Model_with_VC&T_correction

Aya Bahi

08/02/2022

Description

-This file is about the building of the PESTIPOND model.

Set the file and call libraries

Read the data

```
t1 <- Sys.time()

data_1 <- read.csv2("data_1.csv", header = FALSE, skip=0, nrows = 101)          #read the CSV file of d
names(data_1)=data_1[1,]                                                        #rename the columns of
data_1 <- data_1[-1,1:11]                                                       #delete the first line
row.names(data_1)= data_1[,2]                                                   #rename the lines of da

data_1$Date       <- as.Date(data_1$Day , format = "%d/%m/%Y")                  #convert day (d/m/y) to
data_1            <- data_1[,-1]                                                #remove the old date co
data_1            <- as.data.frame(data_1)                                      #convert to a data fram
data_1            <- data_1[,c(11,1:10)]                                        #reorder columns

for (i in 2:11){

  data_1[,i] <- as.numeric(data_1[,i])                                          #convert to numeric

}
data_1 <- as.data.frame(data_1)

A         <- 5269  #m2
Mveg      <- 0.6*11e6/100 #g
```

Build the input functions We don't need to calculate hw with a water balance, we can deduce it from hw = H - ds; with H=1m, and ds is deducted from Msed

```
Qin       <- approxfun(data_1$`Step i`, data_1$`Qin (l/s)`, method = 'constant', rule = 2)
Qout      <- approxfun(data_1$`Step i`, data_1$`Qout (l/s)`,method = 'constant', rule = 2)
TSS_in    <- approxfun(data_1$`Step i`, data_1$`TSS_in (g/l)`,method = 'constant', rule = 2)
TSS_out   <- approxfun(data_1$`Step i`, data_1$`TSS_out (g/l)`,method = 'constant', rule = 2)
Cin       <- approxfun(data_1$`Step i`, data_1$`Cin (ug/l)`,method = 'constant', rule = 2)
Cp_in     <- approxfun(data_1$`Step i`, data_1$`Cp_in(ug/g)`,method = 'constant', rule = 2)
TC        <- approxfun(data_1$`Step i`, data_1$`T (°C)`,method = 'constant', rule = 2)

inputs                 <- NULL
```

```
inputs$'time'           <- seq(1,100,by=0.01)

inputs                  <- as.data.frame(inputs)

inputs$'TC'             <- TC(inputs$'time')
inputs$'Qin(l/s)'       <- Qin(inputs$'time')
inputs$'Qout(l/s)'      <- Qout(inputs$'time')
inputs$'TSS_in (g/l)'   <- TSS_in(inputs$'time')
inputs$'TSS_out (g/l)'  <- TSS_out(inputs$'time')
inputs$'Cdiss_in(ug/l)'<- Cin(inputs$'time')
inputs$'Cp_in(ug/g)'    <- Cp_in(inputs$'time')
```

Write down the differential equations

-The pesticides fate is described with a mass balance equation.

-In this test case we will consider all of the compartments of the pond:

- Dissolved pesticides
- Particulate pesticides
- Sediments
- Vegetation

-We'll consider all the processes corrected according to vegetative cover (VC) and temperature (teta).

Now we will write down this model equation in R :

```
model <- function(t, y, parms){

    with(as.list(c(parms,y)),{

      dMTSS  <- TSS_in(t) * Qin(t) * 24*3600  -  TSS_out(t) * Qout(t) * 24*3600 -

                vs/(1-Msed/(rhob*A*10^6)) * MTSS + vr/(1-Msed/(rhob*A*10^6)) * Msed  #g/d

      #MTSS(0) and Msed(0) should respect the MB at t=0, ds(t=1day) = 1cm, ds(t=-1day) = 0.

      dMsed  <- vs/(1-Msed/(rhob*A*10^6)) * MTSS - vr/(1-Msed/(rhob*A*10^6)) * Msed  #g/d


      dCdiss <- 1/(A*(1-Msed/(rhob*A*10^6))*phi) * Qin(t) * Cin(t)*24*3600/1000- 1/(A*(1-Msed/(rhob*A*1

                (kbio_w + kp + kv) * teta ^ (TC(t)-20) *  Cdiss - 1/(1-Msed/(rhob*A*10^6)) *(-dMsed/(rh

                kads_sed*1e3/(Kd) * Cs + kads_TSS*1e3/(Kd) * Cp + kads_veg*1e3/(Kd*phi) * Cv -

                (kads_sed + kads_TSS + kads_veg) * Cdiss


      dCp    <- 1/(MTSS*phi) * TSS_in(t) * Qin(t) * Cp_in(t)*24*3600- 1/(MTSS*phi) * TSS_out(t) * Qout(

                1/MTSS * dMsed * Cp - kads_TSS * (1-Msed/(rhob*A*10^6)) * A/(MTSS*Kd*1e-6) * Cp +

                1/MTSS * kads_TSS * A * (1-Msed/(rhob*A*10^6)) * 1000 * Cdiss -
                1/(1-Msed/(rhob*A*10^6)) * vs * Cp + Msed/MTSS * 1/(1-Msed/(rhob*A*10^6)) * vr * Cs


      dCs    <- -kbio_sed * teta ^ (TC(t)-20) * Cs - 1/Msed * dMsed * Cs - kads_sed * (1-Msed/(rhob*A*10

                1/Msed * kads_sed * A * (1-Msed/(rhob*A*10^6)) * 1000 * Cdiss + MTSS/Msed * 1/(1-Msed/(

      dCv    <- -kbio_veg * teta ^ (TC(t)-20) * Cv - kads_veg * (1-Msed/(rhob*A*10^6)) * A/(Mveg*Kd*1e-

                1/Mveg * kads_veg * A * (1-Msed/(rhob*A*10^6)) * 1000 * phi * Cdiss

      list(c(dMTSS,dMsed,dCdiss,dCp,dCs,dCv))

    })
}

# ug/l/day
# ug/g/day
```

