````
---
title: "Ocean Alkalinity, Buffering and Biogeochemical Processes -
accompanying scripts"
author: "Karline Soetaert and Mathilde Hagens"
params:
  EVAL: no
output:
  pdf_document: default
  word_document: default
  html_document: default
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE)
```

# Preamble

These scripts were used for the calculations in the Middelburg et al (2020)
paper.

```{r, message=FALSE}
require(AquaEnv)
require(plot3D)
require(RColorBrewer)

darkcols <- brewer.pal(12,"Paired")
darkcols[11] <- grey(0.1)
palette(darkcols)

ColScheme    <- ramp.col(col = c("#1F78B4", "#FFFFCC", "#FC9272") , n =
100)
```

# Environmental settings

The default species composition as in Hagens and Middelburg (2016) is used.

```{r}

# Hagens and middelburg (2016):
SETTINGS <- list(SumCO2 = 2017e-6, SumBOH3 = 427.8e-6, SumH3PO4 = 0.5e-6,
                 SumSiOH4 = 7.4e-6, SumH2S = 2.3e-9, SumH2SO4 = 27930e-6,
                 SumHF = 67.6e-6, SumHNO3 = 5.2e-6, SumHNO2 = 0.1e-6,
                 SumH2O = 55.49, SumNH4 =0.3e-6, S = 34.617, t = 18.252, p
= 0)


Lumpsum <- data.frame(lumpsum = unlist(SETTINGS[c("SumCO2", "SumBOH3",
"SumH3PO4", "SumSiOH4",
                                "SumH2S", "SumH2SO4", "SumHF", "SumHNO3",
                                "SumHNO2", "SumH2O", "SumNH4")]))

expression <- list(expression(sum("CO"[2])), expression(sum("B(OH)"[3])),
                   expression(sum(paste("H"[3],"PO"[4]))),
expression(sum("Si(OH)"[4])),
                   expression(sum(paste("H"[2],"S"))),
expression(sum(paste("H"[2],"SO"[4]))),
                   expression(sum("HF")), expression(sum("HNO"[3])),
                   expression(sum("HNO"[2])),
expression(sum(paste("H"[2],"O"))),
                   expression(sum("NH"[4])))
Lumpsum$expression <- expression

# Redfield ratios
NC <- 16/106
PC <- 1/106

pH.seq <- seq(1, 14, length.out = 100)
````
````

```

# Chemical players

The various chemical species are inputted here, together with their
charges, their contributions to alkalinity, and the lump sum to which they
belong:

```{r}
Speciation <- data.frame(
  species = c("CO2",   "HCO3",   "CO3",
              "BOH3",  "BOH4",
              "H3PO4", "H2PO4",  "HPO4",   "PO4",
              "SiOH4", "SiOOH3", "SiO2OH2",
              "H2S",   "HS",     "S2min",
              "H2SO4", "HSO4",   "SO4",
              "HF",    "F",
              "HNO3",  "NO3",
              "HNO2",  "NO2",
              "OH",    "H2O",    "H",
              "NH4",   "NH3"),

             #SumCO2  SumBOH3 SumH3PO4 SumSiOH4 SumH2S SumH2SO4 SumHF
SumHNO3 SumHNO2 SumH2O SumNH4
  charge   =  c(0:-2, 0:-1,    0:-3,    0:-2,    0:-2,  0:-2,   0:-1,  0:-
1,    0:-1,  -1:1,  1:0),
  alkfac   =  c(0:2,  0:1,     -1:2,    0:2,     0:2,   -2:0,   -1:0, -1:0,
-1:0,   1:-1,  0:1),

  lumpsum = c(rep("SumCO2",  3), rep("SumBOH3",2), rep("SumH3PO4",4),
rep("SumSiOH4",3),
              rep("SumH2S",3), rep("SumH2SO4",3), rep("SumHF",2),
rep("SumHNO3",2),
              rep("SumHNO2",2), rep("SumH2O",3), rep("SumNH4",2))
  )

  expression = list(expression("CO"[2]),             expression("HCO"[3]^"-
"),   expression("CO"[3]^"2-"),
                    expression("BOH"[3]),            expression("BOH"[4]^"-
"),
                    expression(paste("H"[3],"PO"[4])),
expression(paste("H"[2],"PO"[4]^"-")),
                    expression("HPO"[4]^"-"),        expression("PO"[4]^"3-
"),
                    expression("SiOH"[4]),
expression("SiOOH"[3]^"-"), expression(paste("SiO"[2],"OH"[2]^"2-")),
                    expression(paste("H"[2],"S")),     expression("HS"^"-
"),       expression("S"^"2-"),
                    expression(paste("H"[2],"SO"[4])),
expression("HSO"[4]^"-"),   expression("SO"[4]^"2-"),
                    expression("HF"),                  expression("F"^"-"),
                    expression("HNO"[3]),
expression("NO"[3]^"-"),
                    expression("HNO"[2]),
expression("NO"[2]^"-"),
                    expression("OH"^"-"),
expression(paste("H"[2],"O")), expression("H"^"+"),
                    expression("NH"[4]^"+"),           expression("NH"[3]))

  Speciation$expression <- expression

  knitr::kable(Speciation[, 1:4])

```

```{r}

  get.expression <- function(species = NULL, lumpsum = NULL){
    EXP <- NULL
    if (! is.null(species))
      EXP <- Speciation$expression[Speciation$species %in% species]
```
```

```
    if (! is.null(lumpsum))
       EXP <- c(EXP, Speciation$expression[Speciation$lumpsum %in%
lumpsum])
    unlist(EXP)

  }


```

Charge and proton balance

Several functions are created.

* *get.fraction* calculates the contribution of species to a lumpsum
* *get.charge* calculates the charge of a lumpsum.
* *get.CBA* calculates the charge balance alkalinity of a lumpsum.
* *get.PAL* calculates the proton acceptor level of a lumpsum.

```{r}
get.fraction <- function(pH = 4.5,
                         t = 18.252,
                         S = 34.617,
                         p = 0,
                         species = c("CO2", "HCO3", "CO3", "BOH3", "BOH4",
                                     "H3PO4", "H2PO4", "HPO4", "PO4",
                                     "SiOH4", "SiOOH3", "SiO2OH2", "H2S",
"HS", "S2min",
                                     "H2SO4", "HSO4", "SO4", "HF", "F",
"HNO3", "NO3",
                                     "HNO2", "NO2", "OH", "H", "H2O",
"NH4", "NH3"),
                         lumpsum = NULL,
                         SumHF = SETTINGS$SumHF, SumH2SO4 =
SETTINGS$SumH2SO4,
                         ...) {


  AE <- aquaenv(S = S, t = t, pH = pH, p = p,
                SumCO2   = 1e-6, SumNH4 = 1e-6, SumH2S = 1e-6, SumH3PO4 =
1e-6,
                SumSiOH4 = 1e-6, SumHNO3 = 1e-6, SumHNO2 = 1e-6, SumBOH3 =
1e-6,
                SumHF = SumHF, SumH2SO4 = SumH2SO4,
                ...)

  if (!is.null(species))
    species <- as.character(species)

  if (! is.null(lumpsum))
    species <- c(species, as.character(Speciation[Speciation$lumpsum %in%
lumpsum, "species"]))

  species <- unique(species)

  if (is.null(species))
    species <- c("CO2", "HCO3", "CO3", "BOH3", "BOH4", "H3PO4", "H2PO4",
                 "HPO4", "PO4", "SiOH4", "SiOOH3", "SiO2OH2", "H2S", "HS",
"S2min",
                 "H2SO4", "HSO4", "SO4", "HF", "F", "HNO3", "NO3", "HNO2",
"NO2",
                 "OH", "H", "H2O", "NH4", "NH3")

