Package 'spsh'

October 14, 2022

Type Package

Title Estimation and Prediction of Parameters of Various Soil
Hydraulic Property Models

Version 1.1.0

Date 2020-03-23

Author Tobias KD Weber [aut, cre], Efstathios Diamantopoulos [ctb], Melanie Weynants [ctb]

Maintainer Tobias KD Weber <tobias.weber@uni-hohenheim.de>

Description Estimates model parameters of soil hydraulic property functions by inverting measured data. A wide range of hydraulic models, weighting schemes,
global optimization algorithms, Markov chain Monte Carlo samplers, and extended statistical analyses of results are provided.
Prediction of soil hydraulic property model parameters and common soil properties using pedotransfer functions is facilitated.
Parameter estimation is based on identically and independentally distributed (weighted) model residuals, and simple model selection criteria (Hoege, M., Woehling, T., and Nowak, W. (2018) <doi:10.1002/2017WR021902>) can be calculated.
The included models are the van Genuchten-Mualem in its unimodal, bimodal and trimodal form, the the Kosugi 2 parametric-Mualem model, and the Fredlund-Xing model.
All models can be extended to account for non-capillary water storage and conductivity (Weber, T.K.D., Durner, W., Streck, T. and Diamantopoulos, E. (2019) <doi:10.1029/2018WR024584>.
The isothermal vapour conductivity (Saito, H., Simunek, J. and Mohanty, B.P. (2006) <doi:10.2136/vzj2006.0007>) is calculated based on
volumetric air space and a selection of different tortuosity models:
(Grable, A.R., Siemer, E.G. (1968) <doi:10.2136/sssaj1968.03615995003200020011x>, Lai, S.H., Tiedje J.M., Erickson, E. (1976) <doi:10.2136/sssaj1976.03615995004000010006x>,
Moldrup, P., Olesen, T., Rolston, D.E., and Yamaguchi, T. (1997) <doi:10.1097/00010694-199709000-00004>, Moldrup, P., Olesen, T., Yoshikawa, S., Komatsu, T., and Rolston, D.E. (2004) <doi:10.2136/sssaj2004.7500>,
Moldrup, P., Olesen, T., Yoshikawa, S., Komatsu, T., and Rolston, D.E. (2005) <doi:10.1097/01.ss.0000196768.44165.1f>, Millington, R.J., Quirk, J.P. (1961) <doi:10.1039/TF9615701200>,

Pen-
man, H.L. (1940) <doi:10.1017/S0021859600048164>, and Xu, X, Nieber, J.L. Gupta, S.C. (1992) <doi:10.2136/sssaj1992.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.0.2

LazyData true

Depends DEoptim (>= 2.2.4), lhs (>= 0.16.0), pracma (>= 2.1.4), FME
(>= 1.3.5), hypergeo (>= 1.2-13)

Imports Rdpack

RdMacros Rdpack

NeedsCompilation no

Repository CRAN

Date/Publication 2020-04-06 16:30:02 UTC

R **topics documented:**

---

gofFun *Goodness-of-fit and Information Criteria*

Description

Calculates goodness-of-fit criteria and the likelihood-based Akaike and Bayesian Information Criterion based on a given parameter set, typically from the estimation procedure.

Usage

```
gofFun(
  phat,
  shpmodel = "01110",
  retdata = NULL,
  condata = NULL,
  weight,
  psel,
  ivap.query = NULL,
  hclip.query = FALSE
)
```

Arguments

phat	Vector of non-transformed (back-transformed) model parameters after estimation, i.e. the best fit or maximum likelihood estimate.
shpmodel	Character specifying the soil hydraulic property model.
retdata	Dataframe or matrix with 2 columns. The first with pressure head values in log_10 [cm], i.e. pF values, and the second with volumetric water contents in [cm cm-3].
condata	Dataframe or matrix with 2 columns. The first with pressure head values in log_10 [cm], i.e. pF values, and the second with hydraulic conductivity values log_10 [cm d-1].
weight	List of the model residual weights used or estimated by the parameter estimation scheme, to calculate the weighted statistical analyses.
psel	Vector specifying the selected parameters for the parameter estimation from parL.
ivap.query	Specification of *ivap* method, if FALSE selected, no isothermal vapour conductivity is consideredSee Details
hclip.query	Implemented purely for future compatability. Currently no use. See Details

Details

Output for data groups.

