ADP\_TotChla

Qing\_Zhan

19/02/2021

# Library

require(lubridate)

## Loading required package: lubridate

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':  
##   
## date

require(dplyr)

## Loading required package: dplyr

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:lubridate':  
##   
## intersect, setdiff, union

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

# Input data

### Water Budget ####  
WaterBudget <- read.csv("C:\\Users\\QingZ\\PhD-Qing\\GitProject\\Mark\_Dintel\_Vliet\\Modelling\\Advection-Diffusion-Production\\Waterbalance\\WaterBudget.csv")  
  
sTime=WaterBudget$sTime  
  
### Input data preparation ####  
HF\_BlueCh\_MarkCanal <- read.csv("C:\\Users\\QingZ\\PhD-Qing\\GitProject\\Mark\_Dintel\_Vliet\\Modelling\\Advection-Diffusion-Production\\HF\_BlueCh\_MarkCanal.csv")  
HF\_BlueCh\_DintelCanal <- read.csv("C:\\Users\\QingZ\\PhD-Qing\\GitProject\\Mark\_Dintel\_Vliet\\Modelling\\Advection-Diffusion-Production\\HF\_BlueCh\_DintelCanal.csv")  
HF\_BlueCh\_Vliet <- read.csv("C:\\Users\\QingZ\\PhD-Qing\\GitProject\\Mark\_Dintel\_Vliet\\Modelling\\Advection-Diffusion-Production\\HF\_BlueCh\_Vliet.csv")  
  
HF\_Chla\_MarkCanal <- read.csv("C:\\Users\\QingZ\\PhD-Qing\\GitProject\\Mark\_Dintel\_Vliet\\Modelling\\Advection-Diffusion-Production\\HF\_Chla\_MarkCanal.csv")  
HF\_Chla\_DintelCanal <- read.csv("C:\\Users\\QingZ\\PhD-Qing\\GitProject\\Mark\_Dintel\_Vliet\\Modelling\\Advection-Diffusion-Production\\HF\_Chla\_DintelCanal.csv")  
HF\_Chla\_Vliet <- read.csv("C:\\Users\\QingZ\\PhD-Qing\\GitProject\\Mark\_Dintel\_Vliet\\Modelling\\Advection-Diffusion-Production\\HF\_Chla\_Vliet.csv")  
  
mL <- read.csv("C:\\Users\\QingZ\\PhD-Qing\\GitProject\\Mark\_Dintel\_Vliet\\data\\MeteorologicalData\\Radiance.csv")  
mL$Time <- paste(mL$YYYYMMDD, mL$HH)%>%ymd\_h%>%as.numeric  
mL <- approx(x=mL$Time,y=mL$Q,xout=WaterBudget$sTime)%>%as.data.frame()  
  
MDV\_data <- data.frame(sTime=WaterBudget$sTime, Date=WaterBudget$Date%>%ymd\_hms, Temp=NA, Qin\_MarkCanal=WaterBudget$Qin\_MarkCanal, Qin\_Oranjeboombrug=WaterBudget$Qin\_Oranjeboombrug, Qin\_BlauweKamer=WaterBudget$Qin\_BlauweKamer, Qin\_Molenbeak=WaterBudget$Qin\_Molenbeak, Qout\_Dintelsas=WaterBudget$Qout\_Dintelsas, Qout\_Vliet=WaterBudget$Qout\_Vliet, Qout\_Polder=WaterBudget$Qout\_Polder, BlueCh\_MarkCanal=NA, BlueCh\_DintelCanal=NA, BlueCh\_Vliet=NA, TotChla\_MarkCanal=NA, TotChla\_DintelCanal=NA, TotChla\_Vliet=NA, mL=NA)  
  
# global radiance:  
MDV\_data$mL <- mL$y[which(mL$x%in%MDV\_data$sTime)]  
  
# Temperature data  
HF\_Temp <- read.csv("C:\\Users\\QingZ\\PhD-Qing\\GitProject\\Mark\_Dintel\_Vliet\\Modelling\\Advection-Diffusion-Production\\HF\_Temp\_MarkCanal.csv")  
MDV\_data$Temp <- approx(HF\_Temp$Datum%>%dmy\_hms%>%as.numeric, HF\_Temp$Waarde, xout=MDV\_data$sTime)$y

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to  
## unique 'x' values

# Cyanobacteria measurements  
MDV\_data$BlueCh\_MarkCanal <- approx(HF\_BlueCh\_MarkCanal$Datum%>%dmy\_hms%>%as.numeric, HF\_BlueCh\_MarkCanal$Waarde, xout=WaterBudget$sTime)$y  
MDV\_data$BlueCh\_MarkCanal[which(MDV\_data$BlueCh\_MarkCanal<0)]<-0  
MDV\_data$BlueCh\_DintelCanal <- approx(HF\_BlueCh\_DintelCanal$Datum%>%dmy\_hms%>%as.numeric, HF\_BlueCh\_DintelCanal$Waarde, xout=sTime)$y  
MDV\_data$BlueCh\_DintelCanal[which(MDV\_data$BlueCh\_DintelCanal<0)]<-0  
MDV\_data$BlueCh\_Vliet <- approx(HF\_BlueCh\_Vliet$Datum%>%dmy\_hms%>%as.numeric, HF\_BlueCh\_Vliet$Waarde, xout=sTime)$y  
MDV\_data$BlueCh\_Vliet[which(MDV\_data$BlueCh\_Vliet<0)]<-0  
  
# Chla measurements  
MDV\_data$TotChla\_MarkCanal <- approx(HF\_Chla\_MarkCanal$Datum%>%dmy\_hms%>%as.numeric, HF\_Chla\_MarkCanal$Waarde, xout = sTime)$y

