Soilmoisture

# 1. Setup

As implemented in every submodel, we first have to setup the basics - time variable, site data, parameters and some prior calculations of example values.

# Setup soil moisture   
  
### Load packages, defines time units, and set up site data and parameters ----  
  
## Required packages  
library(dplyr)

##   
## Attaching package: 'dplyr'

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

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

library(FME)

## Loading required package: deSolve

## Loading required package: rootSolve

## Loading required package: coda

# Setting some time unit variables in unit seconds ----  
t\_units <- list(hour = 3600,  
 halfh = 1800,  
 day = 86400,  
 month = 2592000,  
 year = 31536000)  
dt <- t\_units$hour # delta time: model time step  
  
## Load and prepare input ----  
# required variables from measured data  
# rh: relative humidity  
# prec: precipitation  
# temp.air: air temperature  
source("setup\_sitedata.R")  
  
# example values for calibration  
input$lw\_out <- 0.95 \* 5.670373e-8 \* input$tair^4 # no data for lw\_out, so making a rough approximation using tair  
input$Rn <- input$sw\_in + input$lw\_in - fluxes$sw\_out - input$lw\_out  
  
## Set initial values ----  
initial\_state <- list(theta=fluxes$swc[1]) # initial value for volumetric water content of soil [m3 m-3]  
  
## Load parameters and adjust units ----  
parsfile <- "pars\_soil.csv"  
source("setup\_parameters.R")

# 2. Main function

The main function for modeling soil moisture contains soil moisture-specific calculations and a for-loop in which the soil water content is calculated in dependence of precipitation, evaporation and drainage.

# Function calculating soil moisture changes  
  
get\_theta\_soil <- function(input, pars, initial\_state) {  
  
 list2env(pars, envir = environment())  
   
 # Getting variables used  
 theta.in <- initial\_state$theta  
 prec <- input$p  
 Rh <- input$rh  
 temp <- input$tair - 273.15  
 time <- input$time  
 gs <- fluxes$g  
 rn <- input$Rn  
   
 # calculations  
 ps <- 1 - (BD/PD) # soil pore space [unitless]: bulk density = 1200 kg m-3 (climate data), particle density = 2650 kg m-3 (literature)  
 V <- 1 \* 1 \* SD # volume of soil layer [m3]  
   
 # for evaporation  
 gamma <- (cp \* 101.325) / (lambda \* MWrat) # psycrometric constant for potential evaporation [kPa K-1]  
 # for drainage  
 # b.min <- 2.91 + 0.159 \* (clay \* 100)  
 # f <- (SOC / (BD \* SD)) \* 1.72 # soil organic matter fraction; f = (%C \* 1.72)/100  
 # b.sm <- (1 - f) \* b.min + f \* b.om  
   
 sps <- theta.sat \* SD  
   
 # equations  
  
 # evaporation from soil  
 # Penman-Monteith potential evapotranspiration  
  
 # Ep = (Δ(Rn – Gs) + ρ cp(es – ea)/ra) / (λ(Δ+γ))  
  
 # where:  
  
 # Ep: potential evapotranspiration (kg m−2 s−1)  
 # Δ: slope of the es to T curve = 4098 \* (0.6108 \* exp( 17.27 \* T / (T + 237.3))) / (T + 237.3)^2 (kPa ºC-1) (T = Tsoil?; T in Celsius!)  
 # rn: net radiation (J m-2 s-1) (from radiation model)  
 # gs: ground heat flux (J m-2 s-1) (from soil temperature model)  
 # cp: specific heat of air (J kg-1)  
 # ρ: the air density (kg m−3)  
 # es: air saturation vapor pressure (kPa)  
 # ea: air actual vapour pressure (kPa)  
 # ra: aerodynamic resistance (s m-1) --> assumed to be 10  
 # γ: psycrometric constant (kPa ◦C−1) --> has to be calculated, but then we assume it to be constant  
 # γ = (cp \* air\_pressure) / (λ \* MWrat)  
 # λ: latent heat of vaporization (J kg−1)  
  
  
 # Actual soil evaporation (adapted from Aydin et. al 2005)  
  
 # Ea = Ep \* ( (log[WP] – log[WPa]) / (log[WPfc] – log[WPa]) ) / (Vsoil/BD)  
  
 # WP = soil water potential  
 # WPa = water potential of air (-100MPa (average value))  
 # WPfc = soil water potential at field capacity  
  
  
  
