

Reaction-transport in porous media: a simple model of phosphorus cycling in marine sediments

Exercises Accompanying the Course Reaction Transport Modelling in the Hydrosphere

Karline Soetaert and Lubos Polerecky, Utrecht University

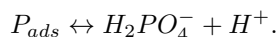
March 2021

Problem formulation

The dynamics of phosphorus (P) in sediments is complex because it is impacted by biogeochemical and physical processes. Among other processes, phosphate that is released by mineralization of organic matter (biogeochemistry) can adsorb to iron oxides, precipitate as apatite (calcium phosphate mineral), or adsorb to calcium carbonate ($CaCO_3$). Adsorption to iron oxides occurs in the oxic zone, while phosphate is released again when iron oxides are reduced in the anoxic zone of the sediment. The last two processes (authigenic apatite formation and P adsorption to $CaCO_3$) can play an important role in the long-term removal of P.

In this exercise, you will make a very simple model of P dynamics in sediments. Assume that P is only exchanging between two pools: dissolved phosphate¹ ($H_2PO_4^-$), and the phosphate adsorbed to $CaCO_3$ (P_{ads} , solid phase). The other processes involved in P dynamics will be ignored.

Assuming an *unlimited* availability of $CaCO_3$, consider that adsorption and desorption of phosphate can be represented by the following reversible reaction:



Assumptions

- The depth of the modelled sediment column is 60 cm.
- Bottom water concentration of dissolved phosphate is $0.0005 \text{ mol P m}^{-3}$.
- The sediment accretion velocity equals to $v = 0.1 \text{ cm yr}^{-1}$.
- The bioturbation mixing coefficient is $5 \text{ cm}^2 \text{ yr}^{-1}$.
- Due to sediment compaction, the sediment porosity [$m^3 \text{ liquid m}^{-3} \text{ solid}$] declines with depth from 0.9 at the surface to 0.6 at depth, with a depth attenuation factor of 0.5 cm^{-1} . Although this variation in porosity influences the advective velocity, we will ignore this in our model.
- Use the R-package *marelac* to estimate the molecular diffusion coefficient of phosphate assuming a temperature of 10 °C and a salinity of 35. Assume *constant pH* of 7. At this *pH*, temperature and salinity, the most abundant phosphate species is $H_2PO_4^{2-}$. The diffusion coefficient needs to be corrected for tortuosity, as explained in the lectures.

For the P dynamics, assume:

¹Dissolved phosphate has four species: H_3PO_4 , $H_2PO_4^-$, HPO_4^{2-} and PO_4^{3-} . The speciation depends on pH. At ambient pH, temperature and salinity assumed in this exercise, $H_2PO_4^-$ is the dominant species.

- Phosphate is released from the mineralization of organic matter.
 - Assume that the rate constant describing the organic matter mineralization is $0.01\ d^{-1}$.
 - The deposition rate of organic carbon is $0.001\ mol\ C\ m^{-2}\ d^{-1}$.
 - The $P:C$ ratio in the organic matter is according to Redfield: 1 mole of P per 106 moles of C.
- Phosphate adsorption to $CaCO_3$ is described by first-order kinetics with respect to $H_2PO_4^-$ (rate constant r_{ads}). This is a reasonable assumption if the concentration of $CaCO_3$ in the sediment is so high that it never becomes limiting.
 - In the initial model run, set $r_{ads} = 5 \times 10^{-4}\ d^{-1}$.
 - Later, perform a sensitivity analysis assuming that the value of r_{ads} varies between 0 and $5 \times 10^{-3}\ d^{-1}$.
- Phosphate desorption from $CaCO_3$ is described by first-order kinetics with respect to P_{ads} . The corresponding rate constant is $r_{des} = 1 \times 10^{-5}\ d^{-1}$.

Tasks

Task 1: Implementation in R

Add P dynamics to the template markdown file for early diagenesis *RTM_porous1D.Rmd*.² You can use the description for the carbon diagenesis in the template as a basis for the present exercise. There is no need to remove the DIC state variable from the model. However, you can ignore the possibility that DIC may be removed or added to the system due to precipitation or dissolution of $CaCO_3$.