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to  
## unique 'x' values

MDV\_data$TotChla\_MarkCanal[which(MDV\_data$Chla\_MarkCanal<0)]<-0  
MDV\_data$TotChla\_DintelCanal <- approx(HF\_Chla\_DintelCanal$Datum%>%dmy\_hms%>%as.numeric, HF\_Chla\_DintelCanal$Waarde, xout = sTime)$y

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to  
## unique 'x' values

MDV\_data$TotChla\_DintelCanal[which(MDV\_data$Chla\_DintelCanal<0)]<-0  
MDV\_data$TotChla\_Vliet <- approx(HF\_Chla\_Vliet$Datum%>%dmy\_hms%>%as.numeric, HF\_Chla\_Vliet$Waarde, xout = sTime)$y

## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to  
## unique 'x' values

MDV\_data$TotChla\_Vliet[which(MDV\_data$Chla\_Vliet<0)]<-0  
  
MDV\_data$dChla\_MarkCanal <- MDV\_data$TotChla\_MarkCanal-MDV\_data$BlueCh\_MarkCanal  
MDV\_data$dChla\_MarkCanal[which(MDV\_data$dChla\_MarkCanal<0)]<-0  
MDV\_data$dChla\_DintelCanal <- MDV\_data$TotChla\_DintelCanal-MDV\_data$BlueCh\_DintelCanal  
MDV\_data$dChla\_DintelCanal[which(MDV\_data$dChla\_DintelCanal<0)]<-0  
MDV\_data$dChla\_Vliet <- MDV\_data$TotChla\_Vliet-MDV\_data$BlueCh\_Vliet  
MDV\_data$dChla\_Vliet[which(MDV\_data$dChla\_Vliet<0)] <- 0  
  
MDV\_data <- MDV\_data%>%na.omit  
  
#sTime=seq(mL2018$Time[2],mL2018$Time[length(mL2018$Time)],by=60\*30)  
  
  
# Nutrient data ####  
PO4\_MarkCanal <- read.csv("C:\\Users\\QingZ\\PhD-Qing\\GitProject\\Mark\_Dintel\_Vliet\\Modelling\\Advection-Diffusion-Production\\LF\_PO4\_MarkCanal.csv")  
PO4\_DintelCanal <- read.csv("C:\\Users\\QingZ\\PhD-Qing\\GitProject\\Mark\_Dintel\_Vliet\\Modelling\\Advection-Diffusion-Production\\LF\_PO4\_DintelCanal.csv")  
PO4\_MVCanal <- read.csv("C:\\Users\\QingZ\\PhD-Qing\\GitProject\\Mark\_Dintel\_Vliet\\Modelling\\Advection-Diffusion-Production\\LF\_PO4\_MVCanal.csv")  
PO4\_Vliet <- read.csv("C:\\Users\\QingZ\\PhD-Qing\\GitProject\\Mark\_Dintel\_Vliet\\Modelling\\Advection-Diffusion-Production\\LF\_PO4\_Vliet.csv")  
  
MDV\_data$PO4\_MarkCanal <- NA  
MDV\_data$PO4\_DintelCanal <- NA  
MDV\_data$PO4\_MVCanal <- NA  
MDV\_data$PO4\_Vliet <- NA  
  
PO4\_MarkCanal<-approx(PO4\_MarkCanal$Datum%>%dmy\_hms%>%as.numeric,PO4\_MarkCanal$Waarde, xout=sTime)%>%as.data.frame()  
MDV\_data$PO4\_MarkCanal[which(MDV\_data$sTime%in%PO4\_MarkCanal$x)]<- PO4\_MarkCanal$y[which(PO4\_MarkCanal$x%in%MDV\_data$sTime)]  
PO4\_DintelCanal <- approx(PO4\_DintelCanal$Datum%>%dmy\_hms%>%as.numeric,PO4\_DintelCanal$Waarde, xout = sTime)%>%as.data.frame()  
MDV\_data$PO4\_DintelCanal[which(MDV\_data$sTime%in%PO4\_DintelCanal$x)] <- PO4\_DintelCanal$y[which(PO4\_DintelCanal$x%in%MDV\_data$sTime)]  
PO4\_MVCanal <- approx(PO4\_MVCanal$Datum%>%dmy\_hms%>%as.numeric,PO4\_MVCanal$Waarde, xout = sTime)%>%as.data.frame()  
MDV\_data$PO4\_MVCanal[which(MDV\_data$sTime%in%PO4\_MVCanal$x)] <- PO4\_MVCanal$y[which(PO4\_MVCanal$x%in%MDV\_data$sTime)]  
PO4\_Vliet <- approx(PO4\_Vliet$Datum%>%dmy\_hms%>%as.numeric,PO4\_Vliet$Waarde,xout=sTime)%>%as.data.frame()  
MDV\_data$PO4\_Vliet[which(MDV\_data$sTime%in%PO4\_Vliet$x)] <- PO4\_Vliet$y[which(PO4\_Vliet$x%in%MDV\_data$sTime)]  
  
MDV\_data <- na.omit(MDV\_data)

# Parameter set-up in model

Num\_Place\_Steps <- 6  
Num\_Time\_Steps <- length(MDV\_data$sTime)#1e5  
Time\_step <- 60\*30 # delta t, at which derivative will be calculated  
  
# Coversion parameters  
SecondPerDay <- 86400 # seconds per day  
TenDays <- 10\*SecondPerDay # s  
DaysPerYear <- 365 # d  
gPermg <- 0.001  
  
#Place\_step <- 50 # meter  
Dispersion <- 7.5 # m2/s  
  
# Chlorophyll/C  
cChDDiat <- (0.004+0.012)/2  
cChDGren <- (0.01+0.02)/2  
cChDBlue <- (0.005+0.015)/2  
cChD.vec <- c(BlueCh=cChDBlue, GreenCh=cChDGren, DiatCh=cChDDiat)  
#(cChDDiat+cChDGren+cChDBlue)  
  