  W <- which(species %in% c("OH", "H2O", "pH", "H",  "H2SO4", "HSO4",
"SO4", "HF", "F"))

  if (length(W)) {
    Species <- species[ -W]
    Water   <- species[W]
  } else {
```

```
    Species <- species
    Water   <- NULL
  }

  if (length(Species))
    fraction <- as.data.frame(AE[Species])*1e6
  else
    fraction <- NULL

  Cwat <- 55.4939   # concentration of SUM water

  if ("OH" %in% Water)
    fraction <- cbind(fraction, OH = unlist(as.data.frame(AE["OH"]))/Cwat)

  if ("H2O" %in% Water)
    fraction <- cbind(fraction, H2O = unlist(1 -
as.data.frame(AE["OH"])/Cwat))

  if ("H" %in% Water)
    fraction <- cbind(fraction, H   = unlist(10^(-
as.data.frame(AE["pH"]))))

  if ("pH" %in% Water)
    fraction <- cbind(fraction, pH   = as.data.frame(AE["pH"]))

  SumH2SO4 <- AE[["SumH2SO4"]]
  SumHF    <- AE[["SumHF"]]
  if ("H2SO4" %in% Water)
    fraction <- cbind(fraction, H2SO4 =
unlist(as.data.frame(AE["H2SO4"])/SumH2SO4))
  if ("HSO4" %in% Water)
    fraction <- cbind(fraction, HSO4  =
unlist(as.data.frame(AE["HSO4"])/SumH2SO4))
  if ("SO4" %in% Water)
    fraction <- cbind(fraction, SO4   =
unlist(as.data.frame(AE["SO4"])/SumH2SO4))
  if ("HF" %in% Water)
    fraction <- cbind(fraction, HF    =
unlist(as.data.frame(AE["HF"])/SumHF))
  if ("F" %in% Water)
    fraction <- cbind(fraction, F     =
unlist(as.data.frame(AE["F"])/SumHF))

  if (length(species) > 1)
      fraction <- fraction[ ,species]  # to have same ordering as input
  if (is.matrix(fraction) )
    fraction <- as.data.frame(fraction)

  row.names(fraction) <- NULL

  return(fraction)
}
```


```{r}
bjerrum <- function(lumpsum = c("SumCO2", "SumBOH3", "SumH3PO4",
"SumSiOH4",
                                "SumH2S", "SumH2SO4", "SumHF", "SumHNO3",
                                "SumHNO2", "SumH2O", "SumNH4"),
                    pH = seq(from = 0, to = 14, by = 0.1), plot = TRUE,
                    legend = TRUE, ...) {
    Lump <- match.arg(lumpsum, several.ok = TRUE)
    bjerrum <- get.fraction(species = NULL, lumpsum = Lump, pH = pH)

    if (plot){
        plt <- list(...)
        if (is.null(plt)) plt <- list()
        if (is.null(plt$main)) plt$main <- Lumpsum[Lump,]$expression
        if (is.null(plt$type)) plt$type <- "l"
        if (is.null(plt$lwd))  plt$lwd <- 2
```

```
        if (is.null(plt$lty))  plt$lty <- 1
        if (is.null(plt$ylab)) plt$ylab <- "-"
        if (is.null(plt$xlab)) plt$xlab <- "pH"
        if (is.null(plt$col))  plt$col <- 1:ncol(bjerrum)

        do.call("matplot", c(alist(x = pH, y = bjerrum), plt))

        if (is.logical(legend)){
          if (legend)
             legend("right", legend = as.expression(get.expression(lumpsum
= Lump)),
                 col = plt$col, lwd = plt$lwd, lty = plt$lty)
        } else if (is.list(legend)){
          if (is.null(legend$legend))
            legend$legend = as.expression(get.expression(lumpsum = Lump))
          if (is.null(legend$col))
            legend$col = plt$col
          if (is.null(legend$lwd))
            legend$lwd = plt$lwd
          if (is.null(legend$lty))
            legend$lty = plt$lty

          do.call("legend", legend)
        }
    }
    invisible(bjerrum)
}
```

```{r}
get.charge <- function(pH = 4.5, t = 18.252, S = 34.617, p = 0,
                       lumpsum = c("SumCO2", "SumBOH3", "SumH3PO4",
"SumSiOH4",
                                   "SumH2S", "SumH2SO4", "SumHF",
                                   "SumHNO3", "SumHNO2", "SumH2O",
"SumNH4")) {
 if (length(pH) == 1) {
   Spec     <- Speciation[Speciation$lumpsum %in% lumpsum, ]
   fraction <- t(get.fraction(pH = pH, t = t, S = S, p = p,
                          species = as.character(Spec$species)))
   Total.charge <- Spec$charge*fraction

   return(tapply(Total.charge, INDEX = Spec$lumpsum, FUN = sum)[lumpsum])
# summed per lumpsum

 } else {

   Charge <- NULL

   for (Lump in lumpsum){
     Spec <- Speciation[Speciation$lumpsum == Lump, ]
     fraction <- t(get.fraction(pH = pH, t = t, S = S, p = p,
                           species = as.character(Spec$species)))
     Total.charge <- Spec$charge*fraction
     CS <- colSums(Total.charge)

     Charge <- cbind(Charge, CS)
   }
  if (is.vector(Charge))
    Charge <- as.matrix(Charge)
  colnames(Charge) <- lumpsum
  return(Charge)
 }

}

get.CBA <- function(pH = 4.5, t = 18.252, S = 34.617, p = 0,
                       lumpsum = c("SumCO2", "SumBOH3", "SumH3PO4",
"SumSiOH4",
                                   "SumH2S", "SumH2SO4", "SumHF",
```

```
                                    "SumHNO3", "SumHNO2", "SumH2O",
"SumNH4")) {

  -get.charge(pH = pH, t = t, S = S, p = p, lumpsum = lumpsum)

}
```

```{r}
get.PAL <-function(pH = 4.5, t = 18.252, S = 34.617, p = 0,
                   lumpsum = c("SumCO2", "SumBOH3", "SumH3PO4",
"SumSiOH4",
                               "SumH2S", "SumH2SO4", "SumHF",
                               "SumHNO3", "SumHNO2", "SumH2O",
"SumNH4")){
 if (length(pH) == 1){
   Spec     <- Speciation[Speciation$lumpsum %in% lumpsum, ]
   fraction <- t(get.fraction(pH = pH, t = t, S = S, p = p,
                         species = as.character(Spec$species)))
   Total.alkalinity <- Spec$alkfac * fraction
   tapply(Total.alkalinity, INDEX = Spec$lumpsum, FUN = sum)[lumpsum]

 } else {

  Alk <- NULL

  for (Lump in lumpsum){
    Spec <- Speciation[Speciation$lumpsum == Lump, ]
    fraction <- t(get.fraction(pH = pH, t = t, S = S, p = p,
                         species = as.character(Spec$species)))
    Ch <- Spec$alkfac*fraction
    CS <- colSums(Ch)

    Alk <- cbind(Alk, CS)
  }
  if (is.vector(Alk))
    Alk <- as.matrix(Alk)
  colnames(Alk) <- lumpsum
  return(Alk)