- Use m , mol , and d to create the units used in the model. Convert all input parameters into these units (e.g., POC deposition flux, bioturbation mixing coefficient, diffusion coefficient).
- First add dissolved phosphate as a state variable, and make sure that the model can be solved with only this species added. Then add the adsorbed phosphate and implement the dynamics.
- Make sure to return from the model function ordinary output variables that will be needed when creating the P budget and interpreting the results:
 - All depth-integrated rates.
 - The fluxes in and out of the model domain.
 - Depth profiles of the process rates (think about the units in which the results will be best displayed).
 - Quotient ($Q = [PO_4^{3-}]/[P_{ads}]$), which is a measure of how far from an equilibrium the dissolved and adsorbed phosphate pools are. (Tip: when calculating Q , consider the units of the different P pools and the value of Q that you expect in an equilibrium.)

Task 2: Model applications

- Run three simulations, with the adsorption rate constant set to 0, 1×10^{-3} , and $5 \times 10^{-3}\ d^{-1}$.
- Plot the results and try to understand what happens. Plot in separate R-chunks the depth profiles of concentrations, process rates and the quotient. This will help you better understand and interpret the results. Discuss with the teachers to check your understanding.
- Check the sensitivity of your model to the depth of the sediment column in which you model the P dynamics. That is, run your model with the depth of the sediment column set to 60 *cm* (initial), 100 *cm* and 200 *cm*. Try to understand what happens and why. Discuss your ideas with the teachers.

²You can obtain this file from Rstudio: File → new File → Rmarkdown → from template → RTM_porous1D. Save this file under a different name. Do not forget to change the heading of this file.

Task 3: Phosphorus budget

- For scenario 1 (no adsorption) and scenario 3 (maximal adsorption), draw the vertically integrated mass balance budget for P, taking into account:
 - the input from the overlying water,
 - the burial rate, and
 - the (vertically integrated) conversions between the different P pools.
- Convert the units to $\mu\text{mol P m}^{-2} \text{ yr}^{-1}$.
- Discuss your diagram and P budget with the teachers.

Answers

Four species are modeled with the following concentrations and boundary conditions:

- POC: Particulate Organic Carbon, [$\text{mol C m}^{-3} \text{ solid}$]
 - imposed flux at the upstream boundary (flux.up),
 - zero-gradient boundary downstream,
- DIC: Dissolved Inorganic Carbon, [$\text{mol C m}^{-3} \text{ liquid}$]
 - imposed concentration upstream (C.up),
 - zero-gradient boundary downstream,
- P_{ads} : Adsorbed phosphorus, [$\text{mol P m}^{-3} \text{ solid}$]
 - zero flux at the upstream boundary (flux.up = 0),
 - zero-gradient boundary downstream,
- $H_2PO_4^-$: Dissolved phosphate, [$\text{mol C m}^{-3} \text{ liquid}$]
 - imposed concentration upstream (C.up),
 - zero-gradient boundary downstream.

Biogeochemical/physical processes:

- First-order mineralization of organic matter produces DIC and $H_2PO_4^-$.
- Phosphate adsorption and desorption to $CaCO_3$.

What can be ignored:

- $CaCO_3$ is assumed to be present in excess, and therefore does not need to be modeled.
- pH is assumed to be constant, and therefore does not need to be modelled.

Task 1. Model implementation in R

The partial derivatives related to transport are approximated by the function tran.1D from the ReacTran package. The steady-state and dynamic solutions are obtained using functions from the rootSolve and deSolve package. The latter two packages are loaded together with ReacTran. The diffusion coefficients are obtained using the marelac package.

```
require(marelac)
require(ReacTran)
```