# maximum growth rate  
cMuMaxDiat <- 2 #(day^-1)  
cMuMaxGren <- 1.5 #(day^-1)  
cMuMaxBlue <- 0.6 #(day^-1)  
cMuMax.vec <- c(BlueCh=cMuMaxBlue, GreenCh=cChDGren, DiatCh=cChDDiat)  
#((cMuMaxDiat+cMuMaxGren+cMuMaxBlue)/3) # /delta t  
  
# maintenance respiration constant rate  
kDRespDiat <- 0.1 #(day^-1)  
kDRespGren <- 0.075 #(day^-1)  
kDRespBlue <- 0.03 #(day^-1)  
kDResp.vec <- c(BlueCh=kDRespBlue, GreenCh=kDRespGren, DiatCh=kDRespDiat)  
#((kDRespDiat+kDRespGren+kDRespBlue)/3) # /delta t  
  
# mortality constant  
kMortDiat <- 0.01 #(day^-1)  
kMortGren <- 0.01 #(day^-1)  
kMortBlue <- 0.01 #(day^-1)  
kMort.vec <- c(BlueCh=kMortBlue, GreenCh=kMortGren, DiatCh=kMortDiat)  
#((kMortDiat+kMortGren+kMortBlue)/3) /delta t  
  
# grazing loss rate  
kLossDiat <- 0.25 #(day^-1)  
kLossGren <- 0.25 #(day^-1)  
kLossBlue <- 0.03 #(day^-1)  
kLoss.vec <- c(BlueCh=kLossBlue, GreenCh=kLossGren, DiatCh=kLossDiat)  
#((kLossDiat+kLossBlue+kLossGren)/3) # /delta t  
  
# Temperature function  
cTmRef <- 20  
  
cTmOptDiat <- 18  
cTmOptGren <- 25  
cTmOptBlue <- 25  
cTmOpt.vec <- c(BlueCh=cTmOptBlue, GreenCh=cTmOptGren, DiatCh=cTmOptDiat)  
  
cSigTmDiat <- 20 # sigma in Gaussian curve  
cSigTmGren <- 15  
cSigTmBlue <- 12  
cSigTm.vec <- c(BlueCh=cSigTmBlue, GreenCh=cSigTmGren, DiatCh=cSigTmDiat)  
   
# radiation  
cLDayAve <- 1e7/SecondPerDay #J/m2/s = W/m2  
cLDayVar <- 8e6/SecondPerDay #J/m2/s = W/m2  
fPAR <- 0.48 # fraction photosynthesically active radiation (PAR)  
fRefl <- 0.2 # the fraction photosynthetically active radiation reflected at the surface  
  
# Day length  
cfDayAve <- 0.5  
cfDayVar <- 0.2  
  
# light extinction coeffcients  
cExtWat <- 0.5 # m^-1  
cExtSpDet <- 0.15 # m2/gDW  
cExtSpIM <- 0.05 # m2/gDW  
  
# half-sat PAR at 20 degrees  
hLRefDiat <- 1000 # Fake value, W/m2 = J/m2/s  
hLRefGren <- 17 # W/m2  
hLRefBlue <- 1000 # Fake value, W/m2  
hLRef.vec <- c(BlueCh=hLRefBlue, GreenCh=hLRefGren, DiatCh=hLRefDiat)  
  
cLOptRefDiat <- 54 # W/m2  
cLOptRefGren <- 1000 # W/m2, Fake value  
cLOptRefBlue <- 13.6 # W/m2  
cLOptRef.vec <- c(BlueCh=cLOptRefBlue, GreenCh=cLOptRefGren, DiatCh=cLOptRefDiat)  
  
# half saturation constant of resource  
hR <- 0.18 # mg/l

# function

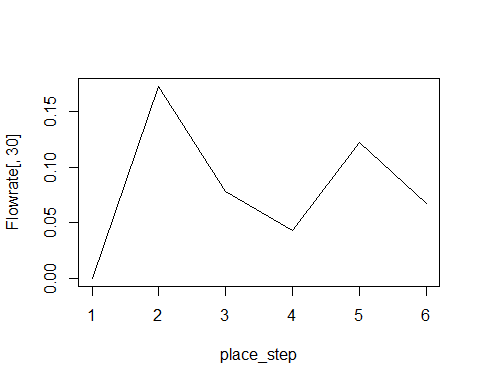
## Temperature function  
TmFunc = function(uTm, cSigTm, cTmOpt){  
 return(exp(-0.5/(cSigTm^2)\*((uTm-cTmOpt)^2-(cTmRef-cTmOpt))))  
}   
  
## Light limitation  
SteeleFunc <- function(aExtCoef, sDepthW, ISurf, uTm, cSigTm, cTmOpt, cLOptRef,TmFunc\_value){  
 IBottom = ISurf\*exp(-aExtCoef\*sDepthW)  
 uLOptRef = cLOptRef\*TmFunc\_value  
 return(exp(1)/(aExtCoef\*sDepthW)\*(exp(-IBottom/uLOptRef)-exp(-ISurf/uLOptRef)))  
} # Tilman (1982), Huisman and Weissing (1994, 1995)  
  
LehmanFunc <- function(aExtCoef,sDepthW,ISurf,uTm, cSigTm, cTmOpt,hLRef,TmFunc\_value){  
 IBottom = ISurf\*exp(-aExtCoef\*sDepthW)  
 uhLRef = hLRef\*TmFunc\_value  
 return(1/(aExtCoef\*sDepthW)\*log((uhLRef\*TmFunc\_value+ISurf)/(uhLRef\*TmFunc\_value+IBottom)))  
}  
  
