΣCO_2 , ΣNH_4 , ΣHNO_3 , and TA are *not* affected by any of the fast reversible Reactions 3–7.

Mass-balance equations for the lump-sum species

Assuming local equilibrium, the above analysis implies the following mass-balance equations for the lump-sum species:

$$\frac{d[O_2]}{dt} = Tran(O_2) + R_{aeration} - 2 \cdot R_{nitri} \quad (Eq. 1)$$

$$\frac{d[\Sigma HNO_3]}{dt} = Tran(\Sigma HNO_3) + R_{nitri} \quad (Eq. 2)$$

$$\frac{d[\Sigma NH_4]}{dt} = Tran(\Sigma NH_4) - R_{nitri} \quad (Eq. 3)$$

$$\frac{d[TA]}{dt} = Tran(TA) - 2 \cdot R_{nitri} \quad (Eq. 4)$$

$$\frac{d[\Sigma CO_2]}{dt} = Tran(\Sigma CO_2) \quad (Eq. 5)$$

¹The complete definition of TA is

$$TA = [HCO_3^-] + 2 \cdot [CO_3^{2-}] + [OH^-] + [B(OH)_4^-] + [HPO_4^{2-}] + 2 \cdot [PO_4^{3-}] + [H_3SiO_4^-] + 2 \cdot [H_2SiO_4^{2-}] \\ + [NH_3] + [HS^-] + 2 \cdot [S^{2-}] - [H^+] - [H_3PO_4] - [HSO_4^-] - [HF] - [HNO_3] - [HNO_2] - 2 \cdot [H_2SO_4].$$

Based on this definition, TA represents the proton deficit of the solution relative to a zero level of protons that is defined at pH of 4.5. For example, CO_3^{2-} can accept two protons (see Reactions 5 and 6), whereas HS^- can accept one proton ($H_2S \leftrightarrow HS^- + H^+$). Thus, these species contribute to TA by $2 \cdot [CO_3^{2-}]$ and $[HS^-]$, respectively. In contrast, H_2SO_4 can donate two protons ($H_2SO_4 \leftrightarrow SO_4^{2-} + 2H^+$), whereas HNO_3 can donate one proton (Reaction 4). Thus, their contributions to TA are $-2 \cdot [H_2SO_4]$ and $-[HNO_3]$, respectively.

In these equations, $Tran(X)$ refers to the transport term for species X , $R_{aeration}$ is the rate of air-water oxygen exchange, and R_{nitri} is the rate of nitrification. Note that in this simplified model, we assume that ΣCO_2 is only affected by transport, i.e., we ignore the air-water CO_2 exchange or any other process that may remove/add ΣCO_2 from/to the system.

We emphasise that pH *is* affected by nitrification, because nitrification consumes 2 moles of alkalinity per mole of total ammonia oxidised² (Reaction 1, Eq. 4). In contrast, pH is *not* affected by aeration, because aeration only affects O_2 , which is not part of TA .

Calculation of pH

In the reader entitled “Local Equilibrium Chemistry in R — part III: Mixed Equilibrium – Irreversible Reactions”, we have shown that, in an equilibrium, there is an intimate relationship between the alkalinity, pH , and the concentrations of the lump-sum species ΣCO_2 and ΣNH_4 (see Equations 5a–5f in that reader). This made it possible to calculate pH from the known values of ΣCO_2 , ΣNH_4 and alkalinity (see function `solveH` on p. 6 of that reader).

This conclusion is also valid for the proton alkalinity defined above, or for the complete TA defined in footnote 1. Thus, in analogy to the function `solveH`, we could construct a function that calculates pH from the value of TA and the concentrations of the lump-sum species ΣCO_2 , ΣNH_4 , ΣHNO_3 , etc. We are not going to do it here, however, because such function is already available. Specifically, the simplest (albeit not the fastest) way to estimate pH is to use the function `aquaenv` from the R-package *AquaEnv*.