 # drainage  
  
 # drain.t = -(k / SD )\*(psi - psi.n1) - k (equation 8.27 from Bonan p. 125 )  
 # k: hydraulic conductivity [m s-1]  
 # SD: soil layer thickness [m]  
 # psi: matric potential of soil [m]  
 # psi.n1: matric potential of soil beneath soil layer [m]  
 # psi.n1 <- psi.sat \* (s^-B) (equation from CLM4.5 p. 172)  
 # s <- 0.5 ((theta.sat + theta)/theta.sat) [unitless]  
 # B <- (1 - f) \* B.min + f \* B.om [unitless]  
 # B.min <- 2.91 + 0.159 \* clay [unitless]  
 # B.om <- 2.7 [unitless]  
 # f <- (SOC(kg m-2) / (BD \* SD)) \* 1.72 # soil organic matter fraction; f = %C \* 1.72  
 # SOC(kg/m-2) = SOC (%)× BD (kg/m3)× SD (m) x 1000  
 # where, SOC - Concentration of soil organic carbon (%); BD - Bulk density (kg/m3); SD- soil sampling depth (m)  
  
  
 # output variables  
  
 theta <- rep(NA, length(time)) # soil moisture [m3 m-3]  
 drain <- rep(NA, length(time)) # drainage [m s-1]  
 runoff <- rep(NA, length(time)) # runoff [m s-1]  
 k <- rep(NA, length(time)) # hydraulic conductivity [m s-1]  
 evap <- rep(NA, length(time)) # evaporation [m s-1]  
 psi <- rep(NA, length(time)) # matric potential [m]  
 s <- rep(NA, length(time)) # coefficient for drainage  
 # psi.n1 <- rep(NA, length(time)) # matric potential of soil beneath soil layer [m]  
  
  
 # Iterative calculations over time  
  
 for(t in time) {  
   
 # if(t==1) browser()  
 # print(t)  
  
 # first water content is taken from climate data, then theta from previous time step is taken for calculation  
 # theta.in is the measured soil water content (averaged for the whole soil) at the start of the time series  
 if(t == 1) {theta.t <- theta.in} else {theta.t <- theta[t-1]}  
  
 # precipitation is taken from climate data  
 prec.t <- prec[t] # [m3 dt-1]  
   
 rn.t <- rn[t]  
 gs.t <- gs[t]  
 temp.t <- temp[t]  
  
 # transpiration data is taken from Leaf Temperature Model  
 # trans.t <- trans[t]  
  
 # evaporation  
 # Calculating potential evaporation  
 delta <- 4098 \* (0.6108 \* exp( 17.27 \* temp.t / (temp.t + 237.3))) / (temp.t + 237.3)^2 # (kPa K-1) temp must be in Celsius  
 es <- 0.6108 \* exp(17.27\* temp.t / (temp.t + 237.3)) # (kPa)  
 ea <- es \* Rh[t] # (kPa)  
 Ep <- (delta \* (rn.t - gs.t) + d\_air \* cp \* (es - ea) / ra) / (lambda \* (delta + gamma)) # (kg m−2 s−1)  
 Ep <- Ep / 1000 # transform units to m3 m-2 s-1  
  
 # Calculating soil water potential  
 psi.t <- psi.sat \* (theta.t / theta.sat)^-b.sm # (m)  
 if(psi.t < psi.a) {psi.t <- psi.a}   
  
 # Calculating actual evaporation  
 evap.t <- epmod \* (Ep \* ( (log(-psi.t) - log(-psi.a)) / (log(-psi.ep) - log(-psi.a)) )) # (m s-1)  
  
 # hydraulic conductivity  
 k.t <- k.sat \* ((theta.t/theta.sat)^(2\*b.sm+3)) # (m s-1)  
  
 # drainage  
   
 # Calculating psi for soil beneath soil layer  
 # s.t <- 0.5 \* ((theta.sat + theta.t) / theta.sat)  
 # if(s.t < 0.01) {s.t <- 0.01}; if(s.t > 1) {s.t <- 1}  
 # psi.n1.t <- psi.sat \* (s.t^-B) # matric potential for layer N+1 (layer beneath layer N) -> equation taken from CLM4.5  
  
 # Calculating drainage  
 # drain.t <- - (k.t / SD) \* (psi.t - psi.n1.t) - k.t  
 drain.t <- (k.t / SD / 2) \* (psi.t - psi.n1) - k.t # new approach using psi.n1 as parameter (F.)  
 # if(drain.t > x) drain.t <- x  
   
 theta.l <- theta.t \* SD  
   
 # theta (water content) is current water content plus infiltration minus drainage  
 theta.l <- theta.l - (evap.t + drain.t) \* dt # multiplication with dt to get infiltration/drainage volume for model time step  
 if(theta.l > sps) {theta.l <- sps}; if(theta.l < 0.03) {theta.l <- 0.03} # making sure there are no impossible results  
   