## Nutrient limitation  
PLimFunc = function(P){return(P/(P+hR))}

#### Morphological size

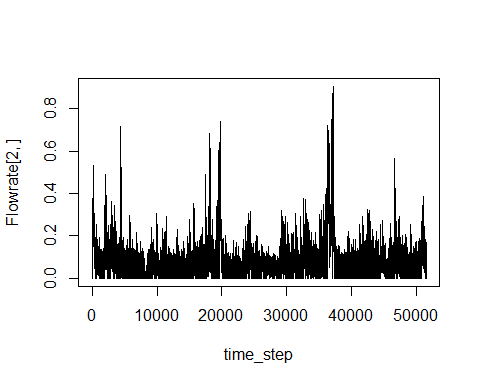
#Conc0 <- 50 # ug/l  
#Dim TotChla(1117,10000)  
#Dim Flowrate(1117)  
#Too\_big\_t\_step <- 0  
#starting\_growth <- 1/24 # ug/l/s: numer ug/l that's added every second  
#Monod\_Speed <- 0.05  
#Monod\_Conc <- 50  
  
#num <- (0.1/2)\*(Place\_step-0.1\*Time\_step)  
  
Begin\_Mark\_canal <- 1  
Width\_Mark\_canal <- 40  
Depth\_Mark\_canal <- 2  
Length\_Mark\_canal <- 114\*50  
  
Begin\_Dintel <- 2 # Place step 114 transition Mark canal to Dintel  
Width\_Dintel <- 50  
Depth\_Dintel <- 3  
Length\_Dintel <- (640-114)\*50  
  
Zevenbergen <- 401  
  
Begin\_Dintel\_wide <- 3 # Here the Dintel becomes much wider  
Width\_Dintel\_wide <- 100  
Depth\_Dintel\_wide <- 3.3  
Length\_Dintel\_wide <- (666-640)\*50  
  
Begin\_MV\_canal <- 4 # place step 666 water disappears in the direction of the non-modelled branch to the discharge sluice\_ Transition to the Mark-Vliet canal profile  
Width\_MV\_canal <- 40  
Depth\_MV\_canal <- 2.8  
Length\_MV\_canal <- (743-666)\*50  
  
Begin\_Vliet\_narrow <- 5  
Width\_Vliet\_narrow <- 26  
Depth\_Vliet\_narrow <- 1.53  
Length\_Vliet\_narrow <- (760-743)\*50  
  
Begin\_Vliet <- 6  
Width\_Vliet <- 30  
Depth\_Vliet <- 2.4  
Length\_Vliet <- (1117-760)\*50

# Flowrate simulation based on mass conservation:

Flowrate <- matrix(data=NA, nrow=Num\_Place\_Steps,ncol = Num\_Time\_Steps,dimnames = list(paste("Place\_step",0:(Num\_Place\_Steps-1)),paste("Time\_step",0:(Num\_Time\_Steps-1))))  
  
for (t in 1:Num\_Time\_Steps) {  
 for (x in Begin\_Mark\_canal) {  
 Flowrate[x,t] = MDV\_data$Qin\_MarkCanal[t]/(Width\_Mark\_canal\*Depth\_Mark\_canal) # m/s  
 }  
   
 for (x in Begin\_Dintel){  
 Flowrate[x,t]=(MDV\_data$Qin\_MarkCanal[t]+MDV\_data$Qin\_Oranjeboombrug[t]+MDV\_data$Qin\_BlauweKamer[t]-MDV\_data$Qout\_Polder[t])/(Width\_Dintel\*Depth\_Dintel) # m/s  
 }  
   
 for (x in Begin\_Dintel\_wide){  
 Flowrate[x,t]=(MDV\_data$Qin\_MarkCanal[t]+MDV\_data$Qin\_Oranjeboombrug[t]+MDV\_data$Qin\_BlauweKamer[t]-MDV\_data$Qout\_Polder[t])/(Width\_Dintel\_wide\*Depth\_Dintel\_wide) # m/s  
 }  
   
 for (x in Begin\_MV\_canal){  
 Flowrate[x,t]=(MDV\_data$Qin\_MarkCanal[t]+MDV\_data$Qin\_Oranjeboombrug[t]+MDV\_data$Qin\_BlauweKamer[t]-MDV\_data$Qout\_Dintelsas[t]-MDV\_data$Qout\_Polder[t])/(Width\_MV\_canal\*Depth\_MV\_canal) # m/s  
 }  
   
 for (x in Begin\_Vliet\_narrow) {  
 Flowrate[x,t]=(MDV\_data$Qin\_MarkCanal[t]+MDV\_data$Qin\_BlauweKamer[t]+MDV\_data$Qin\_Oranjeboombrug[t]-MDV\_data$Qout\_Dintelsas[t]-MDV\_data$Qout\_Polder[t])/(Width\_Vliet\_narrow\*Depth\_Vliet\_narrow) # m/s  
 }  
   
 for (x in Begin\_Vliet){  
 Flowrate[x,t]=(MDV\_data$Qin\_MarkCanal[t]+MDV\_data$Qin\_BlauweKamer[t]+MDV\_data$Qin\_Oranjeboombrug[t]-MDV\_data$Qout\_Dintelsas[t]-MDV\_data$Qout\_Polder[t])/(Width\_Vliet\*Depth\_Vliet) # m/s  
 }  
}  
  
for (i in 1:nrow(Flowrate)) {  
 Flowrate[i,which(Flowrate[i,]<0)]=0  
}  
plot(Flowrate[,30],t="l",xlab="place\_step")

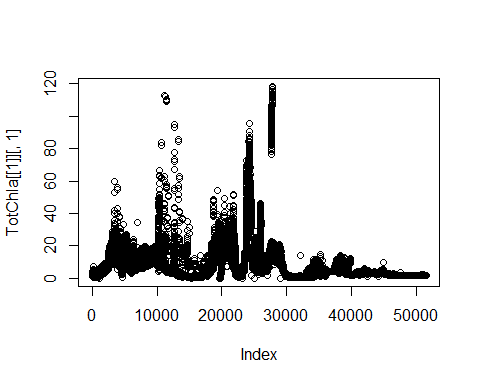


#axis(side = 1,at=c(114,640,666,743,760,1117),labels = c(114,640,666,743,760,1117),col.ticks = "red",las=2,cex=0.7)  
plot(Flowrate[2,],t="l",xlab="time\_step")