The aquaenv function

An example of an output of the `aquaenv` function is shown below. By providing the total alkalinity and the concentrations of the lump-sum species as input, we get in return a lot of output, including the pH :

```
require(AquaEnv)
AE <- aquaenv(S=35, t=20, TA=1.3e-3,
              SumCO2=1.2e-3, SumNH4=0.001e-3, SumHNO3=0.003e-3)
names(AE)
```

## [1]	"S"	"t"	"p"	"T"
## [5]	"Cl"	"I"	"p"	"Pa"
## [9]	"d"	"density"	"SumCO2"	"SumNH4"
## [13]	"SumH2S"	"SumHNO3"	"SumHNO2"	"SumH3PO4"
## [17]	"SumSiOH4"	"SumBOH3"	"SumH2SO4"	"SumHF"
## [21]	"Br"	"ClConc"	"Na"	"Mg"
## [25]	"Ca"	"K"	"Sr"	"molal2molin"
## [29]	"free2tot"	"free2sws"	"tot2free"	"tot2sws"
## [33]	"sws2free"	"sws2tot"	"K0_CO2"	"K0_O2"
## [37]	"fCO2atm"	"fO2atm"	"CO2_sat"	"O2_sat"
## [41]	"K_W"	"K_HS04"	"K_HF"	"K_CO2"
## [45]	"K_HCO3"	"K_BOH3"	"K_NH4"	"K_H2S"
## [49]	"K_H3PO4"	"K_H2PO4"	"K_HPO4"	"K_SiOH4"
## [53]	"K_SiOOH3"	"K_HNO2"	"K_HNO3"	"K_H2SO4"
## [57]	"K_HS"	"Ksp_calcite"	"Ksp_aragonite"	"TA"
## [61]	"pH"	"fCO2"	"CO2"	"HCO3"
## [65]	"CO3"	"BOH3"	"BOH4"	"OH"

²Nitrification removes one mole of NH_3 and adds one mole of H^+ . Thus, because protons contribute negatively to TA , this translates to 2 moles of TA removed per mole of NH_3 in Reaction 1. Note that although Reaction 1 *adds* one mole of NO_3^- , this addition does *not* affect TA because the NO_3^- species is *not* part of the definition of TA !

```
## [69] "H3PO4"      "H2PO4"      "HPO4"      "PO4"
## [73] "SiOH4"      "SiOOH3"     "SiO2OH2"   "H2S"
## [77] "HS"         "S2min"      "NH4"       "NH3"
## [81] "H2SO4"      "HSO4"       "SO4"       "HF"
## [85] "F"          "HNO3"       "NO3"       "HNO2"
## [89] "NO2"        "omega_calcite" "omega_aragonite"
```

```
AE$pH
```

```
## [1] 7.880514
## attr(,"pH scale")
## [1] "free"
```

```
AE$density
```

```
## [1] 1024.763
## attr(,"unit")
## [1] "kg/m3"
```

Note that `aquaenv` requires the input concentrations to be specified in [mol/kg solution], and returns alkalinity in the same unit. Also note that the default concentrations for most lump-sum species are set to zero, except for those with a default value set to `NULL`, for which the concentrations are calculated based on salinity assuming standard seawater composition.³

It is useful to realise that although the total sulphate concentration in seawater ($\Sigma H_2SO_4 \approx 28 \text{ mM}$) is much greater than the total DIC concentration ($\Sigma CO_2 \approx 2 \text{ mM}$), including the contribution of sulphate leads only to minor differences in TA . This is because, for the seawater pH of around 8, the concentrations of the sulphate species that contribute to TA (H_2SO_4 and HSO_4^-) are *marginal* compared to the concentrations of the ΣCO_2 species (CO_3^{2-} and HCO_3^-). To verify this, we calculate TA for water with and without sulphate:

```
AE <- aquaenv(S=35, t=20, pH=8.2, SumCO2=2e-3,
              SumH2SO4=c(0, 28e-3)) # "seawater" without or with sulphate

out <- 1e3*data.frame(TA=AE$TA,                                # contributions due to:
                     HCO3=AE$HCO3, CO3=AE$CO3,              # carbonate ions
                     HSO4=AE$HSO4, H2SO4=AE$H2SO4,          # sulphate
                     BOH4=AE$BOH4,                            # borate
                     OH=AE$OH, H=rep(10^(-AE$pH),2))         # dissociated water

knitr::kable(out) # values in mmol/kg Liquid!
```