The model grid and associated properties

The model domain is divided into a grid of N boxes. The depth of the modeled sediment column is set to 0.6 m. The grid size is exponentially increasing with depth, to achieve a better accuracy of the results in the sediment region where steep spatial gradients are likely present (here: close to the sediment-water interface).

```
# grid: 200 boxes, total length = 0.6 m, later increased to 1 or 2 m.
Length <- 0.6 # [m]
N <- 200

# grid with an exponentially increasing grid size, starting from 0.1 cm
Grid <- setup.grid.1D(L = Length, N = N, dx.1 = 0.1e-2)

# function describing the variation of porosity (volume fraction of liquid) with depth
porFun.L <- function(x, por.SWI, por.deep, porcoef)
  return(por.deep + (por.SWI-por.deep)*exp(-x*porcoef))

# solid volume fraction (svf = 1-porosity)
```

```

porFun.S <- function(x, por.SWI, por.deep, porcoef)
  return(1-porFun.L(x, por.SWI, por.deep, porcoef))

# calculate porosity and svf on the grid (mid-points and box interfaces, etc.)
porLiquid <- setup.prop.1D(func = porFun.L, grid = Grid,
  por.SWI = 0.9, por.deep = 0.6, porcoef = 50)

porSolid <- setup.prop.1D(func = porFun.S, grid = Grid,
  por.SWI = 0.9, por.deep = 0.6, porcoef = 50)

# Sediment diffusion coefficient for HCO3~- and HPO4~2- (m2/d),
# defined at box interfaces!! (N+1 values)
diffHCO3 <- diffcoeff(S=35, t=20)$HCO3 * 3600*24 # m2/s to m2/d
diffH2PO4 <- diffcoeff(S=35, t=20)$H2PO4 * 3600*24 # m2/s to m2/d
porInt <- porLiquid$int # porosity at box interfaces
diffDIC <- diffHCO3 / (1-log(porInt^2)) # corrected for tortuosity
diffP04 <- diffH2PO4 / (1-log(porInt^2)) # corrected for tortuosity

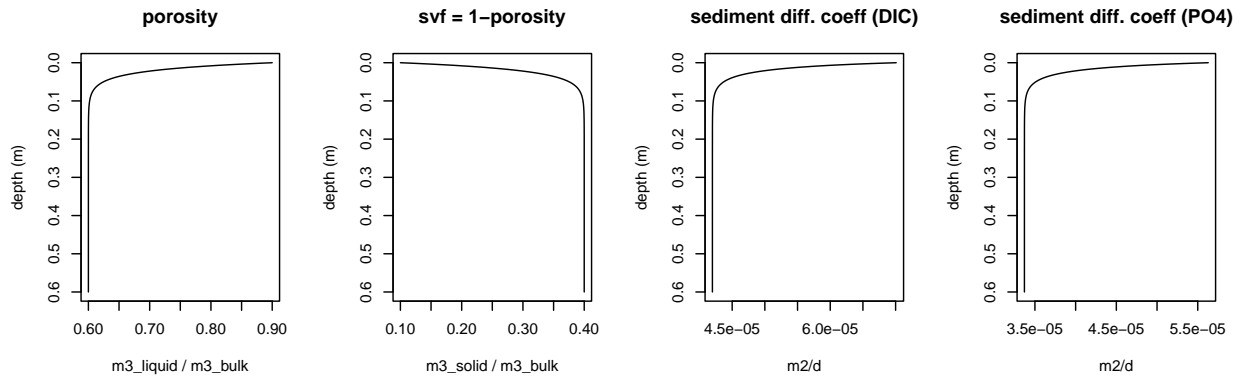
```

Visualise these settings:

```

par(mfrow = c(1,4))
plot(porLiquid, type="l", grid = Grid, xyswap = TRUE, ylab = "depth (m)",
  main = "porosity", xlab = "m3_liquid / m3_bulk")
plot(porSolid, type="l", grid = Grid, xyswap = TRUE, ylab = "depth (m)",
  main = "svf = 1-porosity", xlab = "m3_solid / m3_bulk")
plot(diffDIC, y = Grid$x.int, type="l", ylim = c(Length,0), ylab = "depth (m)",
  main = "sediment diff. coeff (DIC)", xlab = "m2/d")
plot(diffP04, y = Grid$x.int, type="l", ylim = c(Length,0), ylab = "depth (m)",
  main = "sediment diff. coeff (P04)", xlab = "m2/d")

