1D Model

Anne-Karin Cooke

## Experiments

# Model set up

In this model, salt enters the system through rainfall and from the groundwater table with the same concentration.

## Constants

Soil properties were derived from standard Australian soils in Neurotheta (Minasny and McBratney, 2002, as cited in Shah et al, 2011).

# Sandy Clay Loam  
 n<-0.367 # porosity  
 # more soil variables for evaporation & losses  
 # Hydraulic conductivity  
 K\_s<-52.08\*10 # mm/day  
 # campbell's b  
 b<-6.4069 # neurotheta sandy clay loam  
 # van Genuchten parameters  
# avg <- 0.0521  
# nvg <- 1.237  
 s\_fc<-0.2677/n # Field capacity  
 # This is the bubbling pressure  
 psi\_s\_bar<--1.2E-3 #  
   
 h1bar = -psi\_s\_bar   
 hb = psi\_s\_bar\*-10^5 # mm  
   
  
# parameters describing the soil  
 soilpar <- list(b = b, n = n, s\_fc = s\_fc, K\_s = K\_s,   
 psi\_s\_bar = psi\_s\_bar, h1bar = h1bar, hb = hb)  
 #................................................  
 # Vegetation 1 (Grass)  
 # paspalum secateum F-I and R-I, 2004  
  
 Zr = 400 # soil depth (mm) Check Also Table 2...Fernandez-Illescas and Rodriguez-Iturbe...2001  
  
  
 # parameters describing the root zone   
 vegpar <- list(Zr = Zr)  
  
  
# parameters describing plant dynamics and salt features  
  
 alpha\_i=1 #maximum infiltration rate per day (K\_s and therefore soil type dependency in balances function code)  
  
 k=12 # Saco et al, 2013  
 W0=0.2 # Saco et al, 2013  
 gmax=0.05 # Saco et al, 2013  
 k1=5 # Saco et al, 2013  
 c=10 # Saco et al, 2013  
 f= 1 # f is the soil salt leaching efficiency (whether some salt is retained)  
 ConcConst = 0.1 # ConcConst is the concentration of the salt in the infiltrating water in g/l  
 CM.gw = 0.1 # salt concentration in groundwater  
 d=0.24 # fraction of plant mortality  
   
 par <- list(alpha\_i=alpha\_i,k=k, W0=W0, gmax=gmax, k1=k1, c=c, f=f, ConcConst=ConcConst, CM.gw= CM.gw, d=d)

## Infiltration function

Infil <- function(h,P, par){  
   
 I=par$alpha\_i\*h\*(P+par$k\*par$W0)/(P+par$k)  
   
 return (I)  
}

## Water uptake function

WU <- function(M,P,par){   
# using Svir in here means scaling Svir back to M, easier to do at Svir in balances   
# WU=par$gmax\*((M\*(1+Svir))/(((M\*(1+Svir))+par$k1)))\*P   
 WU=par$gmax\*(M/((M+par$k1)))\*P   
   
 return(WU)  
}

## Vertical Flux function

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
# based on Salvucci, 1993  
L\_n <- function(M,Z,soilpar,vegpar) {  
 Zr <- vegpar$Zr  
 hb <- - soilpar$psi\_s\_bar\*10^5 # (mm?)  
 soilpar$s\_fc <- (Z/hb)^(-1/soilpar$b)  
   
 s=M/(soilpar$n\*vegpar$Zr)  
   
 psi = hb\*s^-soilpar$b  
   
 m=2 + 3/soilpar$b#5.64 # in Salvucci's paper it is called n, but I called it m here to not confuse it with porosity  
  
  
q <-((Z/hb)^(-m)-(psi/hb)^(-m))/(1+(psi/hb)^(-m)+(m-1)\*(Z/hb)^(-m))#/(soilpar$n\*Zr) #   
  
# Mass flux  
  
flux <- soilpar$K\_s\*q  
  
return(flux)  
}

## Plant growth function

Gr <- function(M,P,par) {   
   
 Gr = par$c\*WU(M,P,par)  
   
 return(Gr)  
}

## Plant mortality function

Mo <- function(P,M,Svir,par) {  
 # needs to be M/Svir because both are "large" numbers  
 # you want a number ~1 for multiplication, or <0.1 for addition  
 Mo = P\*(par$d\*(M/Svir))  
   
 return(Mo)  
  