 # Precipitation and runoff as excess water  
 theta.l <- theta.l + prec.t # only precipitation can become running  
 if(theta.l > sps) {  
 runoff.t <- theta.l - sps # (m3 dt-1)  
 theta.l <- sps  
 } else {runoff.t <- 0}  
   
  
   
 theta.t <- theta.l / SD  
  
 theta[t] <- theta.t  
 runoff[t] <- runoff.t  
 k[t] <- k.t  
 evap[t] <- evap.t  
 drain[t] <- drain.t  
 psi[t] <- psi.t  
 }  
  
 out <- data.frame(theta, runoff, k, evap, drain, psi, psi.n1)  
 out$swc <- out$theta  
 return(out)  
  
}

# Run model with uncalibrated parameters  
  
parsfile <- "pars\_soil.csv"  
source("setup\_soilmoisture.R")  
source("fun\_soilmoisture.R")  
  
output <- get\_theta\_soil(input = input, pars = pars, initial\_state = initial\_state)

# 3. Sensitivity analysis

### 3.1 Cost function

# cost function for soil moisture  
  
cost\_soilmoisture <- function(pars\_calib, params = pars) {  
   
 print(pars\_calib)  
 # Calibrated pars replace default values  
 for(i in names(pars\_calib)) {params[[i]] <- pars\_calib[[i]]}  
   
 # Call the model function  
 output <- get\_theta\_soil(input = input, pars = params, initial\_state = initial\_state)  
 # browser()  
 # Calculate residuals  
 resid <- output$swc - fluxes$swc  
 resid <- resid[!is.na(resid)]  
   
 return(resid)  
}

### 3.2 Running sensitivity analysis

## sensitivity analysis for soil moisture ##  
  
pars\_calib <- c(b.sm = 11.4, epmod = 1, psi.n1 = -3.4, ra = 10, theta.sat = 0.48)  
  
Sens <- sensFun(cost\_soilmoisture, pars\_calib)

## b.sm epmod psi.n1 ra theta.sat   
## 11.40 1.00 -3.40 10.00 0.48   
## b.sm epmod psi.n1 ra theta.sat   
## 11.40 1.00 -3.40 10.00 0.48   
## b.sm epmod psi.n1 ra theta.sat   
## 11.40 1.00 -3.40 10.00 0.48   
## b.sm epmod psi.n1 ra theta.sat   
## 11.40 1.00 -3.40 10.00 0.48   
## b.sm epmod psi.n1 ra theta.sat   
## 11.40 1.00 -3.40 10.00 0.48   
## b.sm epmod psi.n1 ra theta.sat   
## 11.40 1.00 -3.40 10.00 0.48

summary(Sens)

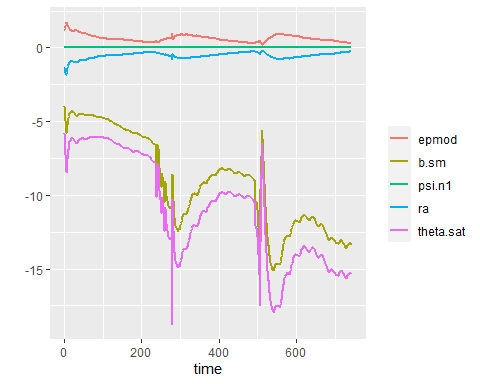
## value scale L1 L2 Mean Min Max N  
## b.sm 11.40 11.40 9.1e+00 9.6e+00 -9.1e+00 -15.68 -4.0e+00 743  
## epmod 1.00 1.00 6.4e-01 6.8e-01 6.4e-01 0.19 1.7e+00 743  
## psi.n1 -3.40 -3.40 1.8e-07 7.6e-07 1.8e-07 0.00 1.2e-05 743  
## ra 10.00 10.00 5.4e-01 5.8e-01 -5.4e-01 -1.90 -1.6e-01 743  
## theta.sat 0.48 0.48 1.1e+01 1.1e+01 -1.1e+01 -18.75 -5.8e+00 743

## Plotting ##  
  
library(ggplot2)  
library(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

library(tidyr)  
  