Parameters

```
parms <- c(vs = 0.05087742, vr = 2.324515e-05,kbio_w=0.03468461,kbio_TSS=0.02818862,kbio_sed=0.05731539
           kads_sed=1.198006, kads_TSS=1.047165, kads_veg=0.3559337, Kd=22,
           teta=1.182469, phi=0.7746798, rhob=0.9819487, A=5269, Mveg=0.6*11e6/100)

# PAR <- as.data.frame(readRDS('C:/aya_local/Model/R_model_new/SA_OUTPUTS_15/16997.RDS')[[1]])
#
# parms <- c(vs = PAR$vs, vr = PAR$vr,kbio_w=PAR$kbio_w,kbio_TSS=PAR$kbio_TSS,kbio_sed=PAR$kbio_sed,kbi
#             kads_sed=PAR$kads_sed, kads_TSS=PAR$kads_TSS, kads_veg=PAR$kads_veg, Kd=PAR$Kd,
#             teta=PAR$teta, phi=PAR$phi, rhob=PAR$rhob, A=5269, Mveg=0.6*11e6/100)


#kdes=kads *A*hw/(Msed*Kd*1e-6)
#Kd = Koc * OC(%), Koc= 850L/kg, log Kow = 3.2
#parms_S <- c(vs=0.1, vr=0.00001)
#yin     <- c(MTSS= vr*0.01*rhob*A*10^4*ds1/(1-vr*0.01),Msed= rhob*A*10^4*ds1)
```

Time We want the concentration C (ug/l & ug/g) to be calculated at every t = 0.01 day

```
times <- seq(1, 100, by = 0.01)
```

Initial values

```
ds1 <- 1
yin <- c(MTSS=parms[[2]]*0.01*parms[[15]]*A*10^4*ds1/(1-parms[[2]]*0.01),
         Msed=round(parms[[15]],6)*A*10^4*ds1,
         Cdiss=0,
         Cp=0,
         Cs=0,
         Cv=0) #g #ug/l #ug/g
```

For this case we choose lsoda solver :

ODE solving

```
out <- as.data.frame(ode(yin, times, model, parms, method = 'lsoda', verbose = T, atol=1e-12, rtol=1e-12


#saveRDS(out, paste0(getwd(),'/out.RDS'))

print(summary(out))

t2 <- Sys.time()

difftime(t2,t1)

#out <- as.data.frame(readRDS('C:/aya_local/Model/R_model_new/SA_OUTPUTS_15/16997.RDS')[[2]])
```

Plot output

```
par(mfrow=c(2,2))
plot(times,out$Cdiss, col='blue',type='l',lwd=2, main='Water',ylab='ug/l', xlab = 'days')
plot(times,out$Cp, col='brown',type='l',lwd=2, main='Suspened solids',ylab='ug/g', xlab='days')
plot(times,out$Cs, col='red',type='l',lwd=2, main='Sediments',ylab='ug/g', xlab='days')
plot(times,out$Cv, col='green',type='l',lwd=2, main='Vegetation',ylab='ug/g', xlab='days')
```