}

## balances function

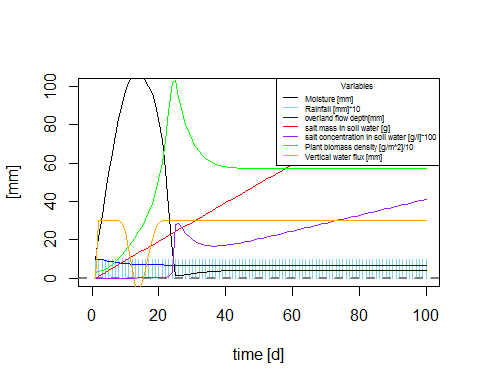
balances <- function(Rain, par, plotit=T,  
 soilpar,  
 vegpar){  
   
# Storage vectors for the daily steps are initialized.  
  
 M <- rep(0,length(Rain)) # soil moisture [mm]  
 h <- rep(0,length(Rain)) # infiltration depth [mm]  
 P <- rep(0,length(Rain)) #biomass density []  
 CM<- rep(0,length(Rain)) # Salt concentration in soil water in g/L or g/mm  
 SmI<- rep(0,length(Rain)) # Salt mass in infiltrating water [g]  
 SmM <- rep(0,length(Rain)) # Salt mass in soil water [g]  
 In <- rep(0,length(Rain)) # infiltration [mm]  
  
 Svir <- rep(0,length(Rain)) # virtual saturation  
 flux<- rep(0,length(Rain)) # drainage and capillary rise flux, according to sign  
  
# Initial values to start the simulation.  
  
   
 M[1] <- 10  
 h[1] <- 10   
 P[1] <- 30  
 CM[1]<- 0  
 Svir[1] <- M[1]  
  
# We decided to split the numerical calculations for the daily into 12 substeps.  
  
 deltat <- 12 # split in 12 increments  
  
   
# Storage vectors for the substeps are initialized.  
  
 M\_sub <- rep(0,deltat)  
 h\_sub <- rep(0,deltat)  
 I\_sub <- rep(0,deltat)  
 WU\_sub <-rep(0,deltat) # Water uptake in mm  
 P\_sub <- rep(0,deltat)   
 Gr\_sub <- rep(0,deltat) # Growth of biomass  
 Mo\_sub<- rep(0,deltat) # Mortality of biomass  
 SmI\_sub <- rep(0,deltat)   
 SmM\_sub<- rep(0,deltat)   
 CM\_sub<- rep(0,deltat)   
 Svir\_sub <- rep(0,deltat) # virtual saturation  
  
 flux\_sub<-rep(0,deltat) # calculates leakage loss without evaporation loss  
  
 U\_salt <-rep(0,deltat) # Salt mass rising  
 L\_salt <-rep(0,deltat) # salt mass drained  
  
  
   
 timeincr= 1/deltat  
   
 for (t in 2:length(Rain)){  
   
 for (tt in 1:(deltat-1)) {  
   
 h.old <- ifelse(tt==1,h[t-1],h\_sub[tt])  
 P.old <- ifelse(tt==1,P[t-1],P\_sub[tt])  
 M.old <- ifelse(tt==1,M[t-1],M\_sub[tt])  
 SmI.old <-ifelse(tt==1,SmI[t-1],SmI\_sub[tt])  
 CM.old <-ifelse(tt==1,CM[t-1],CM\_sub[tt])  
 Svir.old <-ifelse(tt==1,Svir[t-1],Svir\_sub[tt])  
  
  
# Balance for water depth on soil  
  
 h\_sub[tt+1] <- h.old + ifelse(tt==1,(10\*Rain[t]),0)   
 #- Infil(h.old, P.old,par)\*timeincr  
  
 # Infiltration  
 par$alpha\_i <- ifelse(h\_sub[tt+1]<soilpar$K\_s\*timeincr, 1,(1-(h\_sub[tt+1]-soilpar$K\_s\*timeincr)/h\_sub[tt+1]))  
 # Calculate infiltration and recalculate h\_sub   
 I\_sub[tt] <- Infil(h.old, P.old,par)\*timeincr  
 h\_sub[tt+1] <- h\_sub[tt+1] - I\_sub[tt]   
  