# Chlorophyll simulation

## Initial values: ####  
TotChla <- list(  
 BlueCh=matrix(data=NA,nrow=Num\_Time\_Steps,ncol=Num\_Place\_Steps,dimnames = list(paste("Time\_step",0:(Num\_Time\_Steps-1)),paste("Place\_step",0:(Num\_Place\_Steps-1)))),  
 GreenCh=matrix(data=NA,nrow=Num\_Time\_Steps,ncol=Num\_Place\_Steps,dimnames = list(paste("Time\_step",0:(Num\_Time\_Steps-1)),paste("Place\_step",0:(Num\_Place\_Steps-1)))),  
 DiatCh=matrix(data=NA,nrow=Num\_Time\_Steps,ncol=Num\_Place\_Steps,dimnames = list(paste("Time\_step",0:(Num\_Time\_Steps-1)),paste("Place\_step",0:(Num\_Place\_Steps-1))))  
)  
  
TotChla[["BlueCh"]]["Time\_step 0",] =c(  
 MDV\_data$BlueCh\_MarkCanal[1],  
 MDV\_data$BlueCh\_DintelCanal[1],  
 MDV\_data$BlueCh\_DintelCanal[1],  
 MDV\_data$BlueCh\_Vliet[1],  
 MDV\_data$BlueCh\_Vliet[1],  
 MDV\_data$BlueCh\_Vliet[1]  
)  
TotChla[["GreenCh"]]["Time\_step 0",] =c(  
 MDV\_data$BlueCh\_MarkCanal[1],  
 MDV\_data$BlueCh\_DintelCanal[1],  
 MDV\_data$BlueCh\_DintelCanal[1],  
 MDV\_data$BlueCh\_Vliet[1],  
 MDV\_data$BlueCh\_Vliet[1],  
 MDV\_data$BlueCh\_Vliet[1]  
)  
TotChla[["DiatCh"]]["Time\_step 0",] =c(  
 MDV\_data$BlueCh\_MarkCanal[1],  
 MDV\_data$BlueCh\_DintelCanal[1],  
 MDV\_data$BlueCh\_DintelCanal[1],  
 MDV\_data$BlueCh\_Vliet[1],  
 MDV\_data$BlueCh\_Vliet[1],  
 MDV\_data$BlueCh\_Vliet[1]  
)  
  
TotChla[["BlueCh"]]["Time\_step 0",]=c(MDV\_data$BlueCh\_MarkCanal[1], MDV\_data$BlueCh\_DintelCanal[1], MDV\_data$BlueCh\_DintelCanal[1], MDV\_data$BlueCh\_DintelCanal[1], MDV\_data$BlueCh\_Vliet[1], MDV\_data$BlueCh\_Vliet[1])  
  
TotChla[["GreenCh"]]["Time\_step 0",]=c(MDV\_data$dChla\_MarkCanal[1], MDV\_data$dChla\_DintelCanal[1], MDV\_data$dChla\_DintelCanal[1], MDV\_data$dChla\_DintelCanal[1], MDV\_data$dChla\_Vliet[1], MDV\_data$dChla\_Vliet[1])  
  
TotChla[["DiatCh"]]["Time\_step 0",]=c(MDV\_data$dChla\_MarkCanal[1], MDV\_data$dChla\_DintelCanal[1], MDV\_data$dChla\_DintelCanal[1], MDV\_data$dChla\_DintelCanal[1], MDV\_data$dChla\_Vliet[1], MDV\_data$dChla\_Vliet[1])  
   
TotChla[["BlueCh"]][,"Place\_step 0"]=MDV\_data$BlueCh\_MarkCanal  
TotChla[["GreenCh"]][,"Place\_step 0"]=MDV\_data$dChla\_MarkCanal  
TotChla[["DiatCh"]][,"Place\_step 0"]=MDV\_data$dChla\_MarkCanal  
  
plot(TotChla[[1]][,1])



Place\_step <- c(Length\_Mark\_canal, Length\_Dintel, Length\_Dintel\_wide, Length\_MV\_canal, Length\_Vliet\_narrow, Length\_Vliet)  
Place\_Depths <- c(Depth\_Mark\_canal, Depth\_Dintel, Depth\_Dintel\_wide, Depth\_MV\_canal, Depth\_Vliet\_narrow, Depth\_Vliet)  
  
Too\_big\_t\_step <- list(BlueCh=rep(NA,Num\_Place\_Steps),GreenCh=rep(NA,Num\_Place\_Steps),DiatCh=rep(NA,Num\_Place\_Steps))  
Too\_big\_t\_step[["BlueCh"]][1] <- 0  
Too\_big\_t\_step[["GreenCh"]][1] <- 0  
Too\_big\_t\_step[["DiatCh"]][1] <- 0

# State variables to be outputted:

### State variables to be outputted: ####  
#sGrowth <- rep(NA,ncol(TotChla))  
#sDegradation <- rep(NA,ncol(TotChla))  
#sLDay <- rep(NA,ncol(TotChla))  
#sLLim <- rep(NA,ncol(TotChla))  
#sPLim <- rep(NA,ncol(TotChla))  
#sMuMaxTm <- rep(NA,ncol(TotChla))