Mass calculation

```
## Mass error  ####

vs       <- parms[[1]]
vr       <- parms[[2]]
kbio_w   <- parms[[3]]
kbio_TSS <- parms[[4]]
kbio_sed <- parms[[5]]
kbio_veg <- parms[[6]]
kp       <- parms[[7]]
kv       <- parms[[8]]
kads_sed <- parms[[9]]
kads_TSS <- parms[[10]]
kads_veg <- parms[[11]]
Kd       <- parms[[12]]
teta     <- parms[[13]]
phi      <- parms[[14]]
rhob     <- parms[[15]]

Mass           <- NULL
Mass$'time'    <- times
Mass$'TC'      <- TC(times)

Mass           <- as.data.frame(Mass)

Mw             <- NULL
Mp             <- NULL
Ms             <- NULL
Mv             <- NULL
```

```
MT              <- NULL

Mdiss_in        <- NULL
Mdiss_out       <- NULL
Mp_in           <- NULL
Mp_out          <- NULL
MT_in           <- NULL
MT_out          <- NULL

for ( i in 1:length(times)) {

  Mw[i] <- out$Cdiss[i] * A * (1-out$Msed[i]/(rhob*A*10^6)) * phi * 1000 #ug  #we calculate the net mas

  Mp[i] <- out$Cp[i] * out$MTSS[i] * phi   #ug

  Ms[i] <- out$Cs[i] * out$Msed[i] * phi    #ug

  Mv[i] <- out$Cv[i] * Mveg    #ug

  MT[i] <- Mw[i] + Mp[i] + Ms[i] + Mv[i]
}

Mass$'Mw' <- Mw
Mass$'Mp' <- Mp
Mass$'Ms' <- Ms
Mass$'Mv' <- Mv
Mass$'MT' <- MT

for (i in 1:length(times)){

  Mdiss_in[i]  <- Cin(times[i]) * Qin(times[i]) * 24 * 3600 /99    #ug/0.01day

  #we set Mdiss_in(t=0)=0 to respect MB, that's why we divide the inlet mass of 10ug/L per 99 to obtain

  Mdiss_out[i] <- out$Cdiss[i]  * Qout(times[i]) * 24 * 3600 * 0.01

  #we divide the equivalent mass of 150ug/g per 100 to obtain an instantaneous mass (per 0.01day)

  Mp_in[i]     <- Cp_in(times[i]) * TSS_in(times[i]) * Qin(times[i]) * 24 * 3600/99  #ug

  #same goes for the particulate inlet/outlet mass

  Mp_out[i]    <- out$Cp[i] * TSS_out(times[i]) * Qout(times[i]) * 24 *3600 * 0.01

  MT_in[i]     <- Mdiss_in[i] + Mp_in[i] #ug
  MT_out[i]    <- Mdiss_out[i] + Mp_out[i] #ug

}

#even the initial conditions should respect the MB (M0 [in the pond]=M-1+Min0-Mout0-Mdeg0),if C(t=0)=0

Mdiss_in[1] <- 0
Mp_in[1]    <- 0
```

```r
MT_in[1]    <- 0

Mass$'Min'      <- Mdiss_in
Mass$'Mout'     <- Mdiss_out
Mass$'Mp_in'    <- Mp_in
Mass$'Mp_out'   <- Mp_out
Mass$'MT_in'    <- MT_in
Mass$'MT_out'   <- MT_out

Mdeg     <- NULL
Mdegw    <- NULL
Mdegp    <- NULL
Mdegs    <- NULL
Mdegv    <- NULL

Mads_sed <- NULL
Mads_TSS <- NULL
Mads_veg <- NULL

Mdes_sed <- NULL
Mdes_TSS <- NULL
Mdes_veg <- NULL

M_sett   <- NULL
M_res    <- NULL

TE <- Mass$'TC'

for (i in 1:length(times)){

  Mdegw[i]    <- (kbio_w + kp + kv) * teta^(TE[i]-20) *  Mw[i] * 0.01 #we correct degradation according

  Mdegp[i]    <- kbio_TSS * teta^(TE[i]-20)  * Mp[i] * 0.01

  Mdegs[i]    <- kbio_sed * teta^(TE[i]-20) * Ms[i] * 0.01

  Mdegv[i]    <- kbio_veg * teta^(TE[i]-20) * Mv[i] * 0.01

  Mdeg[i]     <- Mdegw[i] + Mdegp[i] + Mdegs[i] + Mdegv[i]  #ug #mdeg = dm/dt

  Mads_sed[i] <- kads_sed * Mw[i] * 0.01 #adsorbed during 0.01 day

  Mads_TSS[i] <- kads_TSS * Mw[i] * 0.01

  Mads_veg[i] <- kads_veg * Mw[i] * 0.01

  Mdes_sed[i] <- kads_sed * A * (1-out$Msed[i]/(rhob*A*10^6))/(out$Msed[i]*Kd*1e-6) * Ms[i] * 0.01 #ug

  Mdes_TSS[i] <- kads_TSS * A * (1-out$Msed[i]/(rhob*A*10^6))/(out$MTSS[i]*Kd*1e-6) * Mp[i] * 0.01

  Mdes_veg[i] <- kads_veg * A * (1-out$Msed[i]/(rhob*A*10^6))/(Mveg*Kd*1e-6) * Mv[i] * 0.01

  M_sett[i]  <- vs * 1/(1-out$Msed[i]/(rhob*A*10^6)) * Mp[i] * 0.01
  M_res[i]   <- vr * 1/(1-out$Msed[i]/(rhob*A*10^6)) * Ms[i] * 0.01

}

Mass$'Mdeg'  <- Mdeg
Mass$'Mdegw' <- Mdegw
Mass$'Mdegp' <- Mdegp
Mass$'Mdegs' <- Mdegs
Mass$'Mdegv' <- Mdegv

Mass$'Mads_sed' <- Mads_sed
Mass$'Mads_TSS' <- Mads_TSS
Mass$'Mads_veg' <- Mads_veg

Mass$'Mdes_sed' <- Mdes_sed
Mass$'Mdes_TSS' <- Mdes_TSS
Mass$'Mdes_veg' <- Mdes_veg

Mass$'M_sett'   <- M_sett
Mass$'M_res'    <- M_res

Mass <- round(Mass,4)

write_xlsx(Mass,"C:\\aya_local\\Model\\Resultats\\Mass_JDD.xlsx")
```