   
   
# Now do all plant uptake and growth  
# water uptake by plants: include infiltration in available water  
  
 WU\_sub[tt] <- WU(M=Svir.old,P.old,par)\*timeincr   
   
 # growth rate  
 Gr\_sub[tt] <- Gr(M=Svir.old, P.old,par)\*timeincr   
 # Mortality  
 Mo\_sub[tt]<- Mo(P.old,M=M.old, Svir=Svir.old, par)\*timeincr  
 # calculate plant biomass balance  
 P\_sub[tt + 1] <- P.old + Gr\_sub[tt]- Mo\_sub[tt]   
   
   
# re-calculate water balance  
# 2. before leaching  
 M\_sub[tt + 1] <- M.old + I\_sub[tt] - WU\_sub[tt] #- L\_sub[tt]   
  
# Calculate salt concentration in the soil  
  
 # 3. calculate leaching and capillary rise amount  
 flux\_sub[tt+1]<-do.call(L\_n,list(M=M\_sub[tt+1],Z=Z,soilpar=soilpar,vegpar=vegpar))  
  
# 4. final adjust soil moisture for leaching or capillary rise  
   
 M\_sub[tt + 1] <- M\_sub[tt + 1] + flux\_sub[tt+1]\*timeincr  
  
  
# calculate saltbalance  
  
# Salt leaching  
 L\_salt[tt+1] <- ifelse(flux\_sub[tt+1]<0, par$f\*CM\_sub[tt+1]\*flux\_sub[tt+1]\*timeincr,0) # leaching of salt  
  
# salt upflow  
 U\_salt[tt+1] <- ifelse(flux\_sub[tt+1]>0, par$CM.gw\*flux\_sub[tt+1]\*timeincr,0) # rise of salt  
  
# salt mass coming in with infiltration  
 SmI\_sub[tt+1]<- SmI.old + I\_sub[tt]\*par$ConcConst   
  
#salt mass in soil  
 SmM\_sub[tt+1] <- SmI\_sub[tt+1] + U\_salt[tt+1] - L\_salt[tt+1]  
# salt concentration in soil  
 CM\_sub[tt+1]<- (SmM\_sub[tt+1]/M\_sub[tt+1])\*(1/58.44) #   
   
# Virtual saturation (Shah et al., 2012), here in [mm] to be in the same unit as M  
 Svir\_sub[tt + 1]<-soilpar$n\*vegpar$Zr\*((soilpar$h1bar\*10^-1)^(1/soilpar$b))\*  
 ((soilpar$h1bar\*10^-1)\*(M\_sub[tt + 1]/  
 (soilpar$n\*vegpar$Zr))^(-soilpar$b)  
 +(3.6\*CM\_sub[tt + 1]))^(-1/soilpar$b)  
  
 }   
   
# Aggregating the substep results to daily values.  
  
 P[t] = P\_sub[deltat]  
 M[t] = M\_sub[deltat]  
 h[t] = h\_sub[deltat]  
 CM[t] = CM\_sub[deltat]  
 SmM[t] = SmM\_sub[deltat]   
 SmI[t]=SmI\_sub[deltat]  
 In[t]= sum(I\_sub)  
 flux[t] = sum(flux\_sub)  
 Svir[t] = Svir\_sub[deltat]  
  
}  
  
  
# Plotting  
   
if (plotit==T) {   
 plot(Rain\*10, type="h",col="skyblue",ylim=c(0,100),xlim=c(0,time),xlab=("time [d]"), ylab=("[mm]"))  
 lines(M, type="l",col="black" )  
   
 lines(h,type="l", col="blue")  
 abline(h=0, col="Gray50",lwd=2,lty=2)  
  
 lines(SmM,type="l", col="red")  
 lines(CM\*100,type="l", col="purple")  
 lines(P/10,type="l", col="green")  
 lines(flux,type="l", col="orange")  
   
   
 legend("topright", title="Variables",cex=0.5, pt.cex=0.1, c("Moisture [mm]","Rainfall [mm]\*10","overland flow depth[mm] ","salt mass in soil water [g]", "salt concentration in soil water [g/l]\*100", "Plant biomass density [g/m^2]/10", "Vertical water flux [mm]"),  
 col=c("black","skyblue","blue","red","purple","green", "orange"),lty=1)  
   