  
# Sensitivity analysis  
  
# plot sensitivity functions of model output to parameters  
  
date <- fluxes[2][1:743, ]  
par\_sens <- cbind(date, Sens[, 3:7])  
  
gg\_df <- gather(par\_sens, key = "type", value = "theta", epmod, b.sm, psi.n1, ra, theta.sat, factor\_key = T)  
  
ggplot() +  
 geom\_line(aes(gg\_df$date, gg\_df$theta, color = gg\_df$type), size = 1) +  
 labs(y = "", x = "time", color = "")



# 4. Calibration

Calibration was done for

- saturated soil water content (theta.sat) [m3 m-3]  
 - exponent for soil hydraulic equations (b.sm) [unitless]

rm(list = ls())  
  
parsfile <- "pars\_soil.csv"  
  
pars\_calib <- c(b.sm = 11.4, theta.sat = 0.48)  
pars\_low <- c(b.sm = 5, theta.sat = 0.3)  
pars\_high <- c(b.sm = 13, theta.sat = 0.6)  
  
source("setup\_soilmoisture.R")  
source("fun\_soilmoisture.R")  
source("fun\_costmoisture.R")  
  
myfit <- modFit(f = cost\_soilmoisture, p = pars\_calib, lower = pars\_low, upper = pars\_high)