In this section we estimate the mass balance error err(%):

Total mass balance error

```
e <- NULL
N <- length(Mass$time)

e[1] <- 0

#instantaneous MB error of the total mass of pesticides in the pond

for (i in 2:N){

  #no need to estimate the error at the initial step because we already know the outputs there (i.e., i

  e[i] <- 100*(Mass$MT[i]-Mass$MT[1]-sum(Mass$MT_in[2:i])+sum(Mass$MT_out[2:i])+sum(Mass$Mdeg[2:i]))/su
}

Mass$'err' <- e

summary(abs(Mass$'err'))

#total MB error of the total mass of pesticides in the pond

MBE <- 100*(Mass$MT[N]-Mass$MT[1]-sum(Mass$MT_in[2:N])+sum(Mass$MT_out[2:N])+sum(Mass$Mdeg[2:N]))/sum(Ma

print(paste('MBE (%) = ',MBE))
```

Solids mass balance error

```
MTSS_in       <- NULL
MTSS_out      <- NULL

Msett         <- NULL
Mres          <- NULL

NS <- length(times)

for (i in 1:NS){

  MTSS_in[i]    <- TSS_in(times[i])   * Qin(times[i]) * 24 * 3600 * 0.01                 #g
  MTSS_out[i]   <- TSS_out(times[i])  * Qout(times[i]) * 24 * 3600 * 0.01
}

for (i in 1:NS){

  Msett[i]     <- vs * 1/(1-out$Msed[i]/(rhob*A*10^6)) * out$MTSS[i] * 0.01                       #g
  Mres[i]      <- vr * 1/(1-out$Msed[i]/(rhob*A*10^6)) * out$Msed[i] * 0.01
}

#the solving scheme is forward Mi+1 = Mi + Mini - Mouti

Msolids    <- out$MTSS + out$Msed

MBE_solids <- 100*(Msolids[NS] - Msolids[1]- sum(MTSS_in[1:NS-1]) + sum(MTSS_out[1:NS-1]))/sum(MTSS_in[

print(paste('MBE_solids (%) = ', MBE_solids ))
```