th   list with goodness of fit statistics for the retention curve *see below*

logKh   list with output same as th but for the log_{10} fitted conductivity curve
combined   list with AIC, AICc, and BIC calculated for the multi-objective function if arguments retdata and condata are

Statistical analyses of the inverse modelling results.

me	mean (weighted) error
mae	mean absolute (weighted) error
mse	mean squared (weighted) error
rss	sum of squared (weighted) errors
rmse	root mean squared (weighted) error
AIC	Akaike Information Criteria
AICc	corrected Akaike Information Criteria
BIC	Bayesian Information Criteria
m	number of observations

Author(s)

Tobias KD Weber , <tobias.weber@uni-hohenheim.de>

References

HÃƒÂ¶ge M, WÃƒÂ¶hling T, Nowak W (2018). "A Primer for Model Selection: The Decisive Role of Model Complexity." *Water Resources Research*, **54**(3), 1688–1715. doi: 10.1002/2017WR021902.

Examples

```
data("shpdata1")
retdata <- shpdata1$TS1$wrc
condata <- shpdata1$TS1$hcc
condata <- condata[!is.na(condata[,1]),]
# Parameter list
parL <- list("p" = c("thr"= 0.05, "ths" = 0.45, "alf1" = 0.01, "n" = 2, "Ks" = 100, "tau" = .5),
             "psel" = c(1, 1, 0, 1, 1, 1),
             "plo" = c(0.001 , 0.2, 0.001, 1.1, 1, -2),
             "pup" = c(0.3, 0.95, 1, 10, 1e4, 10)
             )
# Calulation of the goodness of fit.
gofL <-gofFun(parL$p, shpmodel = "01110", retdata = retdata, condata = condata,
              weight = weightFun(weightmethod = "fix1"), parL$psel,
              ivap.query = NULL, hclip.query = FALSE)
```

Ibeta   *Incomplete Beta Function*

Description

Calculation of the incomplete beta function used in sncFun.01110

Usage

```
Ibeta(z, a, b)
```

Arguments

z	Vector of n model parameters.
a	Vector of n model parameters.
b	Vector of n model parameters.

Details

If a=1 or a=1/n and b=0, this implementation cannot evaluate values for z < 1.014.

Value

Returns a vector of numerical values

Author(s)

Tobias KD Weber , <tobias.weber@uni-hohenheim.de>

Examples

```
x = seq(.5, 4.2 , length = 20)
alf = 0.1
n = 2
y = 1 + (alf * 10^x)^n

result <- Ibeta(z = y, a = 1, b = 0 )
```

---

inipopFun	*Generates an Initial Population of Transformed Soil Hydraulic Prop-erty Model Parameters*

Description

Draws a Latin Hypercube Sample from a set of uniform distributions in the transformed parameter space,in creating a Latin Hypercube Design. This function uses the Columnwise Pairwise (CP) algorithm to generate an optimal design with respect to the S optimality criterion, as implemented in lhs-package.

Usage

```
inipopFun(p, psel, plo, pup, trans.L, Npop = NA)
```

Arguments

p	vector of model parameters
psel	vector of selectors
plo	vector of lower boundary values of non-transformed parameters
pup	vector of upper boundary values of non-transformed parameters
trans.L	list of transformation/backtransformation operators with same length as p, psel, plo, and pup.
Npop	integer of initial population size

Details

Produces and optimum latin hypercube sample from a bounded uniform distribution.

Value

n draws of k parameters in an n x k Latin Hypercube Sample matrix with values uniformly distributed on user specified bounds.

Author(s)

Tobias KD Weber , `<tobias.weber@uni-hohenheim.de>`

See Also

[optimumLHS](#)

Examples

```
# Example based on soil hydraulic property model parameters of shpmodel = "01110" parameters
parL <- list("p" = c("thr"= 0.05, "ths" = 0.45, "alf1" = 0.01, "n" = 2, "Ks" = 100, "tau" = .5),
             "psel" = c(1, 1, 0, 1, 1, 1),
             "plo" = c(0.001 , 0.2, 0.001, 1.1, 1, -2),
             "pup" = c(0.3, 0.95, 1, 10, 1e4, 10)
)
# rules for the parameter transformation
ptransfit<- c(function(x)x, function(x)x,log10,
              function(x)log10(x-1),log10 , function(x)x)
# get latin hypercube sample.
test.inipop <- inipopFun(parL$p, parL$psel,
                         parL$plo, parL$pup, ptransfit, Npop = 20)
# plot the latin hypercube
pairs(test.inipop)
```

KvapFun   *Calculates the Isothermal Water Vapour Conductivity*

Description

Calculates the isothermal vapour conductivity as a function of modelled volumetric air content. Different models are implemented enabling the calculation of the relative gas diffusion coefficient (Ds/Do), based on different expressions for an effective tortuosity.