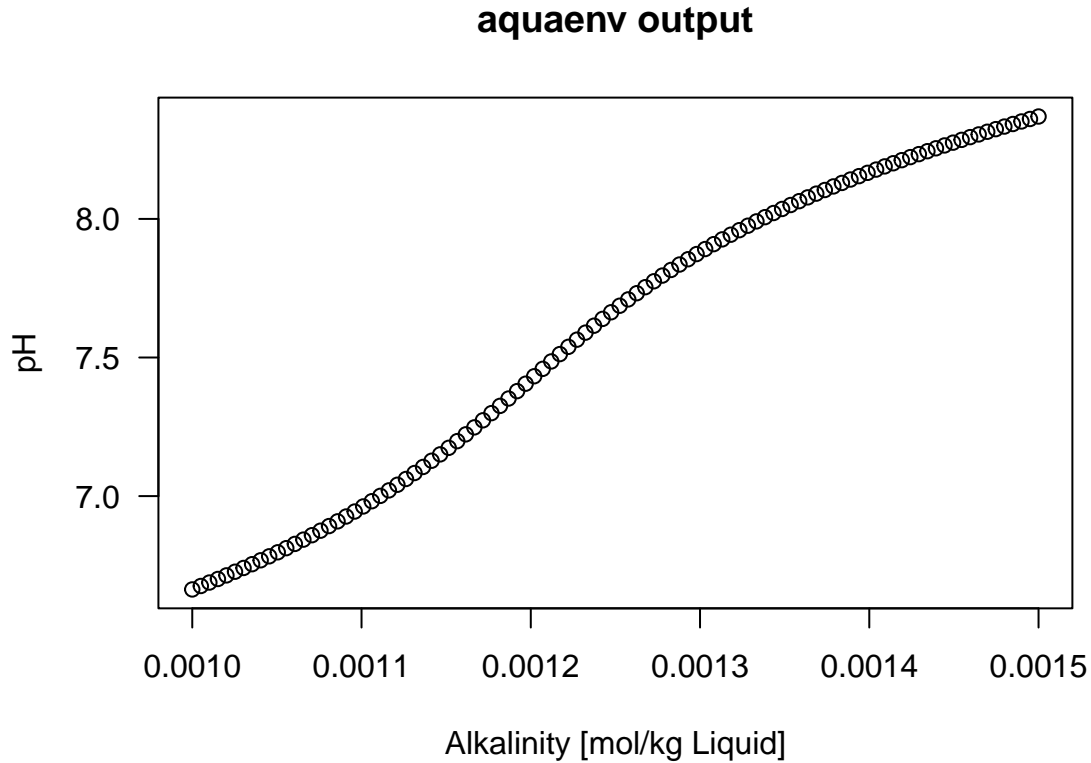
TA	HCO3	CO3	HSO4	H2SO4	BOH4	OH	H
2.353467	1.743232	0.2482037	0.0e+00	0	0.1077494	0.0060853	6.3e-06
2.292140	1.783144	0.2060617	1.5e-06	0	0.0919409	0.0049390	6.3e-06

The table illustrates why, in our simplified model, ΣCO_2 *must* be included but sulphate can be ignored. It also shows that the next most important contributions to TA in standard seawater are $B(OH)_4^-$ (determined by the salinity) and OH^- (due to dissociation of water; Reaction 7). However, their contribution is still marginal compared to that of carbonate ions. This marginal contribution of salinity to TA might suggest that salinity variations can be ignored in our model. However, because salinity affects the dissociation constants of the chemical species quite a bit, salinity variations *cannot* be ignored. Nevertheless, for the sake of simplicity, we will assume salinity to be constant (and marine) in our model.