## b.sm theta.sat   
## 11.40 0.48   
## b.sm theta.sat   
## 11.40 0.48   
## b.sm theta.sat   
## 11.40 0.48   
## b.sm theta.sat   
## 11.40 0.48   
## b.sm theta.sat   
## 5.2464458 0.5812737   
## b.sm theta.sat   
## 11.0963809 0.5226837   
## b.sm theta.sat   
## 11.0963809 0.5226837   
## b.sm theta.sat   
## 11.0963809 0.5226837   
## b.sm theta.sat   
## 9.3951653 0.5581726   
## b.sm theta.sat   
## 11.3217878 0.5261235   
## b.sm theta.sat   
## 11.3217878 0.5261235   
## b.sm theta.sat   
## 11.3217878 0.5261235   
## b.sm theta.sat   
## 11.0706687 0.5360297   
## b.sm theta.sat   
## 11.2293077 0.5315557   
## b.sm theta.sat   
## 11.2293077 0.5315557   
## b.sm theta.sat   
## 11.2293077 0.5315557   
## b.sm theta.sat   
## 10.9733276 0.5403582   
## b.sm theta.sat   
## 11.2044781 0.5330697   
## b.sm theta.sat   
## 11.2044781 0.5330697   
## b.sm theta.sat   
## 11.2044781 0.5330697   
## b.sm theta.sat   
## 11.1399991 0.5355971   
## b.sm theta.sat   
## 11.1399991 0.5355971   
## b.sm theta.sat   
## 11.1399991 0.5355971   
## b.sm theta.sat   
## 11.0232043 0.5401098   
## b.sm theta.sat   
## 11.0232043 0.5401098   
## b.sm theta.sat   
## 11.0232043 0.5401098   
## b.sm theta.sat   
## 10.92661 0.54439   
## b.sm theta.sat   
## 10.92661 0.54439   
## b.sm theta.sat   
## 10.92661 0.54439   
## b.sm theta.sat   
## 10.8428641 0.5481948   
## b.sm theta.sat   
## 10.8428641 0.5481948   
## b.sm theta.sat   
## 10.8428641 0.5481948   
## b.sm theta.sat   
## 10.7696553 0.5515805   
## b.sm theta.sat   
## 10.7696553 0.5515805   
## b.sm theta.sat   
## 10.7696553 0.5515805   
## b.sm theta.sat   
## 10.6314394 0.5572772   
## b.sm theta.sat   
## 10.7208880 0.5539831   
## b.sm theta.sat   
## 10.7208880 0.5539831   
## b.sm theta.sat   
## 10.7208880 0.5539831   
## b.sm theta.sat   
## 10.6227938 0.5581803   
## b.sm theta.sat   
## 10.6227939 0.5581803   
## b.sm theta.sat   
## 10.6227938 0.5581803   
## b.sm theta.sat   
## 10.5519833 0.5617472   
## b.sm theta.sat   
## 10.5519833 0.5617472   
## b.sm theta.sat   
## 10.5519833 0.5617472   
## b.sm theta.sat   
## 10.4916848 0.5647879   
## b.sm theta.sat   
## 10.4916848 0.5647879   
## b.sm theta.sat   
## 10.4916848 0.5647879   
## b.sm theta.sat   
## 10.3812722 0.5696672   
## b.sm theta.sat   
## 10.4504568 0.5669787   
## b.sm theta.sat   
## 10.4504568 0.5669787   
## b.sm theta.sat   
## 10.4504568 0.5669787   
## b.sm theta.sat   
## 10.3699231 0.5706573   
## b.sm theta.sat   
## 10.3699231 0.5706573   
## b.sm theta.sat   
## 10.3699231 0.5706573   
## b.sm theta.sat   
## 10.3157271 0.5736259   
## b.sm theta.sat   
## 10.3157271 0.5736259   
## b.sm theta.sat   
## 10.3157271 0.5736259   
## b.sm theta.sat   
## 10.2709214 0.5760625   
## b.sm theta.sat   
## 10.2709214 0.5760625   
## b.sm theta.sat   
## 10.2709214 0.5760625   
## b.sm theta.sat   
## 10.190602 0.579814   
## b.sm theta.sat   
## 10.2365652 0.5779693   
## b.sm theta.sat   
## 10.2365652 0.5779693   
## b.sm theta.sat   
## 10.2365652 0.5779693   
## b.sm theta.sat   
## 10.1730259 0.5810078   
## b.sm theta.sat   
## 10.1730259 0.5810078   
## b.sm theta.sat   
## 10.1730259 0.5810078   
## b.sm theta.sat   
## 10.1341922 0.5833188   
## b.sm theta.sat   
## 10.1341922 0.5833188   
## b.sm theta.sat   
## 10.1341922 0.5833188   
## b.sm theta.sat   
## 10.1030609 0.5851327   
## b.sm theta.sat   
## 10.1030609 0.5851327   
## b.sm theta.sat   
## 10.1030609 0.5851327   
## b.sm theta.sat   
## 10.0481876 0.5877939   
## b.sm theta.sat   
## 10.0481876 0.5877939   
## b.sm theta.sat   
## 10.0481876 0.5877939   
## b.sm theta.sat   
## 10.0397031 0.5887983   
## b.sm theta.sat   
## 10.0397031 0.5887983   
## b.sm theta.sat   
## 10.0397031 0.5887983   
## b.sm theta.sat   
## 10.0084029 0.5903827   
## b.sm theta.sat   
## 10.0084029 0.5903827   
## b.sm theta.sat   
## 10.0084029 0.5903827   
## b.sm theta.sat   
## 9.9891623 0.5915755   
## b.sm theta.sat   
## 9.9891623 0.5915755   
## b.sm theta.sat   
## 9.9891623 0.5915755   
## b.sm theta.sat   
## 9.955427 0.593251   
## b.sm theta.sat   
## 9.955427 0.593251   
## b.sm theta.sat   
## 9.955427 0.593251   
## b.sm theta.sat   
## 9.9512012 0.5938618   
## b.sm theta.sat   
## 9.9512012 0.5938618   
## b.sm theta.sat   
## 9.9512012 0.5938618   
## b.sm theta.sat   
## 9.9329830 0.5948028   
## b.sm theta.sat   
## 9.9329830 0.5948028   
## b.sm theta.sat   
## 9.9329830 0.5948028   
## b.sm theta.sat   
## 9.9222191 0.5954938   
## b.sm theta.sat   
## 9.9222191 0.5954938   
## b.sm theta.sat   
## 9.9222191 0.5954938   
## b.sm theta.sat   
## 