Usage

```
KvapFun(
  p,
  por = p[2],
  retFun = NA,
  theta = NA,
  model = "MQ61",
  Temp = 20,
  m = 3,
  pF = seq(-3, 7, length = 501),
  output = "log10",
  ...
)
```

Arguments

p	vector of soil hydraulic property model parameters, cf resp soil hydraulic property model for details.
por	skalar value giving the fraction of a porous' media porosity [-](value between [0, 1]), defaults to the saturated water content.
retFun	soil hydraulic property function has to be specified if models PMQ, TPM or TPEM are used, necessary to calculate the air content at h = 100 cm for the parameter eps100.
theta	vector of numerical volumetric water contents [0,1] at which the air content is to be calculated.
model	Implemented models (specify as character):

B	Buckingham (1904)
P	Penman (1940)
MQ60	Millington and Quirck (1960)
MQ61	Millington and Quirck (1961)
GS	Grable and Siemer (1968)
L	Lai et al. (1976)
PMQ	Moldrup et al. (1997)
TPM	Moldrup et al. (2004)
TPEM	Moldrup et al. (2005)

Temp	Soil tempereature [deg C], defaults to 20.
m	PMQ model parameter, default `m = 3`.
pF	monotonically increasing pF values, defined as log10(⎮ pressure head [cm]).
output	Defaults to `log10` indicates the isothermal vapour conductivity is returned as $log_1 0$(conductivity), if `ouput != log10`, the output will be in non-transformed values.
...	more arguments to be passed to `retFun`

Details

More reading on the models reference is made to suggested in (Weber et al. 2019)

Author(s)

Tobias KD Weber , `<tobias.weber@uni-hohenheim.de>`

References

Weber TK, Durner W, Streck T, Diamantopoulos E (2019). "A modular framework for modelling unsaturated soil hydraulic properties over the full moisture range." *Water Resources Research*. ISSN 0043-1397, doi: 10.1029/2018WR024584.

Buckingham, E. (1904). Contributions to Our Knowledge of the Aeration Status of Soils, Bulletin 25, USDA Bureau of Soils, Washington, DC.

Grable, A.R.; Siemer, E.G. (1968).Effects of Bulk Density, Aggregate Size, and Soil Water Suction on Oxygen Diffusion, Redox Potentials, and Elongation of Corn Roots. Soil Sci. Soc. Am. Proc., 32, pp. 180-186. <doi:10.2136/sssaj1968.03615995003200020011x>.

Lai, S.H.; Tiedje J.M.; Erickson, E. (1976). In situ Measurement of Gas Diffusion Coefficient in Soils. Soil Sci. Soc. Am. J., 40, pp. 3-6. <doi:10.2136/sssaj1976.03615995004000010006x>.

Moldrup, P.; Olesen, T.; Rolston, D.E.; and Yamaguchi, T. (1997). Modeling Diffusion and Reaction in Soils: Vii. Predicting Gas and Ion Diffusivity in Undisturbed and Sieved Soils. Soil Science. 162 (9): pp. 632-640.

Moldrup, P.; Olesen, T.; Yoshikawa, S.; Komatsu, T.; and Rolston, D.E. (2004). Three-Porosity Model for Predicting the Gas Diffusion Coefficient in Undisturbed Soil. Soil Sci. Soc. Am. J. 68 (3).pp. 750-759. <doi:10.2136/sssaj2004.7500>.

Moldrup, P.; Olesen, T.; Yoshikawa, S.; Komatsu, T.; and Rolston, D.E. (2005). Predictive-Descriptive Models for Gas and Solute Diffusion Coefficients in Variably Saturated Porous Media Coupled to Pore-Size Distribution: II. Gas Diffusivity in Undisturbed Soil. Soil Sci., 170, pp. 854-866. <doi:10.1097/01.ss.0000196768.44165.1f>.