The `aquaenv` function also accepts vectors as input:

³This applies, for example, to borate ($\Sigma H(BO)_4^-$). See `?aquaenv` for more information.

```
AE <- aquaenv(S=35, t=20, TA=seq(from=1.0e-3, to=1.5e-3, length.out=100),
             SumCO2=1.2e-3, SumNH4=0.001e-3, SumHNO3=0.003e-3)
plot(x=AE$TA, y=AE$pH, xlab="Alkalinity [mol/kg Liquid]", ylab="pH",
     main="aquaenv output", las=1)
```



The extended model of the Scheldt estuary

Here we reproduce the model in the reader entitled “Anoxia in the Scheldt estuary” and expand it by including ΣCO_2 and TA as state variables in addition to O_2 , ΣHNO_3 and ΣNH_4 . Note that because the modelled concentrations are in $mol\ m^{-3}$, we will need the water density (RHO) to convert them to $mol\ kg^{-1}\ Liquid$ when calculating the pH .

```
RHO <- aquaenv(S=35, t=20)$density
RHO
```

```
## [1] 1024.763
## attr(,"unit")
## [1] "kg/m3"
```

Model implementation

First, we define the model domain, state variables, and parameters, including the boundary conditions for the new state variables. The concentrations of ΣCO_2 and TA in the river were obtained from literature (Frankignoulle et al., 1996).

```

require(ReacTran) # package with solution methods

# model grid
Length <- 100000 # m
N <- 100 # - number of boxes
dx <- Length/N # m grid size
x <- seq(dx/2, by = dx, length.out = N) # m position of cells
Area <- 20000 # m2, cross-sectional area
day2sec <- 24*3600 # number of seconds in a day

# state variables; note two new ones!
SVnames <- c("Oxygen", "Nitrate", "Ammonia", "TA", "DIC")
state <- rep(0, times = length(SVnames)*N)

# model parameters
Sali <- 35
tC <- 20
pars <- c(
  riverO2 = 0.1, # river oxygen conc [mol/m3]
  seaO2 = 0.3, # marine oxygen conc [mol/m3]
  riverNit = 0.3, # river nitrate conc [mol/m3]
  seaNit = 0.05, # marine nitrate conc
  riverAmm = 0.1, # river ammonium conc
  seaAmm = 0.01, # marine ammonium conc
  depth = 10, # [m]
  v = 100/Area*day2sec, # advection velocity [m/d],
  Ddisp = 350*day2sec, # dispersion coefficient [m2/d], tidal dispersion
  rNitri = 0.1, # nitrification rate constant, [/d]
  ksO2 = 1e-3, # Monod ct for O2 lim of nitrification [mol/m3]
  piston = 1.0, # piston velocity [m/day]
  O2sat = 0.3, # saturated o2 concentration [mol/m3], solubility

  # EXTRA PARAMETERS for pH CALCULATION:
  seaDIC = 2.0, # boundary conc of DIC in sea [mol/m3]
  riverDIC = 5.3, # boundary conc of DIC in river [mol/m3]
  seaTA = 2.2, # boundary conc of alkalinity in sea [mol/m3]
  riverTA = 5.6, # boundary conc of alkalinity in river [mol/m3]
  RHO = aquaenv(S=Sali, t=tC)$density
)

```

Now, we define the model function, which implements the mass-balance equations 1–5. Note that we use an extra argument in the model function (`pHCalc`) that will allow us to toggle on and off the *pH* calculation. This is because the *pH* calculation from *AquaEnv* slows down the computation quite a bit. Thus, it is better *not* to estimate *pH* while the solver is trying to find the steady-state solution, but instead do this once the solution has been found. Also note that because tidal dispersion affects transport of all modelled species in the same way, we use the same parameters in the transport term for each state variable.⁴

```

# Model function
Scheldt1D <- function(t, C, pars,
  pHCalc = FALSE) { # set to TRUE if pH needs to be estimated
  with (as.list(pars),{