 }

}
```

Charge and proton acceptor level as a function of pH

```{r, fig.width = 8, fig.height = 12}
Charge  <- get.charge(pH = pH.seq)
Charge  <- as.data.frame(Charge)
Charge$CBA <- 1
Charge$pH <- pH.seq

alkalinity  <- get.PAL(pH = pH.seq)
alkalinity  <- as.data.frame(alkalinity)
alkalinity$pH <- pH.seq
alkalinity$TA <- 1

P_C    <- bjerrum("SumCO2",   pH = pH.seq, plot = FALSE)
cnames <- c("pH", "SumCO2", "SumBOH3", "SumH3PO4","SumNH4",
            "SumH2S", "SumH2SO4", "SumHF", "SumHNO2", "SumHNO3",
"SumSiOH4")

PAL <- alkalinity[ ,cnames]
CH  <- Charge    [ ,cnames]

par (mar = c(4,4,4,7), mfrow = c(3,1),las = 1)
matplot(x = pH.seq, y = P_C, type = "l", lty = 1, lwd = 2, xlab = "pH",
ylab = "-",
```

```
        main = "DIC speciation", col = "grey")
CC <- alpha.col (col = 1, alpha = 0.5)
N <- length(pH.seq)
polygon(x = c(pH.seq[1],pH.seq,pH.seq[N],pH.seq[1]),y=c(0,P_C[,1],0,0),
col=CC,border=NA)
polygon(x = c(pH.seq[1],pH.seq,pH.seq[N],pH.seq[1]),y=c(0,P_C[,2],0,0),
col=CC,border=NA)
polygon(x = c(pH.seq[1],pH.seq,pH.seq[N],pH.seq[1]),y=c(0,P_C[,3],0,0),
col=CC,border=NA)

text(3,0.97, expression(paste("H"[2],"CO"[3])))
text(7.6,0.97, expression(paste("HCO"[3]^"-")))
text(12.5,0.97, expression("CO"[3]^"2-"))

text(1.3,0.97,"(A)", cex = 1.2)

par(new = TRUE)
plot(x = PAL$pH, y = PAL$SumCO2, axes = FALSE, xlab = "", ylab="", lty = 1,
lwd = 2, type = "l")
axis (side = 4)
mtext(outer = FALSE, "Proton acceptor level", side = 4, line =-1, las=0)
par(new = TRUE)  #charge
plot(x = CH$pH, y = CH$SumCO2, type = "l", lty = 2, lwd = 2, col =
"darkred", axes = FALSE, xlab="",ylab="")
axis(side=4, line = 4, col = "darkred", lty = 2, lwd = 2)
mtext(outer = FALSE, "Charge", side = 4, line =3, las=0, col = "darkred")
abline (lty = 2, v = 4.5)

LTY  <- c(rep(1,4),rep(2,4),rep(3,2))
COLS <- c(2,4,8,10)

matplot(PAL$pH, PAL[,-1], type = "l", lwd = 2, lty = LTY, las = 1, col =
COLS,
        xlab = "pH", main = "Proton acceptor level", ylab = "-")
text(5.2,0.5,col=2,bty="o",expression(sum("CO"[2])))
text(8.0,0.5,col=4,expression(sum("B(OH)"[3])))
text(9,2,col=8,expression(sum(paste("H"[3], "PO"[4]))))
text(14,0.9,col=10,expression(sum("NH"[4])))
text(8.8,0.9,col=2,expression(sum(paste("H"[2], "S"))))
text(1.1,-0.6,col=4,expression(sum(paste("H"[2], "SO"[4]))))
text(1.7,-0.75,col=8,expression(sum("HF")))
text(3.35,-0.5,col=10,expression(sum("HNO"[2])))
text(14,-0.1,col=2,expression(sum("HNO"[3])))
text(14,1.7,col=4,expression(sum("Si(OH)"[4])))

text(1,2,"(B)", cex = 1.2)

matplot(CH$pH, CH[,-1], type = "l", lwd = 2, lty = LTY, las = 1, col =
COLS,
        xlab = "pH", main = "Charge", ylab = "-")
text(5.4,-0.5,col=2,expression(sum("CO"[2])))
text(7.8,-0.3,col=4,expression(sum("B(OH)"[3])))
text(6.7,-1.5,col=8,expression(sum(paste("H"[3], "PO"[4]))))
text(13.5,0.2,col=10,expression(sum("NH"[4])))
text(7.8,-0.7,col=2,expression(sum(paste("H"[2], "S"))))
text(2.8,-1.8,col=4,expression(sum(paste("H"[2], "SO"[4]))))
text(2.25,-0.5,col=8,expression(sum("HF")))
text(3.35,-0.5,col=10,expression(sum("HNO"[2])))
text(2,-1.1,col=2,expression(sum("HNO"[3])))
text(13.5,-1.5,col=4,expression(sum("Si(OH)"[4])))

text(14,1,"(C)",cex = 1.2)
```

Charge change by processes.

This shows the total charge of species that are involved in a certain reaction and that need to be adjusted by proton uptake/release.

It is similar as the figures in the Soetaert et al. (2007) paper, but then without the multiplication with the "sensitivity" factor.

```{r}
ProcessCoeff <- function(SumCO2=0, SumBOH3=0, SumH3PO4=0, SumSiOH4=0,
                         SumH2S=0, SumH2SO4=0, SumHF=0,
                         SumHNO3=0, SumHNO2=0, SumH2O=0, SumNH4=0,    #
change in concentration of lumpsum species
                         TA = 0,           # change in total alkalinity
                         CBA = 0)          # change in charge balance
(excess negative charge)
{

  c(SumCO2 = SumCO2, SumBOH3=SumBOH3, SumH3PO4=SumH3PO4, SumSiOH4=SumSiOH4,
    SumH2S=SumH2S, SumH2SO4=SumH2SO4, SumHF=SumHF, SumHNO3=SumHNO3,
    SumHNO2=SumHNO2, SumH2O=SumH2O, SumNH4=SumNH4, TA=TA, CBA=CBA)
}

ProcessCoefficients <- t(data.frame(
 Oxicmineralisation = ProcessCoeff(SumCO2=1,   SumNH4=NC, SumH3PO4=PC,
TA=NC-PC,         CBA=0),
 Denitrification    = ProcessCoeff(SumCO2=1,   SumNH4=NC, SumH3PO4=PC,
SumHNO3=-0.8,              TA=0.8+NC-PC,     CBA=0),
 Denitrification2   = ProcessCoeff(SumCO2=1,              SumH3PO4=PC,
SumHNO3=-(0.8+0.6*NC),     TA=0.8+0.6*NC-PC, CBA=0),
 Feoxidation        = ProcessCoeff(SumCO2=1,   SumNH4=NC, SumH3PO4=PC,
TA=NC-PC+8,       CBA=8),
 Mnoxidation        = ProcessCoeff(SumCO2=1,   SumNH4=NC, SumH3PO4=PC,
TA=NC-PC+4,       CBA=4),
 Sulfatereduction   = ProcessCoeff(SumCO2=1,   SumNH4=NC, SumH3PO4=PC,
SumH2SO4=-0.5, SumH2S=0.5, TA=NC-PC+1,       CBA=0),
 Methanogenesis     = ProcessCoeff(SumCO2=0.5, SumNH4=NC, SumH3PO4=PC,
TA=NC-PC,         CBA=0),