Millington, R.J.; Quirk, J.P. (1960). Millington, R. J., and Quirk. J.M. Transport in porous media. pp. 97-106. In: F.A. Van Beren, et al. (ed.) Trans. Int. Congr. Soil Sci., 7 th, Vol. 1, Madison, Wl. 14-24 Aug. 1960. Elsevier, Amsterdam.

Millington, R.J.; Quirk, J.P. (1961). Permeability of Porous Solids. Trans. Faraday Soc., 1961, 57, pp. 1200-1207. <doi:10.1039/TF9615701200>.

Penman, H.L. (1940). Gas and vapour movements in the soil: I. The diffusion of vapours through porous solids. J. Agric. Sci., 30: pp. 437-462. <doi:10.1017/S0021859600048164>.

Xu, X; Nieber, J.L. Gupta, S.C. (1992). Compaction Effect on the Gas Diffusion Coefficient in Soils. Soil Sci. Soc. Am. J.,56, pp. 1743-1750. <doi:10.2136/sssaj1992.03615995005600060014x>.

Examples

```
# | pressure head |
pF <- seq(-3, 7, length = 201)
h <- 10^pF
# van Genuchten-Mualem model parameters
p <- c(0.08, .42, .05, 1.5, 100, .5)
# calculate soil hydraulic property values
shypL <- shypFun.01110(p, h)
# clculate the isothermal vapour conductivity
kvap <- KvapFun(p, por = p[2], retFun = NA, theta = shypL$theta, model = "MQ61",
                Temp = 20, m = 3, pF, output = "log10")
```

---

logLikFun.norm	*Calculation of the Log-likelihood assuming Identically, Independenzly and Normally Distributed errors*

Description

Calculates the i-th log-likelihood of each *y-yhat* pair as described in (Seber and Wild 2004).

Usage

```
logLikFun.norm(y, yhat, sigma)
```

Arguments

y	A vector of n observed properties/variables of interest.
yhat	A vector of n model simulated properties/variables of interest.
sigma	A vector of length 1 considering homoscedastic residuals.

Details

The underlying assumption is, that the model residuals (errors) are independently, and identically distributed (i.i.d.) following a normal distribution. Alternatively consider using dnorm.

Value

log-likelihood value of an normal distribution with N~(0, *sigma^2*)

Note

The assumption of i.i.d. and normal distribution is best investigated *a posteriori*.

Author(s)

Tobias KD Weber , `<tobias.weber@uni-hohenheim.de>`

References

Seber GAF, Wild CJ (2004). *Nonlinear regression*, Wiley series in probability and mathematical statistics. Wiley, New York. ISBN 978-0-471-47135-6.

Examples

```
# homoscedastic residuals
sig.s  <- .01
y.scat <- rnorm(100, 0, sig.s)
yhat   <- (1:100)^1.2
y      <- yhat + y.scat
sum(logLikFun.norm(y, yhat, sig.s))
plot(yhat-y)
```

---

numMualem   *Function to Numerically Compute the Mualem Integral*

Description

This function will calculate numerically Mualems Integral (Mualem 1976) and return Hydraulic Conductivity Values.

Usage

```
numMualem(h, scap, pcon = NA)
```

Arguments

h	vector of length l pressure head values.
scap	Capillary saturation [cm3 cm-3].
pcon	vector of soil hydraulic conductivity model parameters, the first argument is q and the second r.

Details

The numerical solution of Mualems integral relies on the trapezoidal rul of integration.

Value

returns a vector of length l of calulcated conductivity values at h.

Author(s)

Tobias KD Weber , `<tobias.weber@uni-hohenheim.de>`

References

Mualem Y (1976). "New Model for predicting hydraulic conductivity of unsaturated porous media." *Water Resources Research*, **12**(3), 513–522. ISSN 0043-1397, http://www.scopus.com/inward/record.url?eid=2-s2.0-0016961814&partnerID=40&md5=e3773c228d43852dad22d656738389c9.