}  
  
Out <- data.frame(P=P,M=M,h=h, CM=CM, SmM=SmM, In=In, flux=flux, Svir=Svir)  
#return(Out)  
}

## Uniform rain of 1 mm per day

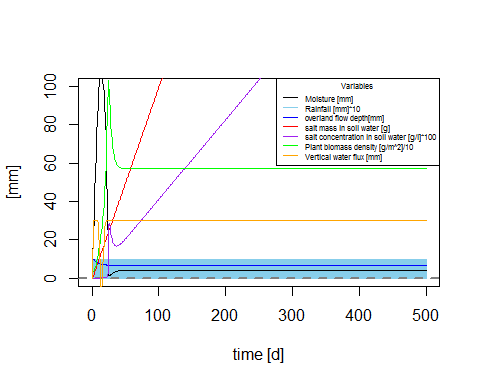
Rain\_function<-function(time){ Rain <- rep(1, time)  
 return(Rain)}

This calls the balances function ## Results

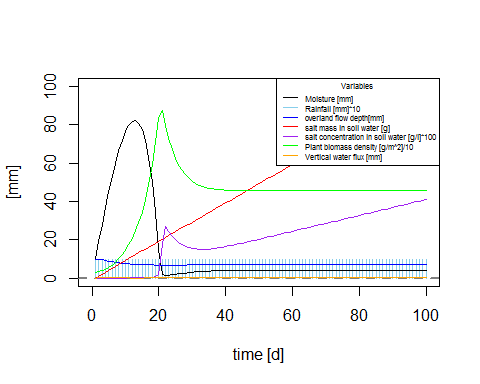
Z =1000 # [mm] groundwater depth   
 time = 100 #days  
 balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)



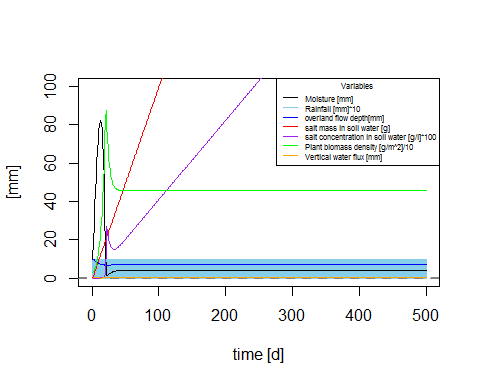
time =500  
 balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)  
# Z =2000 # [mm] groundwater depth   
# time = 100 #days  
# balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)  
# time =500 #days  
# balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)  
# Z =3000 # [mm] groundwater depth  
# time = 100 #days  
# balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)  
# time =500 #days  
 balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)



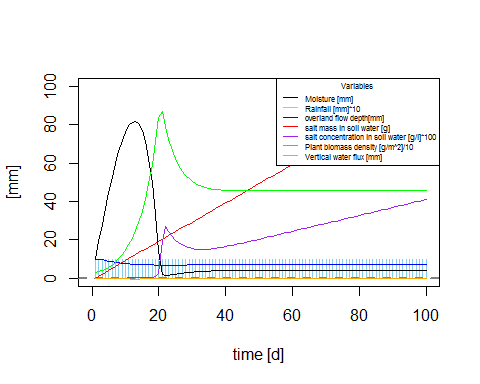
Z =5000 # [mm] groundwater depth  
 time = 100 #days  
 balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)



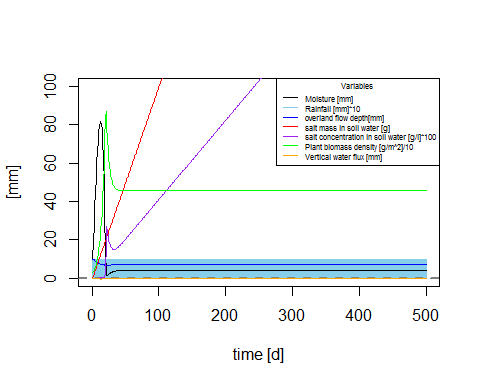
time =500 #days  
 balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)