MBE for water compartment

```
e_w    <- NULL
e_w[1] <- 0

#instantaneous MB error of the mass of dissolved pesticides in water

for (i in 2:N){

  #no need to estimate the error at the initial step because we already know the outputs there (i.e.ini

  e_w[i] <- 100*(Mass$Mw[i]-Mass$Mw[1]-sum(Mass$Min[2:i])-sum(Mass$Mdes_sed[2:i])-sum(Mass$Mdes_TSS[2:i

            sum(Mass$Mout[2:i])+sum(Mass$Mdegw[2:i])+sum(Mass$Mads_sed[2:i])+sum(Mass$Mads_TSS[2:i])+su

            (sum(Mass$Min[2:i])+sum(Mass$Mdes_sed[2:i])+sum(Mass$Mdes_TSS[2:i])+sum(Mass$Mdes_veg[2:i])
}

Mass$'err_w' <- e_w
summary(abs(Mass$'err_w'))

#total MB error of the mass of dissolved pesticides in water

MBE_w <- 100*(Mass$Mw[N]-Mass$Mw[1]-sum(Mass$Min[2:N])-sum(Mass$Mdes_sed[2:N])-sum(Mass$Mdes_TSS[2:N])-
         sum(Mass$Mout[2:N])+sum(Mass$Mdegw[2:N])+sum(Mass$Mads_sed[2:N])+sum(Mass$Mads_TSS[2:N])+sum(Ma

         (sum(Mass$Min[2:N])+sum(Mass$Mdes_sed[2:N])+sum(Mass$Mdes_TSS[2:N])+sum(Mass$Mdes_veg[2:N]))
```

MBE for TSS compartment

```
e_p    <- NULL
e_p[1] <- 0

#instantaneous MB error of the mass of particulate pesticides in water

for (i in 2:N){

  e_p[i] <- 100*(Mass$Mp[i]-Mass$Mp[1]-sum(Mass$Mp_in[2:i])-sum(Mass$Mads_TSS[2:i])+

            sum(Mass$Mp_out[2:i])+sum(Mass$Mdegp[2:i])+sum(Mass$Mdes_TSS[2:i])+sum(Mass$M_sett[2:i])-sum

           (sum(Mass$Mp_in[2:i])+sum(Mass$Mads_TSS[2:i])+sum(Mass$M_res[2:i]))
}

Mass$'err_p' <- e_p
summary(abs(Mass$'err_p'))

#total MB error of the mass of particulate pesticides in water

MBE_p <- 100*(Mass$Mp[N]-Mass$Mp[1]-sum(Mass$Mp_in[2:N])-sum(Mass$Mads_TSS[2:N])+

         sum(Mass$Mp_out[2:N])+sum(Mass$Mdegp[2:N])+sum(Mass$Mdes_TSS[2:N])+sum(Mass$M_sett[2:N])-sum(Ma

         (sum(Mass$Mp_in[2:N])+sum(Mass$Mads_TSS[2:N])+sum(Mass$M_res[2:N]))
```

MBE for sediments compartment

```
e_s    <-NULL
e_s[1] <- 0

#instantaneous MB error of the mass of pesticides in sediments

for (i in 2:N){


  e_s[i] <- 100*(Mass$Ms[i]-Mass$Ms[1]-sum(Mass$Mads_sed[2:i])+sum(Mass$Mdegs[2:i])+sum(Mass$Mdes_sed[2

            sum(Mass$M_sett[2:i])+sum(Mass$M_res[2:i]))/(sum(Mass$Mads_sed[2:i])+sum(Mass$M_sett[2:i]))
}

Mass$'err_s' <- e_s
summary(abs(Mass$'err_s'))

#total MB error of the mass of pesticides in sediments

MBE_s <- 100*(Mass$Ms[N]-Mass$Ms[1]-sum(Mass$Mads_sed[2:N])+sum(Mass$Mdegs[2:N])+sum(Mass$Mdes_sed[2:N]
         sum(Mass$M_sett[2:N])+sum(Mass$M_res[2:N]))/(sum(Mass$Mads_sed[2:N])+sum(Mass$M_sett[2:N]))
```

MBE for vegetation compartment

```
e_v    <- NULL
e_v[1] <- 0

#instantaneous MB error of the mass of pesticides in vegetation

for (i in 2:N){

  e_v[i] <- 100*(Mass$Mv[i]-Mass$Mv[1]-sum(Mass$Mads_veg[2:i])+sum(Mass$Mdegv[2:i])+sum(Mass$Mdes_veg[2
}

Mass$'err_v' <- e_v
summary(abs(Mass$'err_v'))

#total MB error of the mass of pesticides in vegetation

MBE_v <- 100*(Mass$Mv[N]-Mass$Mv[1]-sum(Mass$Mads_veg[2:N])+sum(Mass$Mdegv[2:N])+sum(Mass$Mdes_veg[2:N]
```

Total MBE in each compartment

```
cat('MBE_solids (%) = ', MBE_solids, '\n',
    'MBE_w (%) =',MBE_w,'\n',
    'MBE_p (%) = ', MBE_p,'\n',
    'MBE_s (%) = ', MBE_s, '\n',
    'MBE_v (%) = ', MBE_v, '\n',
    'MBE (%)   =  ', MBE, '\n')
```