9.9034412 0.5964406   
## b.sm theta.sat   
## 9.9034412 0.5964406   
## b.sm theta.sat   
## 9.9034412 0.5964406   
## b.sm theta.sat   
## 9.8950051 0.5970587   
## b.sm theta.sat   
## 9.8950051 0.5970587   
## b.sm theta.sat   
## 9.8950051 0.5970587   
## b.sm theta.sat   
## 9.879805 0.597817   
## b.sm theta.sat   
## 9.879805 0.597817   
## b.sm theta.sat   
## 9.879805 0.597817   
## b.sm theta.sat   
## 9.8747425 0.5982645   
## b.sm theta.sat   
## 9.8747425 0.5982645   
## b.sm theta.sat   
## 9.8747425 0.5982645   
## b.sm theta.sat   
## 9.8646447 0.5987691   
## b.sm theta.sat   
## 9.8646447 0.5987691   
## b.sm theta.sat   
## 9.8646447 0.5987691   
## b.sm theta.sat   
## 9.8620908 0.5990464   
## b.sm theta.sat   
## 9.8620909 0.5990464   
## b.sm theta.sat   
## 9.8620908 0.5990464   
## b.sm theta.sat   
## 9.8562029 0.5993426   
## b.sm theta.sat   
## 9.8562029 0.5993426   
## b.sm theta.sat   
## 9.8562029 0.5993426   
## b.sm theta.sat   
## 9.8550017 0.5994984   
## b.sm theta.sat   
## 9.8550017 0.5994984   
## b.sm theta.sat   
## 9.8550017 0.5994984   
## b.sm theta.sat   
## 9.8518153 0.5996597   
## b.sm theta.sat   
## 9.8518154 0.5996597   
## b.sm theta.sat   
## 9.8518153 0.5996597   
## b.sm theta.sat   
## 9.8512525 0.5997425   
## b.sm theta.sat   
## 9.8512525 0.5997425   
## b.sm theta.sat   
## 9.8512525 0.5997425   
## b.sm theta.sat   
## 9.8495945 0.5998268   
## b.sm theta.sat   
## 9.8495945 0.5998268   
## b.sm theta.sat   
## 9.8495945 0.5998268   
## b.sm theta.sat   
## 9.8493256 0.5998695   
## b.sm theta.sat   
## 9.8493256 0.5998695   
## b.sm theta.sat   
## 9.8493256 0.5998695   
## b.sm theta.sat   
## 9.8484798 0.5999126   
## b.sm theta.sat   
## 9.8484798 0.5999126   
## b.sm theta.sat   
## 9.8484798 0.5999126   
## b.sm theta.sat   
## 9.8483490 0.5999343   
## b.sm theta.sat   
## 9.8483490 0.5999343   
## b.sm theta.sat   
## 9.8483490 0.5999343   
## b.sm theta.sat   
## 9.8479214 0.5999561   
## b.sm theta.sat   
## 9.8479214 0.5999561   
## b.sm theta.sat   
## 9.8479214 0.5999561   
## b.sm theta.sat   
## 9.847857 0.599967   
## b.sm theta.sat   
## 9.847857 0.599967   
## b.sm theta.sat   
## 9.847857 0.599967   
## b.sm theta.sat   
## 9.847643 0.599978   
## b.sm theta.sat   
## 9.847643 0.599978   
## b.sm theta.sat   
## 9.847643 0.599978   
## b.sm theta.sat   
## 9.8476106 0.5999835   
## b.sm theta.sat   
## 9.8476106 0.5999835   
## b.sm theta.sat   
## 9.8476106 0.5999835   
## b.sm theta.sat   
## 9.8475721 0.5999868   
## b.sm theta.sat   
## 9.8475721 0.5999868   
## b.sm theta.sat   
## 9.8475721 0.5999868   
## b.sm theta.sat   
## 9.8474973 0.5999906   
## b.sm theta.sat   
## 9.8474973 0.5999906   
## b.sm theta.sat   
## 9.8474973 0.5999906   
## b.sm theta.sat   
## 9.8474771 0.5999927   
## b.sm theta.sat   
## 9.8474771 0.5999927   
## b.sm theta.sat   
## 9.8474771 0.5999927   
## b.sm theta.sat   
## 9.847460 0.599994   
## b.sm theta.sat   
## 9.847460 0.599994   
## b.sm theta.sat   
## 9.847460 0.599994   
## b.sm theta.sat   
## 9.8474280 0.5999956   
## b.sm theta.sat   
## 9.8474280 0.5999956   
## b.sm theta.sat   
## 9.8474280 0.5999956   
## b.sm theta.sat   
## 9.8474179 0.5999965   
## b.sm theta.sat   
## 9.8474179 0.5999965   
## b.sm theta.sat   
## 9.8474179 0.5999965   
## b.sm theta.sat   
## 9.8474098 0.5999971   
## b.sm theta.sat   
## 9.8474098 0.5999971   
## b.sm theta.sat   
## 9.8474098 0.5999971   
## b.sm theta.sat   
## 9.8473951 0.5999979   
## b.sm theta.sat   
## 9.8473951 0.5999979   
## b.sm theta.sat   
## 9.8473951 0.5999979   
## b.sm theta.sat   
## 9.8473826 0.5999985   
## b.sm theta.sat   
## 9.8473826 0.5999985   
## b.sm theta.sat   
## 9.8473826 0.5999985   
## b.sm theta.sat   
## 9.8473794 0.5999989   
## b.sm theta.sat   
## 9.8473794 0.5999989   
## b.sm theta.sat   
## 9.8473794 0.5999989   
## b.sm theta.sat   
## 9.8473721 0.5999993   
## b.sm theta.sat   
## 9.8473721 0.5999993   
## b.sm theta.sat   
## 9.8473721 0.5999993   
## b.sm theta.sat   
## 9.8473675 0.5999996   
## b.sm theta.sat   
## 9.8473675 0.5999996   
## b.sm theta.sat   
## 9.8473675 0.5999996   
## b.sm theta.sat   
## 9.8473691 0.5999997   
## b.sm theta.sat   
## 9.8473691 0.5999997   
## b.sm theta.sat   
## 9.8473691 0.5999997   
## b.sm theta.sat   
## 9.8473698 0.5999996   
## b.sm theta.sat   
## 9.8473695 0.5999996   
## b.sm theta.sat   
## 9.8473691 0.5999997   
## b.sm theta.sat   
## 9.8473692 0.5999997   
## b.sm theta.sat   
## 9.8473690 0.5999997   
## b.sm theta.sat   
## 9.8473691 0.5999997   
## b.sm theta.sat   
## 9.8473691 0.5999996