Z =10000 # [mm] groundwater depth  
 time = 100 #days  
 balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)



time =500 #days  
 balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)



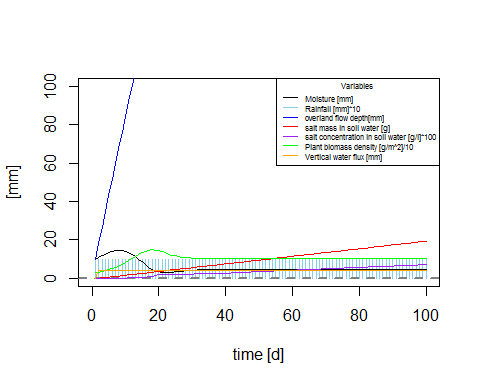
## Discussion - changing Z (distance to groundwater depth)

After some flucations at the beginning which can be considered the initalization of the model, soil moisture, plant biomass and overland flow depths stabilize on the longterm. Whereas salt mass and concentrations continously rise as they are brought in by infiltration with every rainfall and accumulate in the soil. Plant biomass will eventually die, as soon as salt concentration make the soil inhabitable. The shallower the water table, the more water enters the soil moisture storage through capillary rise ("flux" in the graphs is much higher) and can be used by plants. Therefore plant biomass stabilizes at a higher level, the closer the groundwater table.

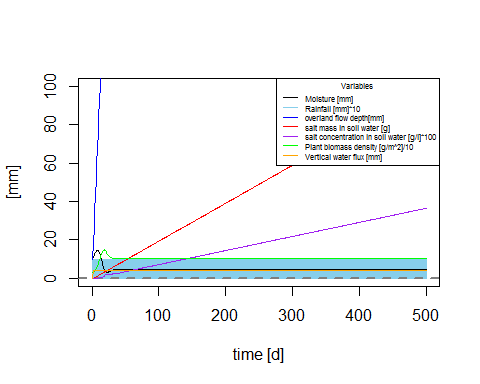
## Testing different soil types

### Medium Heavy Clay

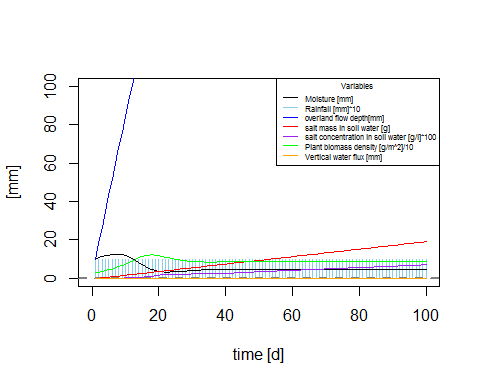
# Medium Heavy Clay  
 n<-0.4473 # porosity  
 # more soil variables for evaporation & losses  
 # Hydraulic conductivity  
 K\_s<-2.82\*10 # mm/day  
 # Campbell's b  
 b<-16.1501 # neurotheta Medium heavy clay  
  
 s\_fc<-0.3936/n # Field capacity  
 # bubbling pressure  
 psi\_s\_bar<--1.4E-3   
 h1bar = -psi\_s\_bar   
 hb = psi\_s\_bar\*-10^5 #mm  
  
  
# parameters describing the soil  
 soilpar <- list(b = b, n = n, s\_fc = s\_fc, K\_s = K\_s,   
 psi\_s\_bar = psi\_s\_bar, h1bar = h1bar, hb = hb)  
  
 Z =1000 # [mm] groundwater depth   
 time = 100 #days  
 balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)



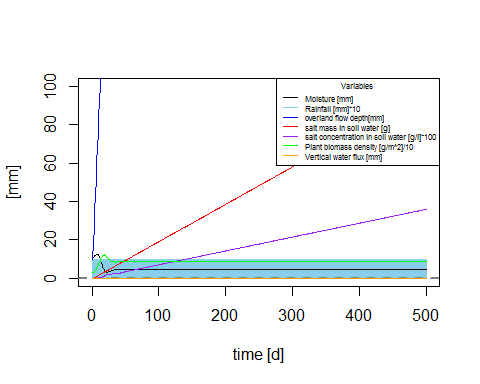
time =500  
 balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)



Z =10000 # [mm] groundwater depth   
 time = 100 #days  
 balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)