```



The parameters

Parameters are expressed in mol, m and day; the input is converted if necessary. We ignore the influence of sediment compaction on the advection velocity.

```

parms <- c(
  biot      = 5e-4/365,      # [m2/d]      bioturbation mixing coefficient
  v         = 0.1e-2/365,    # [m/d]      sediment advection velocity
  rMin      = 0.01,          # [/d]       POC mineralisation rate constant
  depoPOC   = 1e-3,          # [mol/m2/d] POC deposition rate (flux at SWI)

```

```

bwDIC    = 2,           # [mol/m3]    DIC bottom water concentration
bwPO4    = 0.5e-3,      # [mol/m3]    PO4 bottom water concentration
PCratio  = 1/106,       # [molP/molC] P:C ratio in Redfield organic matter
rPads    = 5e-4,        # [/d]        rate constant for P adsorption to CaCO3
rPdes    = 1e-5         # [/d]        P desorption rate constant
)

```

The model function

```

PDiamodel <- function(t, Conc, pars)  # Conc is a long vector
{
  with(as.list(pars),{
    # unpack state variables
    POC <- Conc[ 1 : N ]      # first N elements: POC
    DIC <- Conc[( N+1):(2*N)] # next N elements: DIC
    PO4 <- Conc[(2*N+1):(3*N)] # next N elements: PO4
    Pads <- Conc[(3*N+1):(4*N)] # next N elements: adsorbed P

    # transport - note: zero gradient by default at lower boundaries

    # particulate substances, VF = solid volume fraction = 1-porosity!
    tran.POC <- tran.1D(C = POC, flux.up = depoPOC, # upper boundary: flux
                        dx = Grid, VF = porSolid,   # grid and volume fraction (1-por)
                        D = biot, v = v)            # mixing (bioturbation) and advection

    tran.Pads <- tran.1D(C = Pads, flux.up = 0,      # upper boundary: zero flux
                        dx = Grid, VF = porSolid,   # grid and volume fraction (1-por)
                        D = biot, v = v)            # mixing (bioturbation) and advection

    # dissolved substances, VF = liquid volume fraction = porosity!
    tran.DIC <- tran.1D(C = DIC, C.up = bwDIC,      # upper boundary: concentration
                        dx = Grid, VF = porLiquid,  # grid and volume fraction (por)
                        D = diffDIC, v = v)         # diffusive mixing and advection
                                                    # (bioturbation mixing negligible)

    tran.PO4 <- tran.1D(C = PO4, C.up = bwPO4,      # upper boundary: concentration
                        dx = Grid, VF = porLiquid,  # grid and volume fraction (por)
                        D = diffPO4, v = v)         # diffusive mixing and advection

    # === reaction rates ===
    # POC mineralisation
    Mineralisation <- rMin * POC                  # [mol/m3 SOLID/d] (per volume of solid!)

    # P dynamics
    Adsorption <- rPads * PO4                     # [mol/m3 LIQUID/d]
    Desorption <- rPdes * Pads                    # [mol/m3 SOLID /d]

    # mass balances : dC/dt = transport + reactions
    poro <- porLiquid$mid

    # solid components

```

```

dPOC.dt <- tran.POC$dC - Mineralisation
dPads.dt <- ( tran.Pads$dC
              + Adsorption*poro/(1-poro)
              - Desorption )

# solute components
dDIC.dt <- ( tran.DIC$dC +
             Mineralisation * (1-poro)/poro )
dP04.dt <- ( tran.P04$dC
             + PCratio*Mineralisation*(1-poro)/poro
             + Desorption*(1-poro)/poro
             - Adsorption)

# depth-integrated rates: [mol/m2 BULK/d] !!
TotalMin <- sum(Mineralisation*Grid$dx * (1-poro))
TotalPmin <- PCratio*TotalMin
TotalPdesorp <- sum(Desorption *Grid$dx * (1-poro))
TotalPadsorp <- sum(Adsorption *Grid$dx * poro)

return(list(c(dPOC.dt, dDIC.dt, dP04.dt, dPads.dt), # t-derivatives as a LONG vector

# Depth profiles of process rates, all in molP/m3 BULK/d!
Mineralisation_B = PCratio*Mineralisation*(1-poro),
Adsorption_B      = Adsorption*poro,
Desorption_B      = Desorption*(1-poro),

# quotient is a measure of the distance from an equilibrium
Quotient          = P04*poro / (Pads*(1-poro)), # corrected for porosity!

# for budgetting
TotalMin          = TotalMin,
TotalPmin         = TotalPmin,
TotalPdesorp      = TotalPdesorp,
TotalPadsorp      = TotalPadsorp,
DIC.SWI.Flux      = tran.DIC$flux.up,
DIC.Deep.Flux     = tran.DIC$flux.down,
POC.SWI.Flux      = tran.POC$flux.up,
POC.Deep.Flux     = tran.POC$flux.down,
P04.SWI.Flux      = tran.P04$flux.up,
P04.Deep.Flux     = tran.P04$flux.down,
Pads.SWI.Flux     = tran.Pads$flux.up,
Pads.Deep.Flux    = tran.Pads$flux.down,
POC.SWI.Flux      = tran.POC$flux.up,
POC.Deep.Flux     = tran.POC$flux.down,
POP.SWI.Flux      = PCratio*tran.POC$flux.up,
POP.Deep.Flux     = PCratio*tran.POC$flux.down))
})
}