 Nitrification      = ProcessCoeff(SumNH4=-1, SumHNO3=1,            TA=-2,
CBA=0),
 Anammox            = ProcessCoeff(SumNH4=-1, SumHNO2=-1,           TA=0,
CBA=0),
 MnreoxidationO2    = ProcessCoeff(                                 TA=-2,
CBA=-2),
 FereoxidationO2    = ProcessCoeff(                                 TA=-2,
CBA=-2),
 FereoxidationNO3   = ProcessCoeff(SumHNO3=-0.2,                    TA=-
1.8, CBA=-2),  # NOTE: WRONG IN TABLE SOETAERT ET AL!
 FereoxidationMn    = ProcessCoeff(                                 TA=-1,
CBA=-1),
 SreoxidationO2     = ProcessCoeff(SumH2S=-1, SumH2SO4=1,           TA=-2,
CBA=0),
 MethaneoxidationO2 = ProcessCoeff(SumCO2=1,                        TA=0,
CBA=0),
 AOM                = ProcessCoeff(SumH2SO4=-1, SumH2S=1, SumCO2=1, TA=2,
CBA=0),

 FeSoxidationO2     = ProcessCoeff(SumH2SO4=1,  TA=-2, CBA=0),
 FeSoxidationMn     = ProcessCoeff(SumH2SO4=1,  TA=8,  CBA=10),
 FeSoxidationFe     = ProcessCoeff(             TA=6,  CBA=6),
 FeSprecipitation   = ProcessCoeff(SumH2S=-1.5, TA=0,  CBA=0),
 FeSprecipitationFe = ProcessCoeff(SumH2S=-1,   TA=-2, CBA=-2),
 FeS2formation      = ProcessCoeff(SumH2S=-1,   TA=0,  CBA=0),
 MnCO3formation     = ProcessCoeff(SumCO2=-1,   TA=-2, CBA=-2),
 FeCO3formation     = ProcessCoeff(SumCO2=-1,   TA=-2, CBA=-2),
 CaSO4formation     = ProcessCoeff(SumH2SO4=-1, TA=0,  CBA=-2),
 S0formationFe      = ProcessCoeff(SumH2S=-1,   TA=4,  CBA=4),
 S0formationMn      = ProcessCoeff(SumH2S=-1,   TA=2,  CBA=2),
 Adsorption         = ProcessCoeff(             TA=1,  CBA=1),
 CO2release         = ProcessCoeff(SumCO2=-1,   TA=0,  CBA=0),
 NH3release         = ProcessCoeff(SumNH4=-1,   TA=-1, CBA=0),
 NH4release         = ProcessCoeff(SumNH4=-1,   TA=0,  CBA=1),
```

```
 Primaryproduction  = ProcessCoeff(SumCO2=-1, SumNH4=-NC,  SumH3PO4=-PC,
TA=-NC+PC, CBA=0),
 NO3assimilation    = ProcessCoeff(SumCO2=-1, SumHNO3=-NC, SumH3PO4=-PC,
TA=NC+PC,  CBA=0),
 CaCO3production    = ProcessCoeff(SumCO2=-1, TA=-2, CBA=-2),
 CaCO3dissolution   = ProcessCoeff(SumCO2=1,  TA=2,  CBA=2))
)
```

```{r}
dTA<- function(pH = 1:12, t = 18.252, S = 34.617, p = 0,
               SumCO2=0, SumBOH3=0, SumH3PO4=0, SumSiOH4=0,
               SumH2S=0, SumH2SO4=0, SumHF=0,
               SumHNO3=0, SumHNO2=0, SumH2O=0, SumNH4=0,    # change in
concentration of lumpsum species
               TA = 0,
               process = ProcessCoeff(SumCO2, SumBOH3, SumH3PO4, SumSiOH4,
                                      SumH2S, SumH2SO4, SumHF,
                                      SumHNO3, SumHNO2, SumH2O, SumNH4, TA
= TA)
                           )  {
  pCoeff <- process[process!=0]              # Coefficients that are not 0
and not CBA
  pCoeff <- pCoeff[names(pCoeff) != "CBA"]

  lumpsum <- names(pCoeff)
  lumpsum <- lumpsum[!lumpsum %in% c("TA")]

  if (length(lumpsum))
    ALK <- -get.PAL(pH = pH, t = t, S = S, p = p, lumpsum = lumpsum)   #
PAL at the pH for the relevant lump sums
  else
    ALK <- NULL

  if ("TA" %in% names(pCoeff)) ALK <- cbind(ALK, TA = 1)             # if
TA is added
  if (is.matrix(ALK)){
    dTA <- t( t(ALK) * pCoeff[colnames(ALK)])
    return(rowSums(dTA))
  } else if(length(lumpsum) > 1)
    return (sum(ALK * pCoeff[names(ALK)]))
  else
    return (ALK * pCoeff)
}
```

```{r}
dCBA <- function(pH = 1:12, t = 18.252, S = 34.617, p = 0,
                 SumCO2=0, SumBOH3=0, SumH3PO4=0, SumSiOH4=0,
                 SumH2S=0, SumH2SO4=0, SumHF=0,
                 SumHNO3=0, SumHNO2=0, SumH2O=0, SumNH4=0,    # change in
concentration of lumpsum species
                 CBA = 0,
                 process = ProcessCoeff(SumCO2, SumBOH3, SumH3PO4,
SumSiOH4,
                                        SumH2S, SumH2SO4, SumHF,
                                        SumHNO3, SumHNO2, SumH2O, SumNH4,
CBA = CBA)
                           )  {
  pCoeff <- process[process!=0]
  pCoeff <- pCoeff[names(pCoeff) != "TA"]

  lumpsum <- names(pCoeff)
  lumpsum <- lumpsum[!lumpsum %in% c("CBA")]
  if (length(lumpsum))
    CBA <- get.charge(pH = pH, t = t, S = S, p = p, lumpsum = lumpsum)
  else
    CBA <- NULL
  if ("CBA" %in% names(pCoeff)) CBA <- cbind(CBA, CBA = 1)

  if (is.matrix(CBA)){
```

```
    dCBA <- t( t(CBA) * pCoeff[colnames(CBA)])
    return(rowSums(dCBA))
  } else if(length(lumpsum) > 1)
    return (sum(CBA * pCoeff[names(CBA)]))
  else
    return (CBA * pCoeff)