## Simulation

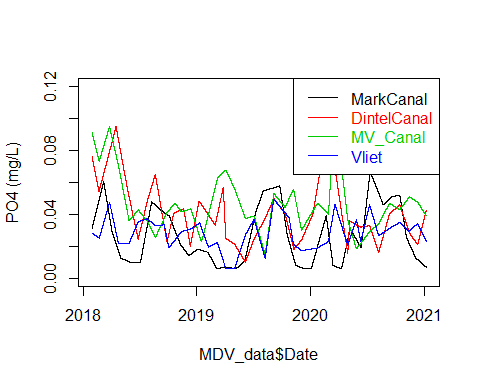
Chla\_vec <- c("BlueCh","GreenCh","DiatCh")  
  
for (Chla in Chla\_vec) {  
 for(t in 2:Num\_Time\_Steps){  
 for(x in 2:Num\_Place\_Steps){#  
 Time = MDV\_data$sTime[t]  
 sDepthW = Place\_Depths[x]  
 #LDay = mLDay[t]  
 #P = P\_Conc[[x]][t]  
   
 # Calculate the time step requirement  
 Egg\_time\_step <-(Place\_step[x])^2/(Flowrate[x,t]\*Place\_step[x]+2\*Dispersion) # s  
 if(Time\_step>Egg\_time\_step){Too\_big\_t\_step[[Chla]][x]=1}else{Too\_big\_t\_step[[Chla]][x]=0}  
   
 # Advection and Diffusion coefficient (unitless):  
 One=Flowrate[x,t]\*Time\_step/Place\_step[x]  
 Two=Dispersion\*Time\_step/(Place\_step[x])^2  
   
 # Parameters:  
 uTm = MDV\_data$Temp[t]  
 cSigTm = cSigTm.vec[Chla]  
 cTmOpt = cTmOpt.vec[Chla]  
 cMuMax = cMuMax.vec[Chla]  
 cChD = cChD.vec[Chla]  
 cSigTm = cSigTm.vec[Chla]  
 cTmOpt = cTmOpt.vec[Chla]  
 cLOptRef = cLOptRef.vec[Chla]  
 kDResp = kDResp.vec[Chla]  
 kLoss = kLoss.vec[Chla]  
 kMort = kMort.vec[Chla]  
 hLRef = hLRef.vec[Chla]  
   
 # Temperature function  
 TmFunc\_value = TmFunc(uTm, cSigTm, cTmOpt)  
 # growth rate:  
   
 uMuMaxTm = cMuMax\*TmFunc\_value  
   
 ## light limitation  
 uLDay = MDV\_data$mL#LDay  
 ISurf = uLDay\*fPAR\*(1-fRefl)  
 aExtCoef = (cExtWat+cExtSpDet\*TotChla[[Chla]][t-1,x]/cChD\*gPermg)  
   
 aLLim = ifelse(Chla=="GreenCh",  
 LehmanFunc(aExtCoef=aExtCoef,sDepthW=sDepthW,ISurf=ISurf,uTm=uTm,cSigTm=cSigTm,cTmOpt=cTmOpt,hLRef=hLRef,TmFunc\_value=TmFunc\_value),  
 SteeleFunc(aExtCoef=aExtCoef,sDepthW=sDepthW,ISurf=ISurf,uTm=uTm,cSigTm=cSigTm,cTmOpt=cTmOpt,cLOptRef=cLOptRef,TmFunc\_value=TmFunc\_value))   
   
 #aPLim = PLimFunc(P)  
 aMuTmL = uMuMaxTm\*aLLim #min(aLLim,aPLim(P),na.rm = T) # Growth rate at current light and temp  
   
 growth = aMuTmL #\*aNutLim  
 #starting\_growth\*(1-(Flowrate[x,t]/(Monod\_Speed+Flowrate[x,t])))\*(1-TotChla[x,(t-1)]/(TotChla[x,(t-1)]+Monod\_Conc))  
   
 # Respiration  
 ukDRespTm = kDResp\*TmFunc\_value  
   
 # Grazing Loss  
 ukLossTm = kLoss\*TmFunc\_value  
 degradation = ukDRespTm + kMort + ukLossTm  
   
 # State variable output  
 #sGrowth[t] = growth  
 #sDegradation[t] = degradation  
 #sLDay[t] = uLDay   
   
 #sLLim[t] = aLLim  
 #sPLim[t] = aPLim  
 #sMuMaxTm[t] = uMuMaxTm  
   
 if(x<Num\_Place\_Steps){#  
 TotChla[[Chla]][t,x]=TotChla[[Chla]][(t-1),(x-1)]\*(One+Two)+TotChla[[Chla]][(t-1),x]\*((1+growth-degradation)^(Time\_step/SecondPerDay)-One-2\*Two)+TotChla[[Chla]][(t-1),(x+1)]\*Two  
 }else{  
 TotChla[[Chla]][t,x]=TotChla[[Chla]][(t-1),(x-1)]\*(One+Two)+TotChla[[Chla]][(t-1),x]\*((1+growth-degradation)^(Time\_step/SecondPerDay)-One-2\*Two)+TotChla[[Chla]][(t-1),(x-1)]\*Two # mirrored edge  
 }  
 }  
 }  
}  
  
Too\_big\_t\_step

## $BlueCh  
## [1] 0 0 0 0 0 0  
##   
## $GreenCh  
## [1] 0 0 0 0 0 0  
##   
## $DiatCh  
## [1] 0 0 0 0 0 0

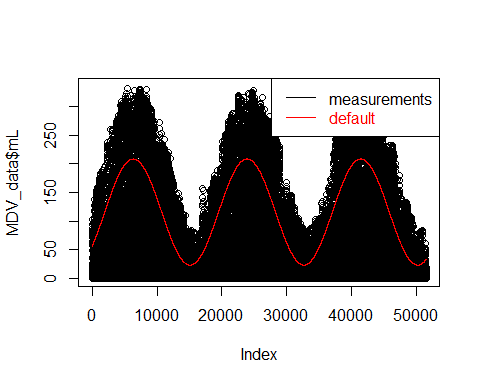
# Visualization:

# Data visualization:

# PO4 concentration (mg/L)####  
plot(MDV\_data$Date, MDV\_data$PO4\_MarkCanal, col=1, ylim = c(0,0.12), ylab = "PO4 (mg/L)", type="l")  
lines(MDV\_data$Date, MDV\_data$PO4\_DintelCanal, col=2)  
lines(MDV\_data$Date, MDV\_data$PO4\_MVCanal, col=3)  
lines(MDV\_data$Date, MDV\_data$PO4\_Vliet, col=4)  
legend("topright",legend = c("MarkCanal","DintelCanal","MV\_Canal","Vliet"),col=1:4,text.col = 1:4,lty=1)



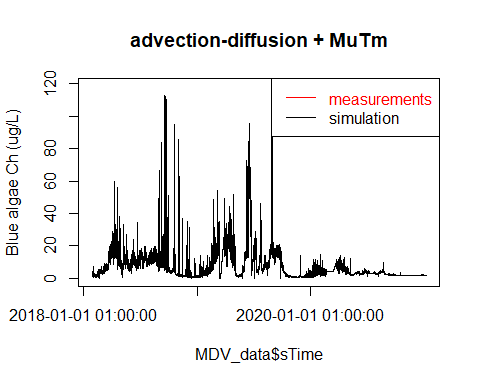
####  
plot(MDV\_data$mL)  
lines(cLDayAve-cLDayVar\*cos(2\*pi\*(MDV\_data$sTime+TenDays)/DaysPerYear/SecondPerDay),col="red")  
  
legend("topright",legend = c("measurements","default"),text.col=c("black","red"),col=c("black","red"),lty=1)



## Mark Canal:

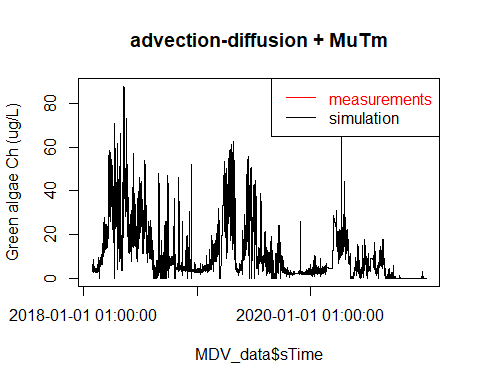
### Cyanobacteria:

plot(MDV\_data$sTime, MDV\_data$BlueCh\_MarkCanal, t="l", col="red",xaxt="n",ylab = "Blue algae Ch (ug/L)",main = "advection-diffusion + MuTm")  
lines(MDV\_data$sTime,TotChla[[1]][,1],xlab = "time.step (\*day)",t="l",ylim = c(0,100),xaxt="n")  
axis(1,at=as.numeric(ymd\_hms(c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))),labels = c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))  
  
legend("topright",legend = c("measurements","simulation"),col = c("red","black"),lty = 1,text.col = c("red","black"))



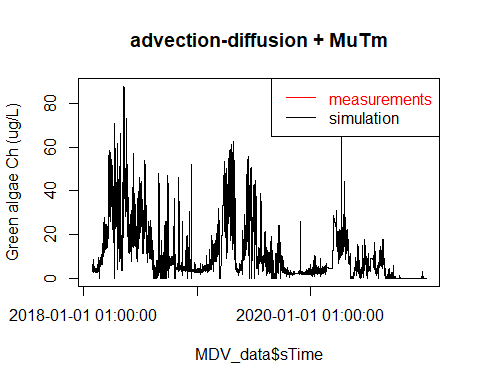
### Greenalgae:

plot(MDV\_data$sTime, MDV\_data$dChla\_MarkCanal, t="l", col="red",xaxt="n",ylab = "Green algae Ch (ug/L)",main = "advection-diffusion + MuTm")  
lines(MDV\_data$sTime,TotChla[[2]][,1],xlab = "time.step (\*day)",t="l",ylim = c(0,100),xaxt="n")  
axis(1,at=as.numeric(ymd\_hms(c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))),labels = c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))  
  
legend("topright",legend = c("measurements","simulation"),col = c("red","black"),lty = 1,text.col = c("red","black"))



### Diatom:

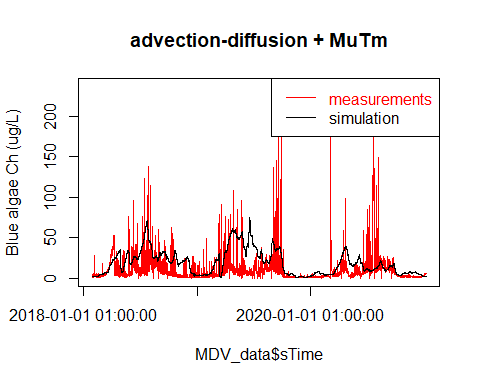
plot(MDV\_data$sTime, MDV\_data$dChla\_MarkCanal, t="l", col="red",xaxt="n",ylab = "Green algae Ch (ug/L)",main = "advection-diffusion + MuTm")  
lines(MDV\_data$sTime,TotChla[[3]][,1],xlab = "time.step (\*day)",t="l",ylim = c(0,100),xaxt="n")  
axis(1,at=as.numeric(ymd\_hms(c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))),labels = c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))  
  
legend("topright",legend = c("measurements","simulation"),col = c("red","black"),lty = 1,text.col = c("red","black"))