summary(myfit)

##   
## Parameters:  
## Estimate Std. Error t value Pr(>|t|)   
## b.sm 9.8474 1.0150 9.702 <2e-16 \*\*\*  
## theta.sat 0.6000 0.0606 9.900 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.005995 on 742 degrees of freedom  
##   
## Parameter correlation:  
## b.sm theta.sat  
## b.sm 1.0000 -0.9999  
## theta.sat -0.9999 1.0000

# 4. Output

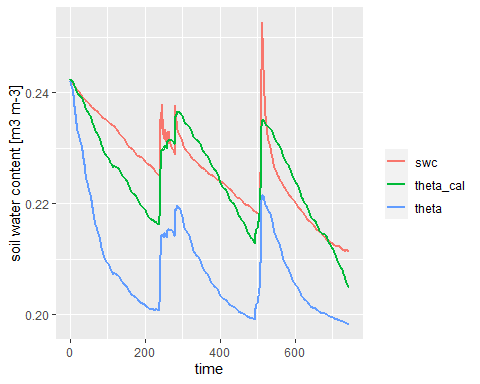
### 4.1 Run model

Finally, we run the model with the calibrated parameters and visualize the output.

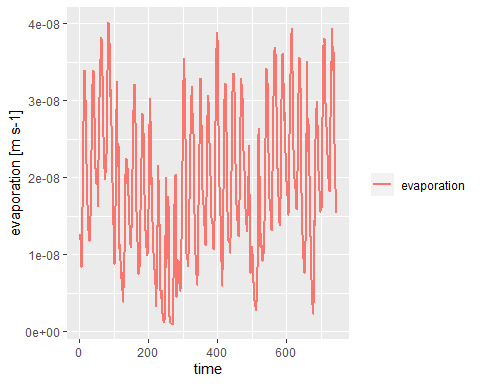
## Run model ##  
  
  
# Run model with uncalibrated parameters  
  
parsfile <- "pars\_soil.csv"  
source("setup\_soilmoisture.R")  
source("fun\_soilmoisture.R")  
  
output <- get\_theta\_soil(input = input, pars = pars, initial\_state = initial\_state)  
  
  
# Run model choosing calibrated parameter file  
  
parsfile <- "pars\_soil\_calib1.csv"  
pars\_soil\_calib1 <- read.csv(parsfile)  
  
pars\_soil\_calib1[2,2] <- 9.85  
pars\_soil\_calib1[33,2] <- 0.6  
  
  
source("setup\_soilmoisture.R")  
source("fun\_soilmoisture.R")  
  
output\_cal <- get\_theta\_soil(input = input, pars = pars, initial\_state = initial\_state)

### 4.2 Plotting

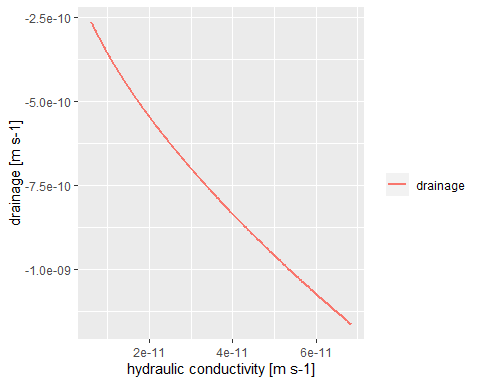
## Plotting ##  
  
library(ggplot2)  
library(gridExtra)  
library(tidyr)  
  
  
# Compare measured data to modeled data and pre-calibrated data  
  
out\_theta <- data.frame(fluxes$time, fluxes$swc, output\_cal$theta, output$theta)  
names(out\_theta) <- c("time", "swc", "theta\_cal", "theta")  
  
gg\_theta <- gather(out\_theta, key = "type", value = "theta", swc, theta\_cal, theta, factor\_key = T)  
  
ggplot() +  
 geom\_line(aes(gg\_theta$time, gg\_theta$theta, color = gg\_theta$type), size = 1) +  
 labs(y = "soil water content [m3 m-3]", x = "time", color = "")