```

Task 2. Model applications

Three steady-state solutions are calculated, with different adsorption rate constants.

```

# Number of species in the model, and their names
names <- c("POC", "DIC", "PO4", "Pads")
nspec <- length(names)

# Initial conditions
Conc <- runif(nspec*N)

# three runs with different adsorption rates
p1 <- p2 <- p3 <- parms

p1["rPads"] <- 0      # /day
sol1 <- steady.1D (y=Conc, func=PDiamodel, parms=p1,
                  nspec=nspec, dims=N, names=names,
                  pos=TRUE)

p2["rPads"] <- 1e-3   # /day
sol2 <- steady.1D (y=Conc, func=PDiamodel, parms=p2,
                  nspec=nspec, dims=N, names=names,
                  pos=TRUE)

p3["rPads"] <- 5e-3   # /day
sol3 <- steady.1D (y=Conc, func=PDiamodel, parms=p3,
                  nspec=nspec, dims=N, names=names,
                  pos=TRUE)

```

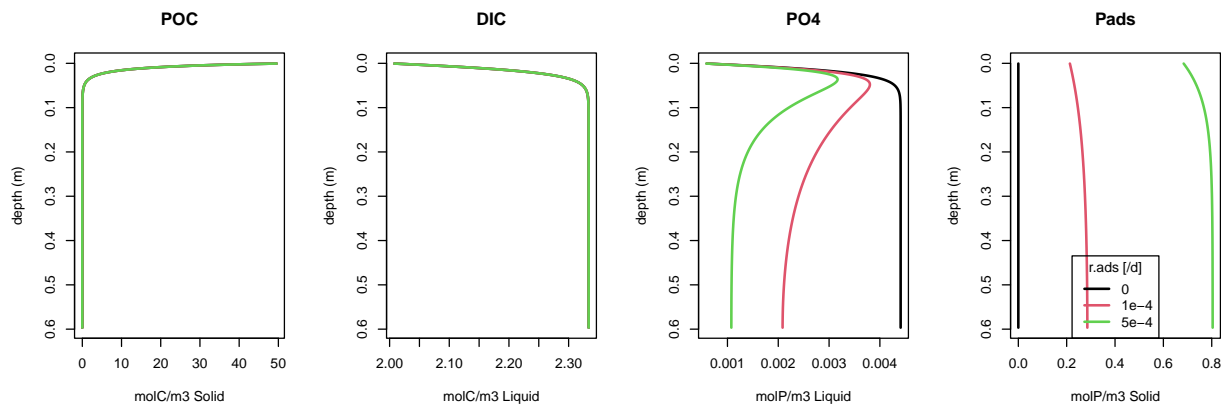
Here, we plot the concentration gradients. They show that the higher the adsorption rate, the more the dissolved phosphate is scavenged into the adsorbed phase, and the lower its concentration at depth.

```

plot(sol1, sol2, sol3, xyswap=TRUE, grid = Grid$x.mid, lty=1, lwd=2,
     xlab=c("molC/m3 Solid", "molC/m3 Liquid",
            "molP/m3 Liquid", "molP/m3 Solid"),
     ylab="depth (m)", mfrow=c(1,4))

legend("bottom", legend=c("0", "1e-4", "5e-4"), title="r.ads [/d]",
      lwd=2, lty=1, col=1:3)