## Dintel canal:

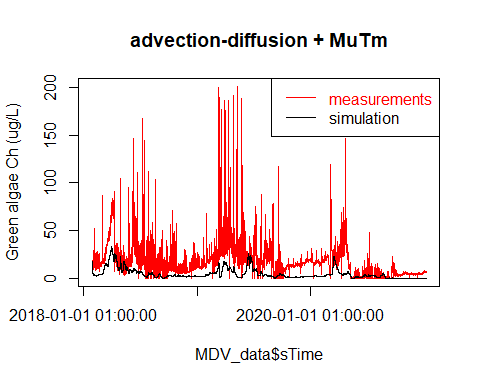
### Cyanobacteria:

plot(MDV\_data$sTime, MDV\_data$BlueCh\_DintelCanal, t="l", col="red",xaxt="n",ylab = "Blue algae Ch (ug/L)",main = "advection-diffusion + MuTm")  
lines(MDV\_data$sTime,TotChla[[1]][,2],xlab = "time.step (\*day)",t="l",ylim = c(0,100),xaxt="n")  
axis(1,at=as.numeric(ymd\_hms(c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))),labels = c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))  
  
legend("topright",legend = c("measurements","simulation"),col = c("red","black"),lty = 1,text.col = c("red","black"))



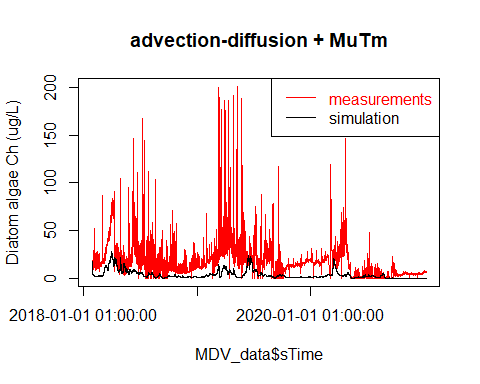
### Greenalgae:

plot(MDV\_data$sTime, MDV\_data$dChla\_DintelCanal, t="l", col="red",xaxt="n",ylab = "Green algae Ch (ug/L)",main = "advection-diffusion + MuTm")  
lines(MDV\_data$sTime,TotChla[[2]][,2],xlab = "time.step (\*day)",t="l",ylim = c(0,100),xaxt="n")  
axis(1,at=as.numeric(ymd\_hms(c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))),labels = c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))  
  
legend("topright",legend = c("measurements","simulation"),col = c("red","black"),lty = 1,text.col = c("red","black"))



### Diatom:

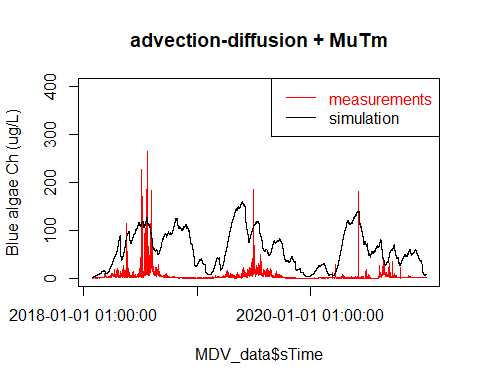
plot(MDV\_data$sTime, MDV\_data$dChla\_DintelCanal, t="l", col="red",xaxt="n",ylab = "Diatom algae Ch (ug/L)",main = "advection-diffusion + MuTm")  
lines(MDV\_data$sTime,TotChla[[3]][,2],xlab = "time.step (\*day)",t="l",ylim = c(0,100),xaxt="n")  
axis(1,at=as.numeric(ymd\_hms(c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))),labels = c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))  
  
legend("topright",legend = c("measurements","simulation"),col = c("red","black"),lty = 1,text.col = c("red","black"))



## Dintel canal:

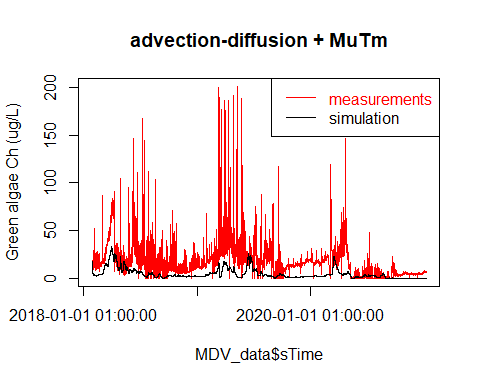
### Cyanobacteria:

plot(MDV\_data$sTime, MDV\_data$BlueCh\_Vliet, t="l", col="red",xaxt="n",ylab = "Blue algae Ch (ug/L)",main = "advection-diffusion + MuTm",ylim = c(0,400))  
lines(MDV\_data$sTime,TotChla[[1]][,6],xlab = "time.step (\*day)",t="l",ylim = c(0,100),xaxt="n")  
axis(1,at=as.numeric(ymd\_hms(c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))),labels = c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))  
  
legend("topright",legend = c("measurements","simulation"),col = c("red","black"),lty = 1,text.col = c("red","black"))



### Greenalgae:

plot(MDV\_data$sTime, MDV\_data$dChla\_DintelCanal, t="l", col="red",xaxt="n",ylab = "Green algae Ch (ug/L)",main = "advection-diffusion + MuTm")  
lines(MDV\_data$sTime,TotChla[[2]][,2],xlab = "time.step (\*day)",t="l",ylim = c(0,100),xaxt="n")  
axis(1,at=as.numeric(ymd\_hms(c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))),labels = c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))  
  
legend("topright",legend = c("measurements","simulation"),col = c("red","black"),lty = 1,text.col = c("red","black"))



### Diatom:

plot(MDV\_data$sTime, MDV\_data$dChla\_DintelCanal, t="l", col="red",xaxt="n",ylab = "Diatom algae Ch (ug/L)",main = "advection-diffusion + MuTm")  
lines(MDV\_data$sTime,TotChla[[3]][,2],xlab = "time.step (\*day)",t="l",ylim = c(0,100),xaxt="n")  
axis(1,at=as.numeric(ymd\_hms(c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))),labels = c("2018-01-01 01:00:00","2019-01-01 01:00:00","2020-01-01 01:00:00"))  
  
legend("topright",legend = c("measurements","simulation"),col = c("red","black"),lty = 1,text.col = c("red","black"))