}
```

Bufferfactors and ENC

```{r}
dCBAdH.function <- function(ae, lumpsum = c("SumCO2", "SumBOH3",
"SumH3PO4", "SumSiOH4",
                    "SumH2S", "SumH2SO4", "SumHF", "SumHNO3", "SumHNO2",
"SumNH4"))
{
  with(as.list(ae),
       {
         H <- 10^(-pH)

         dENCdH <- (-1/H) * (OH + H)       # Internal enhancement of
buffering

         if ("SumCO2" %in% lumpsum & SumCO2 >0)
           dENCdH <- dENCdH + (-1/H) * (HCO3 * (c1 - c3) + 2 * CO3 * (2 *
c1 + c2))

         if ("SumBOH3" %in% lumpsum & SumBOH3 >0)
           dENCdH <- dENCdH + (-1/H) * (BOH4 * b1)

         if ("SumH3PO4" %in% lumpsum & SumH3PO4 >0)
           dENCdH <- dENCdH + (-1/H) * (H2PO4 * (p1 - p3 - 2 * p4) +
                         2 * HPO4 * (2 * p1 + p2 - p4) + 3 * PO4 * (3 *
p1 + 2 * p2 + p3))

         if ("SumSiOH4" %in% lumpsum & SumSiOH4 >0)
           dENCdH <- dENCdH + (-1/H) * (SiOOH3 * (si1 -si3) + 2*SiO2OH2 *
(2*si1 + si2))

         if ("SumH2S" %in% lumpsum & SumH2S > 0)
           dENCdH <- dENCdH + (-1/H) * (HS * (s1 - s3) + 2 * S2min * (2 *
s1 + s2))

         if ("SumH2SO4" %in% lumpsum & SumH2SO4 >0)
           dENCdH <- dENCdH + (-1/H) * (HSO4 * (so1 - so3) + 2 * SO4 * (2 *
so1 + so2))

         if ("SumHF" %in% lumpsum & SumHF >0)
           dENCdH <- dENCdH + (-1/H) * (ae[["F"]] * f1)

         if ("SumHNO3" %in% lumpsum & SumHNO3 >0)
           dENCdH <- dENCdH + (-1/H) * (NO3 * na1)

         if ("SumHNO2" %in% lumpsum & SumHNO2 > 0)
           dENCdH <- dENCdH + (-1/H) * (NO2 * ni1)

         if ("SumNH4" %in% lumpsum & SumNH4 > 0)
           dENCdH <- dENCdH + (-1/H) * (-NH4 * n2)

         return(dENCdH)
         })
}

dHdCBA.function <- function(ae, lumpsum = c("SumCO2", "SumBOH3",
"SumH3PO4", "SumSiOH4",
                    "SumH2S", "SumH2SO4", "SumHF", "SumHNO3", "SumHNO2",
"SumNH4"))
  1/dCBAdH.function (ae, lumpsum = lumpsum)
```

```
dpH.dCBA <- function(ae, lumpsum = c("SumCO2", "SumBOH3", "SumH3PO4",
"SumSiOH4",
                     "SumH2S", "SumH2SO4", "SumHF", "SumHNO3", "SumHNO2",
"SumNH4"))
  (-log(10)*as.numeric(10^(-ae$pH)) * dCBAdH.function(ae = ae, lumpsum =
lumpsum))^(-1)

```

```{r}
dTAdH.function <- function(ae, lumpsum = c("SumCO2", "SumBOH3", "SumH3PO4",
"SumSiOH4",
                     "SumH2S", "SumH2SO4", "SumHF", "SumHNO3", "SumHNO2",
"SumNH4"))
{
  with(as.list(ae),
       {
         H <- 10^(-pH)
         dTAdH <- (-1/H) * (OH + H)       # Internal enhancement of
buffering

         if ("SumCO2" %in% lumpsum & SumCO2 >0)
           dTAdH <-dTAdH + (-1/H) * (HCO3 * (c1 - c3) + 2 * CO3 * (2 * c1 +
c2))

         if ("SumBOH3" %in% lumpsum & SumBOH3 > 0)
           dTAdH <-dTAdH + (-1/H) * (BOH4 * b1)

         if ("SumH2S" %in% lumpsum & SumH2S > 0)
           dTAdH <-dTAdH + (-1/H) * (HS * (s1 - s3) + 2 * S2min * (2 * s1 +
s2))

         if ("SumSiOH4" %in% lumpsum & SumSiOH4 > 0)
           dTAdH <-dTAdH + (-1/H) * (SiOOH3 * (si1 - si3) + 2 * SiO2OH2 *
(2 * si1 + si2))

         if ("SumNH4" %in% lumpsum & SumNH4 > 0)
           dTAdH <-dTAdH + (-1/H) * (NH3 * n1)

         if ("SumH3PO4" %in% lumpsum & SumH3PO4 > 0)
           dTAdH <-dTAdH + (-1/H) * (-H3PO4 * (-p2 - 2 * p3 - 3 * p4) +
HPO4 * (2 * p1 + p2 -
                                                    p4) + 2 * PO4 * (3 * p1
+ 2 * p2 + p3))

         if ("SumHNO3" %in% lumpsum & SumHNO3 > 0)
           dTAdH <-dTAdH + (-1/H) * (-HNO3 * na2)

         if ("SumHNO2" %in% lumpsum & SumHNO2 > 0)
           dTAdH <-dTAdH + (-1/H) * (-HNO2 * ni2)

         if ("SumHF" %in% lumpsum & SumHF > 0)
           dTAdH <-dTAdH + (-1/H) * (-HF * f2)

         if ("SumH2SO4" %in% lumpsum & SumH2SO4 > 0)
           dTAdH <-dTAdH + (-1/H) * (-2 * H2SO4 * (-so2 - 2 * so3) - HSO4 *
(so1 - so3))

        return(dTAdH)
         })
}

dHdTA.function <- function(ae, lumpsum = c("SumCO2", "SumBOH3", "SumH3PO4",
"SumSiOH4",
                     "SumH2S", "SumH2SO4", "SumHF", "SumHNO3", "SumHNO2",
"SumNH4"))
  1/dTAdH.function(ae, lumpsum = lumpsum)

dpH.dTA <- function(ae, lumpsum = c("SumCO2", "SumBOH3", "SumH3PO4",
"SumSiOH4",
                     "SumH2S", "SumH2SO4", "SumHF", "SumHNO3", "SumHNO2",
```

```
"SumNH4"))
  (-log(10)*as.numeric(10^(-ae$pH)) * dTAdH.function(ae = ae, lumpsum =
lumpsum))^(-1)

```

```{r}
ae = with(SETTINGS, aquaenv(pH = pH.seq, SumCO2 = SumCO2, SumBOH3 =
SumBOH3,
            SumH3PO4 = SumH3PO4, SumSiOH4 = SumSiOH4, SumH2S = SumH2S,
            SumH2SO4 = SumH2SO4, SumHF = SumHF, SumHNO3 = SumHNO3,
            SumHNO2 = SumHNO2, SumNH4 = SumNH4, S = S, t = t, p = p, dsa =
TRUE) )
dpHdCBA <- dpH.dCBA(ae)/1e6
dpHdTA <- dpH.dTA(ae)/1e6


species = c("SumCO2", "SumBOH3", "SumH2S", "SumSiOH4", "SumNH4",
"SumH3PO4",
            "SumHNO3", "SumHNO2", "SumHF", "SumH2SO4")
dpHdDIC <- NULL
RF <- NULL

for (pH in pH.seq)   {
  ae = with(SETTINGS, aquaenv(pH = pH, SumCO2 = SumCO2, SumBOH3 = SumBOH3,
              SumH3PO4 = SumH3PO4, SumSiOH4 = SumSiOH4, SumH2S = SumH2S,
              SumH2SO4 = SumH2SO4, SumHF = SumHF, SumHNO3 = SumHNO3,
              SumHNO2 = SumHNO2, SumNH4 = SumNH4, S = S, t = t, p = p, dsa
= TRUE) )
   dpHdDIC <- c(dpHdDIC,BufferFactors(ae = ae)$dpH.dtotX["SumCO2"]/1e6 )
   RF      <- c(RF,      BufferFactors(ae = ae)$RF[1])
}
```