```

⁴This would not be the case if molecular diffusion were the dominant mode of transport. However, describing how important this issue is, and how to deal with it, goes beyond the scope of this reader.

```

Oxygen <- C[ 1 : N]
Nitrate <- C[ (N+1) : (2*N)]
Ammonia <- C[(2*N+1) : (3*N)]
TA <- C[(3*N+1) : (4*N)] # Note: extra state variables
DIC <- C[(4*N+1) : (5*N)] #

# transport terms
TranOxygen <- tran.1D(C=Oxygen, C.up=riverO2, C.down=seaO2,
                      D=Ddisp, v=v, dx=dx)
TranNitrate <- tran.1D(C=Nitrate, C.up=riverNit, C.down=seaNit,
                      D=Ddisp, v=v, dx=dx)
TranAmmonia <- tran.1D(C=Ammonia, C.up=riverAmm, C.down=seaAmm,
                      D=Ddisp, v=v, dx=dx)
TranTA <- tran.1D(C=TA, C.up=riverTA, C.down=seaTA,
                  D=Ddisp, v=v, dx=dx)
TranDIC <- tran.1D(C=DIC, C.up=riverDIC, C.down=seaDIC,
                  D=Ddisp, v=v, dx=dx)

# reactions
Nitrification <- rNitri * Ammonia * Oxygen/(Oxygen+ksO2)
Aeration <- -piston/depth * (Oxygen - O2sat)

# the rates of change = transport + net reaction
dNitrate.dt <- TranNitrate$dC + Nitrification
dAmmonia.dt <- TranAmmonia$dC - Nitrification
dOxygen.dt <- TranOxygen$dC - 2*Nitrification + Aeration
dTA.dt <- TranTA$dC - 2*Nitrification # TA dynamics
dDIC.dt <- TranDIC$dC # only transport

# calculate pH
if (pHCalc)
  pH <- aquaenv(S=Sali, t=tC, SumCO2=DIC/RHO, SumNH4=Ammonia/RHO,
                SumHNO3=Nitrate/RHO, TA=TA/RHO)$pH
else
  pH <- rep(0, times=N)

# output: the rates of change and pH
list(c(dOxygen.dt, dNitrate.dt, dAmmonia.dt, dTA.dt, dDIC.dt),
     pH=pH)
}

```

Model solution

Solving the model for the situation in the 1970s and 2000s is now done in two steps. First, we find the steady-state solution without calculating pH (`pHCalc=FALSE`). Subsequently, pH is estimated by calling the model function (`Scheldt1D`) with the input argument `C` set to the steady-state solution (`ScheldtXX$y`) and `pHCalc` set to `TRUE`.

```

# model parameters for the 2000s (default)
Scheldt00 <- steady.1D(y=state, parms=pars, func=Scheldt1D, positive=TRUE,
                      nspec=length(SVnames), dimens=N, names=SVnames,
                      pHCalc=FALSE)

```

```
Scheldt00$pH <- Scheldt1D(t=0, C=Scheldt00$y, pars, pHCalc=TRUE)$pH
```

```
# model parameters for the 1970s
```