```



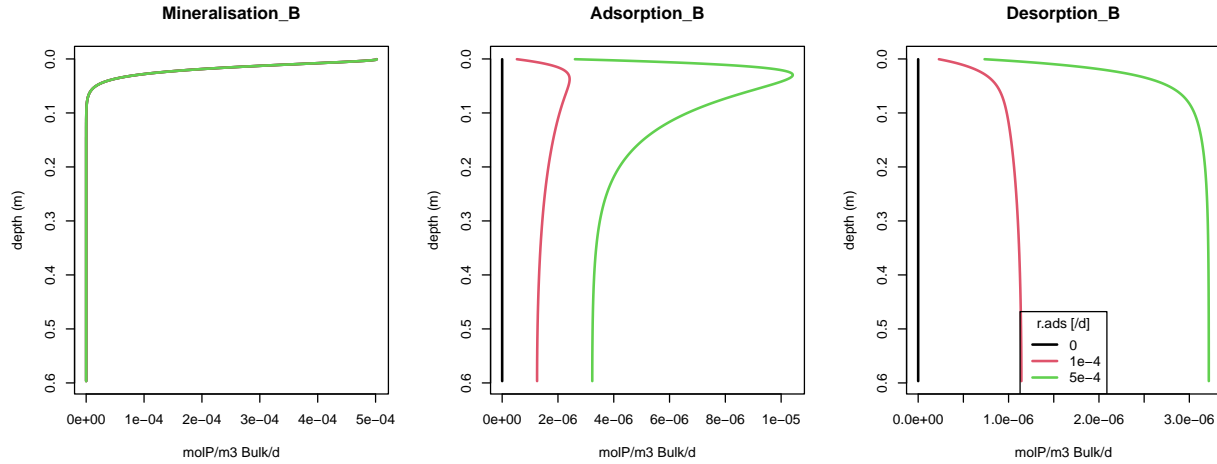
To aid interpretation of the depth profiles of concentrations, we also plot the depth profiles of the process rates. The process rates are best expressed per volume of *bulk* (indicated by the subscript 'B'), as they can be directly compared to each other.


```

plot(sol1, sol2, sol3, xyswap=TRUE, grid = Grid$x.mid, lty=1, lwd=2,
     which=c("Mineralisation_B", "Adsorption_B", "Desorption_B"),
     xlab=c("molP/m3 Bulk/d", "molP/m3 Bulk/d", "molP/m3 Bulk/d"),
     ylab="depth (m)", mfrow=c(1,3))

legend("bottom", legend=c("0", "1e-4", "5e-4"), title="r.ads [/d]",
      lwd=2, lty=1, col=1:3)