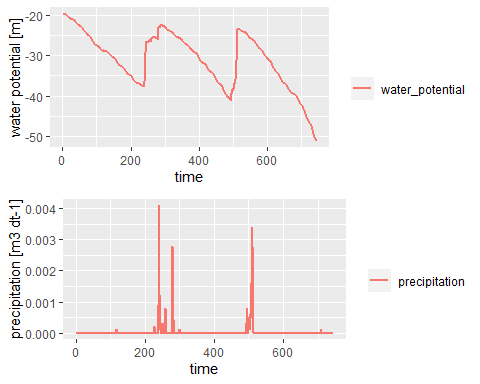
# Plotting output variables  
  
out <- data.frame(fluxes$time, output\_cal[ , -8], input$p)  
names(out) <- c("time", "theta", "runoff", "hyd\_conductivity", "evaporation", "drainage", "water\_potential", "water\_potential\_sub", "precipitation")  
  
  
# evaporation  
  
gg\_theta <- gather(out, key = "type", value = "theta", evaporation, factor\_key = T)  
  
ggplot() +  
 geom\_line(aes(gg\_theta$time, gg\_theta$theta, color = gg\_theta$type), size = 1) +  
 labs(y = "evaporation [m s-1]", x = "time", color = "")



# drainage against hydraulic conductivity  
  
gg\_theta <- gather(out, key = "type", value = "theta", drainage, factor\_key = T)  
  
ggplot() +  
 geom\_line(aes(gg\_theta$hyd\_conductivity, gg\_theta$theta, color = gg\_theta$type), size = 1) +  
 labs(y = "drainage [m s-1]", x = "hydraulic conductivity [m s-1]", color = "")



# water potential and precipitation  
  
gg\_thetaWP <- gather(out, key = "type", value = "theta", water\_potential, factor\_key = T)  
  
WP <- ggplot() +  
 geom\_line(aes(gg\_thetaWP$time, gg\_thetaWP$theta, color = gg\_thetaWP$type), size = 1) +  
 labs(y = "water potential [m]", x = "time", color = "")  
  
  
gg\_thetaP <- gather(out, key = "type", value = "theta", precipitation, factor\_key = T)  
  
P <- ggplot() +  
 geom\_line(aes(gg\_thetaP$time, gg\_thetaP$theta, color = gg\_thetaP$type), size = 1) +  
 labs(y = "precipitation [m3 dt-1]", x = "time", color = "")  
  
grid.arrange(WP, P)



# multiple plots together for comparison  
  
gg\_thetaA <- gather(out, key = "type", value = "theta", theta, factor\_key = T)  
  
A <- ggplot() +  
 geom\_line(aes(gg\_thetaA$time, gg\_thetaA$theta, color = gg\_thetaA$type), size = 1) +  
 labs(y = "theta [m3 m-3]", x = "time", color = "")  
  
gg\_thetaB <- gather(out, key = "type", value = "theta", precipitation, factor\_key = T)  
  
B <- ggplot() +  
 geom\_line(aes(gg\_thetaB$time, gg\_thetaB$theta, color = gg\_thetaB$type), size = 1) +  
 labs(y = "precipitation [m3 dt-1]", x = "time", color = "")  
  
gg\_thetaC <- gather(out, key = "type", value = "theta", water\_potential, factor\_key = T)  
  
C <- ggplot() +  
 geom\_line(aes(gg\_thetaC$time, gg\_thetaC$theta, color = gg\_thetaC$type), size = 1) +  
 labs(y = "water potential [m]", x = "time", color = "")  
  
gg\_thetaD <- gather(out, key = "type", value = "theta", drainage, factor\_key = T)  
  
D <- ggplot() +  
 geom\_line(aes(gg\_thetaD$time, gg\_thetaD$theta, color = gg\_thetaD$type), size = 1) +  
 labs(y = "drainage [m s-1]", x = "time", color = "")  
  
gg\_thetaE <- gather(out, key = "type", value = "theta", hyd\_conductivity, factor\_key = T)  
  
E <- ggplot() +  
 geom\_line(aes(gg\_thetaE$time, gg\_thetaE$theta, color = gg\_thetaE$type), size = 1) +  
 labs(y = "hydraulic conductivity [m s-1]", x = "time", color = "")  
  
grid.arrange(A, B, C, D, E, ncol = 2)