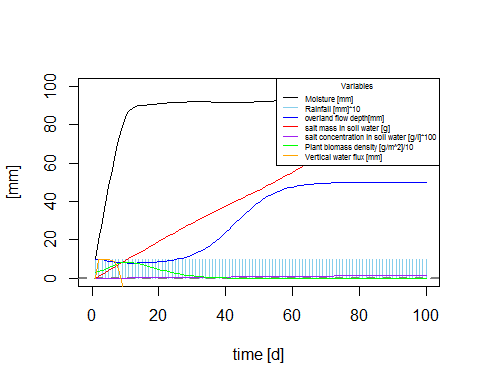
time =500  
 balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)



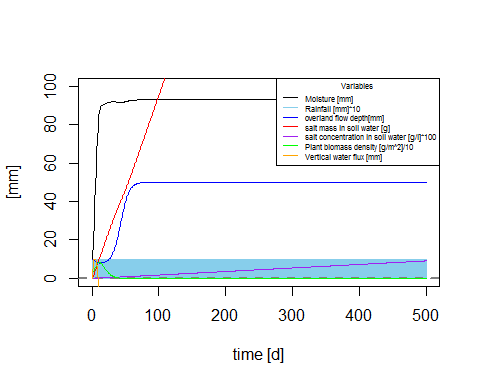
## Testing different soils

### Coarse Sand

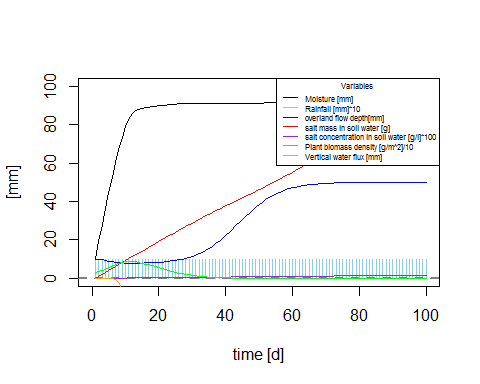
# Coarse Sand  
 n<-0.368 # porosity  
 # more soil variables for evaporation & losses  
 # Hydraulic conductivity  
 K\_s<-182.68\*10 # mm/day  
 # Campbell's b  
 b<- 4.1152  
  
 s\_fc<-0.1895/n # Field capacity  
   
 psi\_s\_bar<--0.61E-3   
 h1bar = -psi\_s\_bar   
 hb = psi\_s\_bar\*-10^5 #mm  
  
# parameters describing the soil  
 soilpar <- list(b = b, n = n, s\_fc = s\_fc, K\_s = K\_s,   
 psi\_s\_bar = psi\_s\_bar, h1bar = h1bar, hb = hb)  
  
 Z =1000 # [mm] groundwater depth   
 time = 100 #days  
 balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)



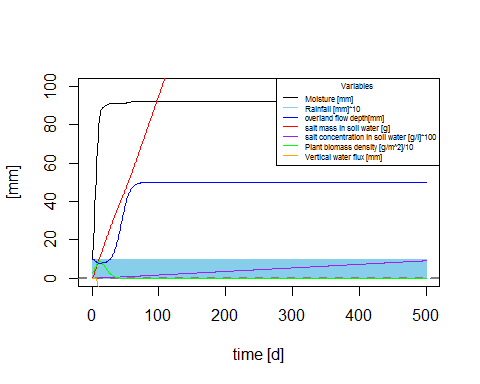
time =500  
 balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)



Z =10000 # [mm] groundwater depth   
 time = 100 #days  
 balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)



time =500  
 balances(Rain=Rain\_function(time), par=par, plotit=T, soilpar, vegpar)



## Discussion - Testing different soils

For deep water tables (10 m) the three soil types don't differ. For shallow water tables (1 m), the highest longterm capillary rise (30 mm) and highest plant biomass (150 g/m^2) is found for sandy clay loam, soil moisture stabilizes at about 10 mm. The smallest capillary rise and plant biomass is found for heavy medium clay. Coarse sand and clay show the same amount of soil moisture, whereas coarse sand is able to maintain higher plant biomass as a result of higher capillary rise than clay. Rise in salt is not effected by soil type nor by distance to the groundwater table.