Examples

```
h <- 10^seq(-3, 6.8, length = 501)
p = c(.05, .5, .01, 1.8, 100, .5)
shyp.L <- shypFun.01110(p, h)

Ks <- p[5]
tau <- p[6]
Se <- shyp.L[['Se']]
Khrnum <- numMualem(h, pcon = tau, scap = Se)

Khnum <- Ks * Se^tau * Khrnum

plot(log10(h), log10(shyp.L[['Kh']]), ylim = c(-10, 2.3),
    xlim = c(-1,6), ylab = "log10 Kunsat [ cm/d ]", xlab = "pF [ - ]", type = "l", lwd = 8)
lines(log10(h), log10(Khnum), col = "red", lwd = 2)
```

---

ptf.cW | *Title Corrected Weynants et al. (2009) Pedotransfer Function*

Description

Title Corrected Weynants et al. (2009) Pedotransfer Function

Usage

```
ptf.cW(CLAY, SAND, BD, OC)
```

Arguments

CLAY — A vector of n elements with soil clay content (particle diameters <= 2 x 10e-6 m), in percent [0, 100]).

SAND — A vector of n elements with soil sand content (particle diameters < 2 mm and > 50 x 10e-6 m), in percent [0, 100]).

BD — A vector of n elements with soil bulk density (g/cm3).

OC — A vector of n elements with soil organic carbon content, in percent [0, 100].

Value

Pedotransfer function returns the van Genuchten - Mualem model parameters given CLAY, SAND, BD, and OC. The correction of the original paper presented by (Weynants et al. 2009), were made by (Weihermüller et al. 2017), which is implemented.

Note

The PTF is not suitable for predicting the hydraulic conductivity curve at pressured heads > -6 cm (Weynants et al. 2009).

Author(s)

Melanie Weynants, <mweynants@gmail.com>

Tobias KD Weber , <tobias.weber@uni-hohenheim.de>

References

Weynants M, Vereecken H, Javaux M (2009). "Revisiting Vereecken Pedotransfer Functions: Introducing a Closed-Form Hydraulic Model." *Vadose Zone Journal*, **8**(1), 86. ISSN 1539-1663, doi: 10.2136/vzj2008.0062. WeihermÃ¼ller L, Herbst M, Javaux M, Weynants M (2017). "Erratum to "Revisiting Vereecken Pedotransfer Functions: Introducing a Closed-Form Hydraulic Model"." *Vadose Zone Journal*, **16**(1), 0. ISSN 1539-1663, doi: 10.2136/vzj2008.0062er.

Examples

```
result <- ptf.cW(CLAY = .4, SAND = .4, BD = 1.6, OC = .5)
```

---

ptf.vG2BW   *Parameter Transfer Function for Weber et al.(2019) model.*

Description

Predicts (Weber et al. 2019) model parameters in the van Genuchten-Mualem variant *01110FM*, from given van Genuchten-Mualem parameters for the constrained van Genuchten-Mualem model.

Usage

```
ptf.vG2BW(x)
```

Arguments

x   vector of 6 van Genuchten-Model parameters. The order is sensitive see for
                shypFun.01110

thr	Residual water content (-), alway equal to zero
ths	Saturated water content (-)
alf1	Shape parameter (cm^-1)
n1	Shape parameter (-)
Ks	Hydraulic conductivity at 0 potential (cm/day)
tau	Shape parameter (-)

Details

Pedotransfer function returns the van Genuchten - Mualem model `shypFun.01110` parameters in the Brunswick-Model variant *01110FM*, based on previously determined van Genuchten-Mualem parameters. The transfer function is based on an ordinary linear regression between the i-th *01110* and *01110FM*. The paraemeters were based on model parameter estimation to a dataset of >400 samples with retention and conductivity measurements.

Value

thsnc	Saturated water content of the non-capillary part.
thsc	Saturated water content of the capillary part.
alf	Shape parameter (cm^-1)
n	Shape parameter (-)
Ksc	Saturated hydraulic conductivity of the capillary part [cm day-1]
tau	Shape parameter (-)
Ksnc	Saturated hydraulic conductivity of the non-capillary part [cm day-1]
a	slope of the non-capillary unsaturated conductivity [-]
h0	anker point at which the water content is 0

Note

The parameter transfer function was derived by weighted linear regression with weights in x and y, by regressing the estimated the Brunswick model parameters in the Genuchten Mualem variant (Weber et al. 2019) inferred from matching measured soil water retention and hydraulic conductivity data against the constrained van Genuchten model parameters. The regression of alf1 and Ks, and (n-1) was done in the log[10])-transformed space, and Kncs is a fixed value of 10^-1.72.