```

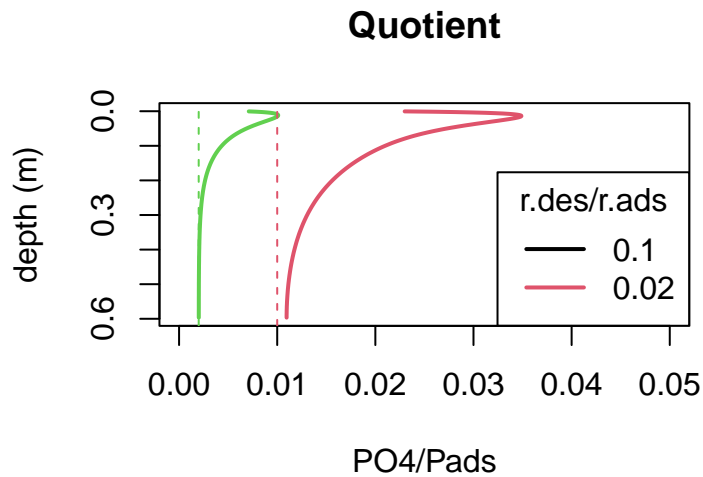


Finally, we also plot the quotient as a function of depth. We expect the value of Q in equilibrium to be equal to the equilibrium constant, which is equal to the ratio between the forward and backward reaction rate constants, $K_{eq} = r_{des}/r_{ads}$.

```

plot(sol2, sol3, xyswap=TRUE, grid = Grid$x.mid, lty=1, lwd=2,
     which = c("Quotient"), xlim = c(0,0.05), col=c(2,3),
     xlab=c("PO4/Pads"), ylab="depth (m)", mfrow=c(1,1))
abline(v=p2["rPdes"]/p2["rPads"], col=2, lty=2)
abline(v=p3["rPdes"]/p3["rPads"], col=3, lty=2)
legend("bottomright", legend=c("0.1", "0.02"), title="r.des/r.ads",
      lwd=2, lty=1, col=1:2)

```



We see that the dissolved and adsorbed phases of P are in *disequilibrium* close to the SWI, and *approaching the equilibrium* with depth. (This explains the increasing concentrations of P_{ads} with depth, as a result of the *net* rate of P adsorption being positive.) However, the equilibrium between the dissolved and adsorbed phases is *not* reached within the sediment column of 0.6 m. This changes if the model is run for the sediment column depth set to 2 m. (Not shown in this document, but try it to see it for yourself.)

Note, that if the length of the sediment column is changed, the solution will *change*! This is understandable, because by changing the length of the spatial domain, we *also* we change the *boundary conditions*! This highlights the importance of setting the boundary conditions so that they *realistically* represent the *in-situ* conditions of the studied system.

Task 3. Phosphorus Budget

```
convert <- 365000 # conversion from mol/m2/d to mmol/m2/year

toselect <- c("TotalPmin", "TotalPdesorp", "TotalPadsorp",
             "PO4.SWI.Flux", "PO4.Deep.Flux", "Pads.SWI.Flux", "Pads.Deep.Flux",
             "POP.SWI.Flux", "POP.Deep.Flux")
BUDGET <- data.frame(NoAdsorption = unlist(sol1[toselect]),
                    MediumAdsorption = unlist(sol2[toselect]),
                    HighAdsorption = unlist(sol3[toselect]))*convert

# Show results
knitr::kable(BUDGET, digit = 2)
```

	NoAdsorption	MediumAdsorption	HighAdsorption
TotalPmin	3.44	3.44	3.44
TotalPdesorp	0.00	0.23	0.67
TotalPadsorp	0.00	0.34	1.00
PO4.SWI.Flux	-3.44	-3.33	-3.12
PO4.Deep.Flux	0.00	0.00	0.00
Pads.SWI.Flux	0.00	0.00	0.00
Pads.Deep.Flux	0.00	0.11	0.32
POP.SWI.Flux	3.44	3.44	3.44
POP.Deep.Flux	0.00	0.00	0.00

By integrating the rates over the entire spatial domain, and by considering the fluxes at the upper and lower boundaries, we construct a conceptual diagram for P flow in the modelled sediment column as shown in Figure 1.

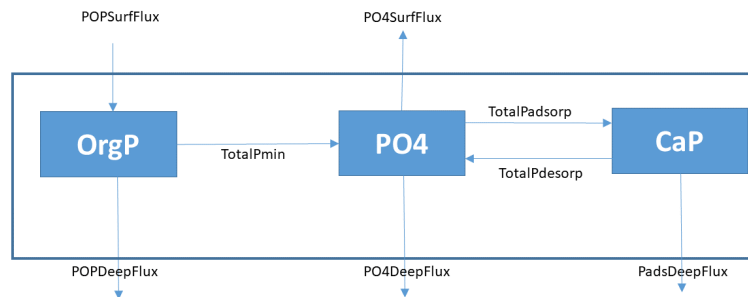


Figure 1: Integrated budgets. Numbers to be added from the table above (or below).

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. <https://CRAN.R-project.org/package=rootSolve>
- 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.
- Soetaert, Karline and Thomas Petzoldt (2020). marelac: Tools for Aquatic Sciences. R package version 2.1.10. <https://CRAN.R-project.org/package=marelac>