Supporting Information

Approach for using measured soil gas diffusion coefficients in Hydrus 1D with examples from forest soils

Christoph Haas*, Sinikka Paulus, Martin Maier, Hubert Jochheim, Horst H. Gerke

10.1002/jpln.202000147

jpln202000147-sup-0001-misc_information.pdf

```
#Approach for using measured soil gas diffusion coefficients in Hydrus 1D
with examples from forest soils

#Christoph Haas, Sinikka, Paulus, Martin Maier, Hubert Jochheim, Horst H.
Gerke
#corresponding author: christoph.haas@zalf.de

#See the input file "Be1O.txt" for input data formatting.
#The Levenberg-Marquardt Nonlinear Least-Squares Algorithm Found in
MINPACK (Elzhov et al., 2016) is required.
#Uncomment the following line if the additional package (i.e.,
minpack.lm) is not installed on your system.
#install.packages("minpack.lm")
require(minpack.lm)#require minpack.lm
citation("minpack.lm")#How to cite minpack.lm

getwd()#print actual working direcory
path_to_data_folder<-getwd()#Define the path to the folder containing the
input files; modify this line if your data is not stored in the working
directory; make sure that all txt-files match the required file
formatting.
files<-list.files(path=path_to_data_folder, pattern = ".txt")#Select all
txt-files in the data_folder
setwd(path_to_data_folder)#set working directory

D0_Ne<-0.315 #molecular diffusion coefficient of gas used for the
determination of Ds in free air; here: Neon.

#Create the header of the result file. The tab-separated file can be read
with text editors like NotePad. See l. 35 ff. for more details.
write(file="Results_LMA.out",paste("Horizon\tf\tconfi_2.5\tconfi_97.5\tde
viance\tdf.residual\tlogLik\tmin_air\tmax_air\tnumber_of_sampling_points\
tmean_air\tsd_air\tmin_total_PV\tmax_total_PV\tmean_total_PV\tsd_total_PV
",sep=""), append=F)

for(y in c(1:length(files))){#for loop to process all input files
  print(files[y])#print acutal input file
  name<-strsplit(files[y],".txt")#Use the file name as identifier (i.e.,
Horizon in the result file)
  data<-read.table(file=files[y], header=T, dec=",")#read actual input
file and store the content in data
  data$tau<-data$air_filled^(7/3)/data$total_PV^2 #See Eq. 2 in the
manuscript
  data$DsDO<-data$Ds/D0_Ne #Calculate the relative diffusion coefficient

  #Use the LMA to derive the value of the fitting factor f. Start the
algorithm assuming that f equals 1. For more details; see Eqs. 3-4.
  fm_data_in <- nlsLM(DsDO ~ f *tau,
                      data = data,
                      start = list(f = 1))
  #store the results to "res":
  res<-paste(name,#i.e., the sample identifier
             coef(fm_data_in),#The value of f
             confint(fm_data_in)[1],#The lower and...
             confint(fm_data_in)[2],#... upper borders of the confidence
intervall
             deviance(fm_data_in),#The sum of the squared residual
vector.
             df.residual(fm_data_in),#Degrees of freedom
             logLik(fm_data_in)[1],#The Log Likelihood value
```

```
              min(data$air_filled),#Minimum value of the air-filled
porosity
              max(data$air_filled),#Maximum value of the air-filled
porosity
              length(data$tau),#Number of measured data
              mean(data$air_filled),#Mean value of the air-filled porosity
              sd(data$air_filled),#Standard deviation of the air-filled
porosity
              min(data$total_PV),#Minimum value of the total porosity
              max(data$total_PV),#Maximum value of the total porosity
              mean(data$total_PV),#Mean value of the total porosity
              sd(data$total_PV),#Standard deviation of the total porosity
              sep="\t")#Seperate the results by a tabulator.
  write(file="Results_LMA.out",res, append=T)#Write all results to the
results file.
  ####
if(T==F){#set T==F to plot the data
  sequence<-seq(0.0001,mean(data$total_PV),0.0001)#create a sequence of
numbers ranging from 0.0001 to the mean value of the total porosity
  tau_mod<-
seq(0.0001,mean(data$total_PV),0.0001)^(7/3)/(mean(data$total_PV))^2
#Calculate values of the unfitted model
  DsD0_mod<- tau_mod * coef(fm_data_in)#Calculate values of the fitted
model

  #Create the plot and save it to png file
  png(paste(name, ".png", sep=""))
  par(mar=c(5.5,7.5,4.8,3.5))
  plot(data$DsDO~data$air_filled,ylim=c(0,.5), xlim=c(0,0.9), cex=2,
ylab="", xlab="", axes=F)
  par(new=T)
  plot(DsD0_mod~sequence, col="red", ylim=c(0,.5), xlim=c(0,0.9),
ylab="", xlab="", axes=F, cex=.2)
  par(new=T)
  plot(tau_mod~sequence,  ylim=c(0,.5), xlim=c(0,0.9), ylab="", xlab="",
axes=F, cex=.2, col="black")
  lwd=2
  axis(side = 1,  cex.axis=2, lwd=2, las=1, line=0,srt=45,
at=c(0,.2,.4,.6,.8), labels = c(0,0.2,0.4,0.6,0.8))
  axis(side = 2,  cex.axis=2, lwd=2, las=1, line=0,srt=45)
  mtext(expression(theta[a]),1,2.9, cex=2)
  mtext(expression(paste(D[s], D[0]^-1)),2,3.8, cex=2)
  box(lwd=2)
  dev.off()
}
}
print("Check data folder for results. Open Out-file with a text editor,
e.g., NotePad.")
```

Be10.txt

Horizont	Ds	air_filled	total_PV
Be1O	0,0636724903284105	0,69	0,92
Be1O	0,0748734011640143	0,747	0,92
Be1O	0,0490322181666645	0,645	0,92
Be1O	0,0159715728645724	0,528	0,92
Be1O	0,0789465764464768	0,62	0,83
Be1O	0,086905948778813	0,69	0,83
Be1O	0,06185639740587	0,581	0,83
Be1O	0,0138679065068973	0,41	0,83
Be1O	0,0912789660516978	0,66	0,81
Be1O	0,103286455529467	0,717	0,81
Be1O	0,0338578890629249	0,469	0,81
Be1O	0,0791465848895334	0,62	0,79
Be1O	0,119884522994537	0,673	0,79
Be1O	0,0967315182234266	0,673	0,79
Be1O	0,11223424935116	0,554	0,79
Be1O	0,0610720162254296	0,554	0,79
Be1O	0,00403816252315918	0,355	0,79
Be1O	0,0726114079850124	0,72	0,88
Be1O	0,077355632423191	0,778	0,88
Be1O	0,052367460485848	0,689	0,88
Be1O	0,0836710128209265	0,689	0,88
Be1O	0,0053253001231031	0,478	0,88