Author(s)

Efstathios Diamantopoulos , <ed@plen.ku.dk>

Tobias KD Weber , <tobias.weber@uni-hohenheim.de>

See Also

[shypFun](#)

Examples

```
p      <- c(0.08, 0.42, 0.01, 1.5, 100, 0.5)
result <- ptf.vG2BW(p)
```

---

resFun   *Calculation of the Objective Function Value*

Description

Contains the objective functions to calculate (weighted) sum of squared residuals or likelihoods. The assumption made is that the residuals are identically, independantly and normally distributed. The normal distribution of the model residuals is standardly given with mean = 0, and variance = 1.if weighting is not considered. There are three methods to consider weights: a) fixed skalar values for each data type, b) a vector of weights for each data type. The vector has to have the same length as the vector of the data type. c) It is possible to estimate the

Usage

```
resFun(
  p,
  shpmodel = "01110",
  retdata = NULL,
  condata = NULL,
  pretrans = NULL,
  weight = NULL,
  method = "rss",
  trim.query = FALSE,
  ivap.query = NULL,
  hclip.query = FALSE,
  parL = NULL
)
```

Arguments

p	Vector of model parameters handed used to calculate the soil hydraulic property model values in shypFun. Depends on shpmodel and the pressure head values specified in retdata and condata
shpmodel	Character identifying the soil hydraulic property model. See shypFun.
retdata	A dataframe or matrix with 2 columns. The first with pressure head values in [cm] and the second with volumetric water contents in [cm cm-3].
condata	A dataframe or matrix with 2 columns. The first with pressure head values in [cm] and the second with hydraulic conductivity values in log10[cm d-1].
pretrans	A vector to back transform the parameters before the soil hydraulic property function values calculated.
weight	Specification of weight method. See weightFun.

rss	Default for the optimisation algortihm DEoptim. resFun returns skalar sum of squared (weighted) residuals.
res	Weighted residuals are computed and returnd. Required for post hoc analyses.
-2logLik	-2 * log-likelihood is returned.

method	
trim.query	Default FALSE. If a trimodal soil hydraulic property function is used, this has to be specified by setting the argument to (*TRUE*) which ensures the sum of modal weights == 1.
ivap.query	Default is FALSE, and no *ivap* method is specified. See KvapFun
hclip.query	Implemented purely for future compatability.
parL	Defaults to NULL, only inserted for compatbility with modMCMC used in shypEst-Fun. modMCMC, only handled parameters which are estimated, other model parameters need to be passed through parL. See Details of shypEstFun.

Details

Model errors may be specified or estimated as nuisance parameters weighting the data classes. In case the model error !=1, the output statistics are weighted accordingly.

Value

Returns scalar of sum of squared (weighted) residuals or vector of weighted residuals, as specified by

user	user defined weights
none	no weights are considered, i.e. no measurement error assumed
range	rescaling (normalization of observations to the intervall [0,1]
fix1	fixed scalar weight for THETA is 1/0.05^2 and weight for log10K is 1
est1	Two scalar model weights 1/sigma_i^2 are treated as free parameters to be estimated by inversion, one for THETA and

Examples

```
# load data
data("shpdata1")

# observations
retdata <- shpdata1$LFH1$wrc[!is.na(shpdata1$LFH1$wrc[,1]),]
condata <- shpdata1$LFH1$hcc

# 7 - resFun --------------------------------------------------------
# soil hydraulic property model parameters, van Genuchten-Mualem
p <- c("thr" = 0.16, "ths" = 0.46, "alf1" = 0.03, "n1" = 1.42, "Ks" = 26, "tau" = .5)

# calculate weighted residuals
wres <- resFun(p, retdata = retdata, condata = condata, pretrans = NULL,
               weight = list("wth" = 0.0025, "wKh" = 1), method = "res", trim = FALSE)

## residuals of the soil water retention curve [-]
theta.wres <- wres[1:dim(retdata)[1]]

## residuals of the log10 hydraulic conductivity curve [cm/d]
log10K.wres <- wres[(dim(retdata)[1]+1) : length(wres)]
```

Description

Data from evaporation experiments using the UMS HYPROP device on two soils with different textures Data ist reported in [cm3 cm-3]

Usage

```
data(shpdata1)
```

Format

An object of class /code"list".

Source

References

Kettcheson, Price, Weber (2018): Initial variability, evolution and model parameterization of the soil hydrophysical properties of a reclaimed oil sands watershed and constructed wetland, in revision.