```{r, fig.width=8, fig.height=10}
par(mfrow = c(3,2), las = 1, mar = c(5, 5.5, 4, 0.5))

pH.peaks.CBA <-
c(pH.seq[which(dpHdCBA==max(dpHdCBA[0:(length(pH.seq)*1/3)]))],

pH.seq[which(dpHdCBA==max(dpHdCBA[(length(pH.seq)*1/3+1):(length(pH.seq)*2/3)]))]
,

pH.seq[which(dpHdCBA==max(dpHdCBA[(length(pH.seq)*2/3+1):length(pH.seq)]))]
)
plot(pH.seq, dpHdCBA, type = "l", lwd = 3, col = 1, xlab = "pH", ylab =
"",main =  "dpH/dCBA")
mtext(expression(paste("/(",mu,"mol/kg)")),side=2, las = 0, padj = -3.4)
abline(v = pH.peaks.CBA, col = "grey")
text(14,0.0045,"(A)",cex=1.2)
plot(pH.seq, dpHdCBA, xlim = c(6,9), type = "l", lwd = 3, col = 2, xlab =
"pH", ylab = "",main =  "dpH/dCBA")


pH.peaks.DIC <-
c(pH.seq[which(dpHdDIC==min(dpHdDIC[0:(length(pH.seq)*2/3)]))],

pH.seq[which(dpHdDIC==min(dpHdDIC[(length(pH.seq)*2/3+1):length(pH.seq)]))]
)
plot(pH.seq, dpHdDIC, type = "l", lwd = 3, col = 1, xlab = "pH", ylab =
"",main =  "dpH/dDIC")
mtext(expression(paste("/(",mu,"mol/kg)")),side=2, las = 0, padj = -3.4)
abline(v = pH.peaks.DIC, col = "grey")
text(14,-0.0034,"(B)",cex=1.2)
plot(pH.seq, dpHdDIC, xlim = c(6,9), type = "l", lwd = 3, col = 2, xlab =
"pH", ylab = "",main =  "dpH/dDIC")


pH.peaks.RF <- c(pH.seq[which(RF==max(RF[0:(length(pH.seq)*2/3)]))],
```

```
pH.seq[which(RF==max(RF[(length(pH.seq)*2/3+1):length(pH.seq)]))])
plot(pH.seq, RF, type = "l", lwd = 3, col = 1, xlab = "pH", ylab = "",main
=  "Revelle sensitivity factor")
mtext("-",side=2, las = 0, padj = -3.4)
abline(v = pH.peaks.RF, col = "grey")
text(14,17,"(C)",cex=1.2)
plot(pH.seq, RF, xlim = c(6,9), type = "l", lwd = 3, col = 2, xlab = "pH",
ylab = "",main =  "Revelle sensitivity factor")
```

pH versus DIC/alkalinity

```{r}
DIC.seq  <- seq(from = 1950, to = 2150, length.out = 100)
TA.seq   <- seq(from = 2200, to = 2450, length.out = 100)
pH.mat   <- outer(DIC.seq, TA.seq,
                  FUN = function(DIC, TA)
                  aquaenv(SumCO2 = DIC/1e6, SumBOH3 =SETTINGS$SumBOH3,
SumH3PO4=SETTINGS$SumH3PO4,
                          SumSiOH4=SETTINGS$SumSiOH4,
SumH2S=SETTINGS$SumH2S, SumH2SO4=SETTINGS$SumH2SO4,
                          SumHF=SETTINGS$SumHF, SumHNO3=SETTINGS$SumHNO3,
SumHNO2=SETTINGS$SumHNO2,
                          SumNH4=SETTINGS$SumNH4, S=SETTINGS$S,
t=SETTINGS$t, p=SETTINGS$p,
                          TA = TA/1e6)$pH)
pCO2.mat <- outer(DIC.seq, TA.seq,
                FUN = function(DIC, TA)
                  aquaenv(SumCO2 = DIC/1e6, SumBOH3 =SETTINGS$SumBOH3,
SumH3PO4=SETTINGS$SumH3PO4,
                          SumSiOH4=SETTINGS$SumSiOH4,
SumH2S=SETTINGS$SumH2S, SumH2SO4=SETTINGS$SumH2SO4,
                          SumHF=SETTINGS$SumHF, SumHNO3=SETTINGS$SumHNO3,
SumHNO2=SETTINGS$SumHNO2,
                          SumNH4=SETTINGS$SumNH4, S=SETTINGS$S,
t=SETTINGS$t, p=SETTINGS$p,
                          TA = TA/1e6)$fCO2*1e6)
```

```{r, fig.width=6, fig.height=10}
require(shape)
par(mfrow = c(2,1), mar = c(4,4,2,3))
image2D(z = pH.mat, x = DIC.seq, y = TA.seq, las = 1, xlab =
expression(paste("DIC(",mu,"M)")),
        clab = expression("pH"), colkey = list(length = 0.5, width = 0.5,
dist = 0.02),
        ylab = expression(paste("TA(",mu,"M)")), contour = list(col =
"darkblue"),
        col = ColScheme)
text(2140,2440,"(A)")

center <- c(2100, 2250)

dConc <- 30
Hplus <- center+dConc*c(0, -1)
CO2   <- center+dConc*c(1, 0)
f1 <- uniroot(f = function(x) sqrt(2*x^2)-1, c(0,dConc))$root
HCO3  <- center+dConc*c(1, 1)*f1
f2 <- uniroot(f = function(x) sqrt(1.5*x^2)-1, c(0,dConc))$root
CO3  <- center+dConc*c(0.5, 1)*f2

Arrows(center[1], center[2], Hplus[1], Hplus[2], arr.type = "triangle")
cat("pH change by proton, from", p1 <- pH.mat[which.min(abs(DIC.seq -
center[1])), which.min(abs(TA.seq - center[2]))])  # corr  pH value
cat( " to ", p2 <- pH.mat[which.min(abs(DIC.seq - Hplus[1])),
which.min(abs(TA.seq - Hplus[2]))], " diff = ", p2-p1, "\n")
```

```
Arrows(center[1], center[2], CO3[1], CO3[2], arr.type = "triangle")
cat("pH change by CO3, from", p1 <- pH.mat[which.min(abs(DIC.seq -
center[1])), which.min(abs(TA.seq - center[2]))])  # corr  pH value
cat(" to ", p2 <- pH.mat[which.min(abs(DIC.seq - CO3[1])),
which.min(abs(TA.seq - CO3[2]))], " diff = ", p2-p1, "\n")

center <- c(1970, 2400)
dConc <- 30
Hplus <- center+dConc*c(0, -1)
CO2   <- center+dConc*c(1, 0)
HCO3  <- center+dConc*c(1, 1)*f1
CO3  <- center+dConc*c(0.5, 1)*f2

Arrows(center[1], center[2], Hplus[1], Hplus[2], arr.type = "triangle")
cat("pH change by proton, from", p1<-pH.mat[which.min(abs(DIC.seq -
center[1])), which.min(abs(TA.seq - center[2]))])  # corr  pH value
cat (" to ", p2<-pH.mat[which.min(abs(DIC.seq - Hplus[1])),
which.min(abs(TA.seq - Hplus[2]))], " diff = ", p2-p1, "\n")