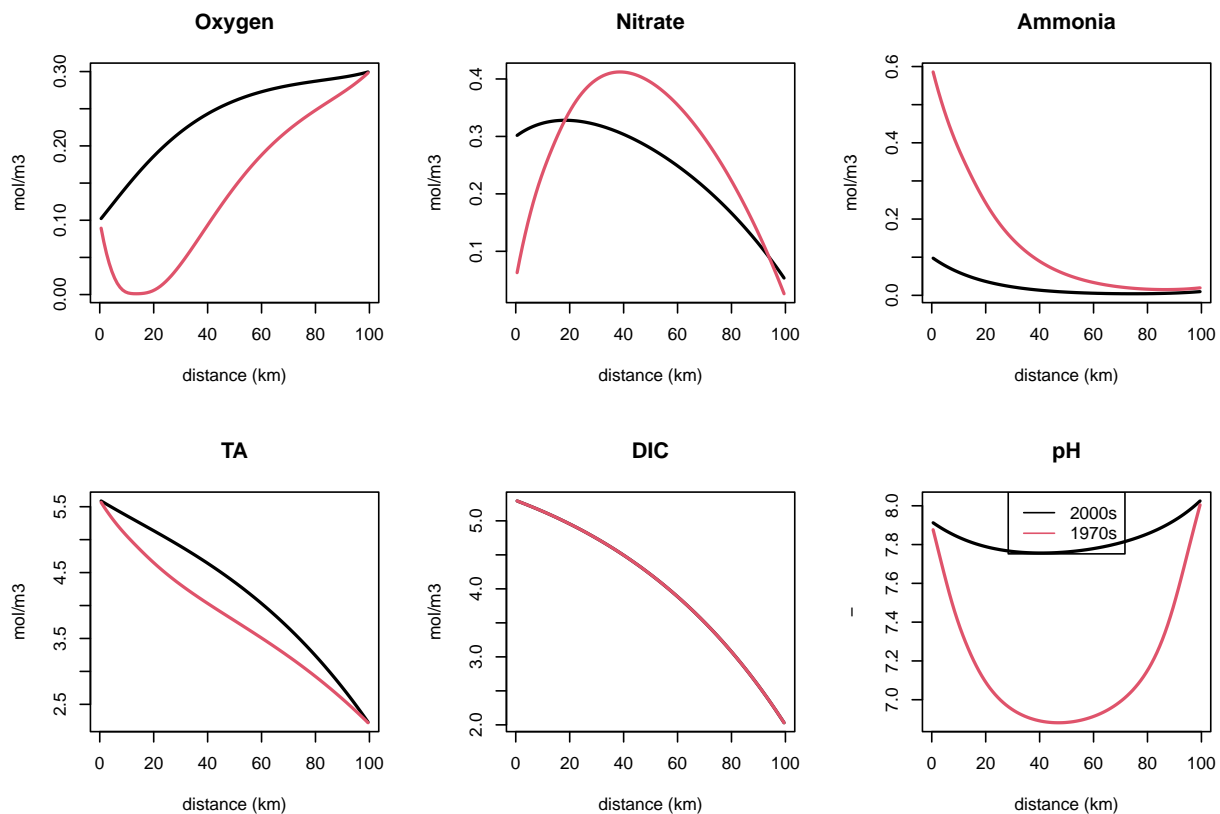
```
par70 <- pars
par70["riverNit"] <- 0.05 # river nitrate conc
par70["seaNit"] <- 0.02 # marine nitrate conc
par70["riverAmm"] <- 0.60 # river ammonium conc
par70["seaAmm"] <- 0.02 # marine ammonium conc
```

```
Scheldt70 <- steady.1D(y=state, parms=par70, func=Scheldt1D, positive=TRUE,
  nspec=length(SVnames), dims=N, names=SVnames,
  pHCalc=FALSE)
```

```
Scheldt70$pH <- Scheldt1D(t=0, C=Scheldt70$y, pars, pHCalc=TRUE)$pH
```

Results show quite a dramatic impact of nitrification on pH:

```
plot(Scheldt00, Scheldt70, grid=x/1000, lwd=2, lty=1,
  mfrow=c(2,3), ylab="mol/m3", xlab="distance (km)")
plot(Scheldt00, Scheldt70, grid=x/1000, lwd=2, lty=1,
  which="pH", mfrow=NULL, ylab="-", xlab="distance (km)")
legend("top", legend = c("2000s", "1970s"), lty = 1, col = 1:2)
```



References

- R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Soetaert Karline (2009). rootSolve: Nonlinear root finding, equilibrium and steady-state analysis of ordinary differential equations. R-package version 1.6
- Soetaert, Karline and Meysman, Filip, (2012). Reactive transport in aquatic ecosystems: Rapid model prototyping in the open source software R Environmental Modelling & Software, 32, 49-60.
- Andreas F. Hofmann, Karline Soetaert, Jack J. Middelburg and Filip J. R. Meysman, 2010. AquaEnv - an Aquatic acid-base modelling Environment in R. Aquatic Geochemistry. DOI: 10.1007/s10498-009-9084-1.
- Karline Soetaert, Andreas F. Hofmann, Jack J. Middelburg, Filip J.R. Meysman, Jim Greenwood, 2007. The effect of biogeochemical processes on pH. Marine Chemistry 105 30–51. DOI: 10.1016/j.marchem.2006.12.012
- Frankignoulle, M., I.Bourge, R. Wollast, 1996. Atmospheric CO₂ fluxes in a highly polluted estuary (the Scheldt). Limnol. Oceanogr., 41(2), 1996, 365–369. DOI: 10.4319/lo.1996.41.2.0365