Examples

```
data(shpdata1)
ticksatmin <- -2
tcllen <- 0.4
ticksat    <- seq(ticksatmin,5,1)
pow <- ticksatmin:5

TS.wrc <- shpdata1$TS1$wrc; TS.hcc <- shpdata1$LFH1$wrc
TS.hcc <- shpdata1$TS1$hcc; LFH.hcc <- shpdata1$LFH1$hcc

# PLOT THE MEASURED WATER RETENTION CURVE
plot(TS.wrc[,1:2]  , ylim = c(.1,.50), xlim = c(0,4), ylab = "", xlab = "",
col = "darkgrey", axes=FALSE, main = "Measured Water Retention Curve")
points(TS.wrc[,1:2], pch = 4,col = "darkblue")
legend("topright", c("TS1", "LFH1"), pch = c(1,4), bty = "n", cex=1.2,
col = c("darkgrey","darkblue"))
axis(1, at = pow, labels = parse(text = paste('10^',(pow), sep = "")),tcl = tcllen)
axis(2, tcl = tcllen)
axis(4, labels = NA, tcl = tcllen)
axis(3, labels = NA, tcl = tcllen)
mtext("pressure head |h| [cm]", 1, cex = 1.2, line = 2.8)
mtext("vol. water content [ - ]", 2, cex = 1.2, line = 2.8)
```

```
box()

# PLOT THE MEASURED HYDRAULIC CONDUCTIVITY CURVE

plot(TS.hcc, ylim = c(-8,2), xlim = c(0,4), ylab = "", xlab = "", col = "darkgrey",
axes = FALSE, main = "Measured Hydraulic Conductivity Curve")
points(TS.hcc, pch = 4, col = "darkblue")

legend("topright", c("TS1", "LFH1"), pch = c(1,4), bty = "n", cex = 1.2,
col = c("darkgrey","darkblue"))

axis(1, at = pow, labels = parse(text = paste('10^',(pow), sep = "")), tcl = tcllen)
axis(2, tcl = tcllen)
axis(4, labels = NA, tcl = tcllen)
axis(3, labels = NA, tcl = tcllen)

mtext("log10 K [cm/d]", 2, cex = 1.2, line = 2.8)
mtext("pressure head |h| [cm]",1 , cex = 1.2, line = 2.8)
box()
```

---

shypEstFun	*Wrapper function for the Estimation of Soil Hydrologic Property Model Parameters*

Description

Estimates model parameters of implemented soil hydraulic property functions. This function sets up the parameter estimation, given a set of arguments, and enables minimisation of (weighted) sum of squared residuals, assuming independent and identically distributed model residuals. More information on the options is given in the *Details*

Usage

```
shypEstFun(
  shpmodel = "01110",
  parL,
  retdata,
  condata,
  ivap = NULL,
  hclip = FALSE,
  weightmethod = "none",
  LikModel = "rss",
  ALG = "DE",
  set.itermax = 200,
  ALGoptions = NULL,
  lhs.query = FALSE
)
```

Arguments

Character specifying the soil hydraulic property model. Currently valid models as documented in shypFun and are:
01110 constrained unimodal van Genuchten-Mualem.
01210 constrained bimodal van Genuchten-Mualem.
01310 constrained trimodal van Genuchten-Mualem.
02110 unimodel Kosugi 2 parametric-Mualem model (Kosugi, 1996)
03110 unimodel Fredlund-Xing-Mualem model, with the contraint of m = 1-1/n (Fredlund D.G., and A. Xing, 1994)

shpmodel	
parL	`list` of 4 vectors with named vectors, the order in the list is sensitive.

p	vector with length `l` of model specific initial parameters, has to coincide with the chosen soil hydraulic property model
psel	vector with length `l` identifying which parameters are to be estimated
plo	vector of lower bounds (non-transformed parameter boundaries)
pup	vector of upper bounds (non-transformed parameters boundaries)

retdata	A dataframe or matrix with 2 columns. The first with log10 values of pressure head values in [cm] and the second with volumetric water contents in [cm cm-3].
condata	A dataframe or matrix with 2 columns. The first with log10 values of pressure head values in [cm] and the second with hydraulic conductivity values log10[cm d-1].
ivap	Specification if isothermal vapour conductivity after Saito et al. (2006) is accounted, defaults to `NULL` and no isothermal vapour conducitvity is considered. See Details.
hclip	Implemented for future development reasons and is not yet functional. Specification if the hydraulic conductivity model should be 'clipped', i.e. constrained to a maxium pore diamater as introduced by Iden et al. (2015), defaults to `FALSE`.
weightmethod	Specification of weight method. The implemented methods are

none	no weights are considered, i.e. no measurement error assumed
range	normalization of observations to the intervall [0,1]
fix1	fixed scalar weight for THETA is 0.05^2 and weight for log10K is 1
est1	Two scalar model weights ($1/sigma^2$) are treated as free parameters to be estimated by inversion, one for THETA an