Arrows(center[1], center[2], CO3[1], CO3[2], arr.type = "triangle")
cat("pH change by CO3, from", p1<-pH.mat[which.min(abs(DIC.seq -
center[1])), which.min(abs(TA.seq - center[2]))])  # corr  pH value
cat(" to", p2<-pH.mat[which.min(abs(DIC.seq - CO3[1])),
which.min(abs(TA.seq - CO3[2]))], " diff = ", p2-p1,  "\n")




image2D(z = pCO2.mat, x = DIC.seq, y = TA.seq, las = 1, xlab =
expression(paste("DIC(",mu,"M)")),
        clab = expression("pCO"[2]), colkey = list(length = 0.5, width =
0.5, dist = 0.02),
        ylab = expression(paste("TA(",mu,"M)")), resfac = 4, contour =
list(col = "darkblue"),
        col = ColScheme)
text(2140,2440,"(B)")

center <- c(2100, 2250)
dConc <- 30
Hplus <- center+dConc*c(0, -1)
CO2   <- center+dConc*c(1, 0)
HCO3  <- center+dConc*c(1, 1)*f1
CO3  <- center+dConc*c(0.5, 1)*f2

Arrows(center[1], center[2], Hplus[1], Hplus[2], arr.type = "triangle")
cat("pCO2 change by proton from", p1 <-pCO2.mat[which.min(abs(DIC.seq -
center[1])), which.min(abs(TA.seq - center[2]))])  # corr  pH value
cat(" to ", p2<-pCO2.mat[which.min(abs(DIC.seq - Hplus[1])),
which.min(abs(TA.seq - Hplus[2]))], " diff = ", p2-p1, "\n")

Arrows(center[1], center[2], CO3[1], CO3[2], arr.type = "triangle")
cat("pCO2 change in CO3, from", p1<-pCO2.mat[which.min(abs(DIC.seq -
center[1])), which.min(abs(TA.seq - center[2]))])  # corr  pH value
cat("to ",p2<-pCO2.mat[which.min(abs(DIC.seq - CO3[1])),
which.min(abs(TA.seq - CO3[2]))], " diff = ", p2-p1, "\n")

center <- c(1970, 2400)
dConc <- 30
Hplus <- center+dConc*c(0, -1)
CO2   <- center+dConc*c(1, 0)
HCO3  <- center+dConc*c(1, 1)*f1
CO3   <- center+dConc*c(0.5, 1)*f2

Arrows(center[1], center[2], Hplus[1], Hplus[2], arr.type = "triangle")
cat("pCO2 change by proton, from", p1<-pCO2.mat[which.min(abs(DIC.seq -
center[1])), which.min(abs(TA.seq - center[2]))])  # corr  pH value
cat("to", p2<-pCO2.mat[which.min(abs(DIC.seq - Hplus[1])),
which.min(abs(TA.seq - Hplus[2]))], " diff = ", p2-p1, "\n")

Arrows(center[1], center[2], CO3[1], CO3[2], arr.type = "triangle")
cat("pCO2 change in CO3, from", p1<-pCO2.mat[which.min(abs(DIC.seq -
```

```
center[1])), which.min(abs(TA.seq - center[2]))])  # corr  pH value
cat("to", p2<-pCO2.mat[which.min(abs(DIC.seq - CO3[1])),
which.min(abs(TA.seq - CO3[2]))], " diff = ", p2-p1, "\n")

```


Effect of biogeochemical processes on pH

```{r}

# function to estimate the pH change
# SETTINGS = default speciation as in Hagens

dPH.numeric <- function(Species = SETTINGS,
                        dSumCO2   = 0, dSumBOH3 = 0, dSumH2S = 0,   #
change in summed conc due to process
                        dSumSiOH4 = 0, dSumNH4  = 0, dSumH3PO4 = 0,
# default = 0
                        dSumHNO3  = 0, dSumHNO2 = 0, dSumHF = 0, dSumH2SO4
= 0,
                        dTA = 0,  dC = 1e-8 ,
# change in total alkalinity due to process
                        pHSEQ = pH.seq) {                      # pH range
for which to estimate dpHdProcess


  # unless otherwise mentioned the change in species composition = 0
  dS <- list(SumCO2   = dSumCO2,   SumBOH3 = dSumBOH3,  SumH3PO4 =
dSumH3PO4, # change in summed conc due to process
             SumSiOH4 = dSumSiOH4, SumH2S = dSumH2S, SumH2SO4 = dSumH2SO4,
             SumHF = dSumHF,
             SumHNO3  = dSumHNO3,  SumHNO2 = dSumHNO2, SumNH4  = dSumNH4,
dS = 0, dt = 0, dp = 0)

  SPEC <- Species[-which(names(Species) == "SumH2O")]
  SP2  <- unlist(SPEC) + dC*unlist(dS)  # new concentrations, including the
small perturbation due to the process

  dpH_dSpec <- NULL      # will contain the ultimate results

  for (pH in pHSEQ){

    # given the pH and the default species composition (SPEC), calculate TA
    TA  <- do.call("aquaenv" , c(as.list(SPEC), pH = pH))$TA

    # given new TA (+ the production by process) and the new species
composition (SP2), calculate pH
    pH2 <- do.call("aquaenv" , c(as.list(SP2), TA = TA + dTA*dC))$pH

    # estimate dpHdspec by numerical differencing. Divide by 1e6 so that pH
change is per micromol rather than per mol.
    dpH_dSpec <- c(dpH_dSpec, (pH2-pH)/(dC*1e6))  # units in per umolC
  }
  return(dpH_dSpec)
}
DENIT <- dPH.numeric(dSumCO2 =  1, dSumNH4 = NC, dSumH3PO4 = PC, dSumHNO3 =
-0.8, dTA = 0.8 + NC-PC)

```

```{r}

Numerical <- cbind(
 oxicmineralisation = dPH.numeric(dSumCO2=1,   dSumNH4=NC, dSumH3PO4=PC,
dTA=NC-PC),
 denitrification    = dPH.numeric(dSumCO2=1,   dSumNH4=NC, dSumH3PO4=PC,
dSumHNO3=-0.8,              dTA=0.8+NC-PC),
 denitrification2   = dPH.numeric(dSumCO2=1,               dSumH3PO4=PC,
dSumHNO3=-(0.8+0.6*NC),     dTA=0.8+0.6*NC-PC),
```

```
 Feoxidation        = dPH.numeric(dSumCO2=1,   dSumNH4=NC, dSumH3PO4=PC,
dTA=NC-PC+8),
 Mnoxidation        = dPH.numeric(dSumCO2=1,   dSumNH4=NC, dSumH3PO4=PC,
dTA=NC-PC+4),
 Sulfatereduction   = dPH.numeric(dSumCO2=1,   dSumNH4=NC, dSumH3PO4=PC,
dSumH2SO4=-0.5, dSumH2S=0.5,dTA=NC-PC+1),
 Methanogenesis     = dPH.numeric(dSumCO2=0.5, dSumNH4=NC, dSumH3PO4=PC,
dTA=NC-PC),

 Nitrification      = dPH.numeric(dSumNH4=-1, dSumHNO3=1,             dTA=-
2),
 Anammox            = dPH.numeric(dSumNH4=-1, dSumHNO2=-1,
dTA=0),
 Mnreoxidation      = dPH.numeric(                                    dTA=-
2),
 Fereoxidation      = dPH.numeric(                                    dTA=-
2),
 FereoxidationNO3   = dPH.numeric(dSumHNO3=-0.2,                      dTA=-
1.8),  # NOTE: WRONG IN TABLE SOETAERT ET AL!
 FereoxidationMn    = dPH.numeric(                                    dTA=-
1),
 Sreoxidation       = dPH.numeric(dSumH2S=-1, dSumH2SO4=1,            dTA=-
2),
 Methaneoxidation   = dPH.numeric(dSumCO2=1,
dTA=0),
 AOM                = dPH.numeric(dSumH2SO4=-1, dSumH2S=1, dSumCO2=1,
dTA=2),

 FeSoxidation       = dPH.numeric(dSumH2SO4=1,  dTA=-2),
 FeSoxidationMn     = dPH.numeric(dSumH2SO4=1,  dTA=8),
 FeSoxidationFe     = dPH.numeric(              dTA=6),
 FeSprecipitation   = dPH.numeric(dSumH2S=-1.5, dTA=0),
 FeSprecipitationFe = dPH.numeric(dSumH2S=-1,   dTA=-2),
 FeS2formation      = dPH.numeric(dSumH2S=-1,   dTA=0),
 MnCO3formation     = dPH.numeric(dSumCO2=-1,   dTA=-2),
 FeCO3formation     = dPH.numeric(dSumCO2=-1,   dTA=-2),
 CaSO4formation     = dPH.numeric(dSumH2SO4=-1, dTA=0),
 S0formationFe      = dPH.numeric(dSumH2S=-1,   dTA=4),
 S0formationMn      = dPH.numeric(dSumH2S=-1,   dTA=2),
 adsorption         = dPH.numeric(              dTA=1),
 CO2release         = dPH.numeric(dSumCO2=-1,   dTA=0),
 NH3release         = dPH.numeric(dSumNH4=-1,   dTA=-1),
 NH4release         = dPH.numeric(dSumNH4=-1,   dTA=0),

 primaryproduction  = dPH.numeric(dSumCO2=-1, dSumNH4=-NC,  dSumH3PO4=-PC,
dTA=-NC+PC),
 NO3assimilation    = dPH.numeric(dSumCO2=-1, dSumHNO3=-NC, dSumH3PO4=-PC,
dTA=NC+PC),
 CaCO3production    = dPH.numeric(dSumCO2=-1, dTA=-2),
 CaCO3dissolution   = dPH.numeric(dSumCO2=1,  dTA=2)
)

OXIC  <- dPH.numeric(dSumCO2 =  1, dSumNH4 = NC, dSumH3PO4 = PC, dTA = NC-
PC)
OXIC2 <- dPH.numeric(dSumCO2 =  1, dSumHNO3= NC, dSumH3PO4 = PC, dTA = -NC-
PC)
SULF  <- dPH.numeric(dSumCO2 =  1, dSumNH4 = NC, dSumH3PO4 = PC, dTA = 1  +
NC-PC)
METH  <- dPH.numeric(dSumCO2 =0.5, dSumNH4 = NC, dSumH3PO4 = PC, dTA =
NC-PC)

PPROD <- dPH.numeric(dSumCO2 = -1, dSumNH4 =-NC, dSumH3PO4 =-PC, dTA = -
NC+PC)
PPROD2<- dPH.numeric(dSumCO2 = -1, dSumHNO3=-NC, dSumH3PO4 =-PC, dTA =
NC+PC)

CALC  <- dPH.numeric(dSumCO2 = -1                          , dTA =  -2)
DISS  <- dPH.numeric(dSumCO2 =  1                          , dTA =   2)
ANNAMOX   <- dPH.numeric(dSumNH4 = -1, dSumHNO2 = -1, dTA = 0)

# Point of calcification/dissolution switch
```

```
OMEGA <- NULL
SPEC <- SETTINGS[-which(names(SETTINGS) == "SumH2O")]
for (pH in pH.seq){
  OMEGA  <- c(OMEGA, do.call("aquaenv" , c(as.list(SPEC), pH =
pH))$omega_calcite)
}

crit <- which.min(abs(OMEGA-1))
DISS[crit:length(pH.seq)] <- NA
CALC[1:crit] <- NA
```

```{r, fig.width=8, fig.height=3}
par(mfrow = c(1,3), las = 1, mar = c(5,6,4,0))
DENIT <- dPH.numeric(dSumCO2 =  1, dSumNH4 = NC, dSumH3PO4 = PC, dSumHNO3 =
-0.8, dTA = 0.8+ NC-PC)

plot(pH.seq, dpHdCBA, type = "l", ylab = "", main = "sensitivity", xlab =
"pH")
mtext(expression(paste("/(",mu,"mol/kg)")),side=2, las = 0, padj = -
3.8,cex=0.9)
text(13.7,0.00435,"(A)")

plot(pH.seq, dCBA(pH.seq, process = ProcessCoefficients["Denitrification",
]), type = "l", ylab = "", main = expression(paste(Delta,"charge",sep="")),
xlab = "pH")
mtext("-",side=2, las = 0, padj = -3,cex=0.9)
text(1.3,-1.2,"(B)")
abline(h = 0, lty = 2)

plot(pH.seq, DENIT, type = "l", ylab = "", main = "response" , xlab = "pH")
mtext("dpH",side=2, las = 0, padj = -3.9,cex=0.9)
text(13.7,0.004,"(C)")
abline(h = 0, lty = 2)
```

```{r, fig.width=10, fig.height=8}
par(mfrow = c(2,2), las = 1)
ylab <- ""
ylab = "dpH"
xlab <- "pH"
col = c("black","darkgrey")
matplot(x = pH.seq, y = cbind(OXIC, OXIC2), type = "l", ylab = ylab,
        xlab = xlab, lty = 1:2, lwd = 3, col = col)
abline(h=0, lty = 3)
legend("bottomleft", col = col, lty = 1:2, lwd = 2,
       legend = c("Aerobic", "Aerobic+Nitrif"), cex = 0.8)
text(14, (min(cbind(OXIC, OXIC2))),"(A)")
matplot(x = pH.seq, y = cbind(SULF, METH), type = "l", ylab = ylab,
        xlab = xlab, lty = 1:2, lwd = 3, col = col)
abline(h=0, lty = 3)
legend("topright", col = col, lty = 1:2, lwd = 2,
       legend = c("Sulphate reduction", "Methanogenesis"), cex = 0.8)
text(14, (min(cbind(SULF, METH))),"(B)")
matplot(x = pH.seq, y = cbind(PPROD, PPROD2), type = "l", ylab = ylab,
        xlab = xlab, lty = 1:2, lwd = 3, col = col)
abline(h=0, lty = 3)
legend("topleft", col = col, lty = 1:2, lwd = 2,
       legend = c("Prim prod (ammonium)", "Prim prod (nitrate)"), cex =
0.8)
text(14, (min(cbind(PPROD, PPROD2))),"(C)")
matplot(x = pH.seq, y = cbind(CALC, DISS), type = "l", ylab = ylab,
        xlab = xlab, lty = 1:2, lwd = 3, col = col)
abline(h=0, lty = 3)
legend("topright", col = col, lty = 1:2, lwd = 2,
       legend = c("Calcification", "Dissolution"), cex = 0.8)
text(14, (min(cbind(CALC, DISS),na.rm=TRUE)),"(D)")
```

References

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.

Middelburg, J.J., K. Soetaert and M. Hagens, 2020. Ocean alkalinity, buffering and biogeochemical processes. Reviews of Geophysics.

Hagens M. and J.J. Middelburg, 2016 Generalised expressions for the response of pH to changes in ocean chemistry. Geochimica et Cosmochimica Acta 187: 334-349. DOI: 10.1016/j.gca.2016.04.012.

Soetaert, K., AF.Hofmann, J.J.Middelburg, F.J.R.Meysman, J. Greenwood, 2007. The effect of biogeochemical processes on pH. Marine Chemistry 105, 30-51