	Alternatively, a list of vectors can be provided specifying the user given model weights ($1/sigma^2$). Either as skalar for each data class, or a vector with the same length as the number of data points given for each of the measurements in the respective data class. The length of the list has to coincide with the data groups.
LikModel	Specification of inverse modelling type. Has to be specified but implemented for future compatability)

rss	Default for the optimisation algorithm `DEoptim`. `resFun` returns skalar sum of squared (weighted) residuals
-2loglik	Specified if `ALG == -2*log-likelihood` value, which is minimised assuming Gaussian and i.i.d (weighted) resid

ALG	Select global optimisation algorithm or a Markov chain Monte Carlos (MCMC) sampler.

DE	Default for the optimisation algortihm DEoptim. resFun returns a skalar sum of squared (weighted) residuals if Lik
modMCMC	Default for the DRAM (Delayed Rejection Adaption Metropolis) algrothim implemented in modMCMC of the FME p

set.itermax	Integer specifying the maximum number of iterations default = 200.
ALGoptions	A list with named entries setting the algorithm options. Each list element name is required to be identical with the names as documented in the respective algo-rtihm help [DEoptim.control](#) and [modMCMC](#). set.itermax overrides the maximum iterations argument.
lhs.query	default FALSE, TRUE will produce a Latin Hypercube Sample for the initial pop-ulation when using DEoptim.

Details

Several in-built methods for weighting the (multi-) objective function residuals are available, they may be specified, or estimated as nuisance parameters for the two data groups. More details see weightFun. Weights are the inverse of the squared standard deviation of the residuals (variance).

Generally, soil hydraulic property model parameters are estimated as transformed parameters: log10 for alpha_i, Ks, and log10 for n_i-1, Kc, Knc

For model codes in *ivap* please refer to [KvapFun](#).

Parallel computing for package DEoptim is not supported. And the optional arguments in modMCMC are not supported.

Value

list returns the result of the optimisation algrorithm or MCMC sampler and all settings.

settings	a list with output of the optimisation and summary of settings:

weigth	the list with weights for the retention and conductivity data.
parL	the list of initial and selected model parameters, and upper and lower bounds.
transL	list of parameter transformation rules used
shpmodel	the used soil hydraulic property model
ivap	isothermal vapour conductivity model
hclip	for future compatability
LikModel	the adopted method to calculate the objective function value
data	a list of 2 objects with a) retention data and b) conductivity data used for the parameter estimation.

out	result of algorithm function DEoptim or modMCMC

Examples

```
## Not run:
data("shpdata1")
retdata <- shpdata1$TS1$wrc
```

```
condata <- shpdata1$TS1$hcc
condata <- condata[!is.na(condata[,1]),]

weightmethod <- "range"
ivap          <- NULL
set.itermax   <- 1
LikModel      <- "rss" # ALTERNATIVE OPTION: LikModel = "-2logLik"
ALG           <- "DE"        # ALTERNATIVE OPTION: ALG = "modMCMC"

parL<-list("p"=c("thr"=0.05,"ths"=0.45,"alf1"=0.01,"n"=2,"Ks"=100,"tau"=.5),
           "psel" = c(1, 1, 1, 1, 1, 1),
           "plo"= c(0.001 , 0.2 , 0.001 , 1.1, 1, -2),
           "pup"= c(0.3 , 0.8 , .1, 11 , 1e4, 10))

out <- shypEstFun(shpmodel = "01110",
                  parL = parL,
                  retdata = retdata, condata = condata,
                  ivap = ivap,
                  hclip = FALSE,
                  weightmethod = weightmethod,
                  LikModel = LikModel,
                  ALG = ALG,
                  set.itermax = set.itermax,
                  lhs.query = FALSE)
\dontshow{
}
\donttest{
data("shpdata1")
retdata <- ret <- shpdata1$TS1$wrc
condata <- con <- shpdata1$TS1$hcc
condata <- condata[!is.na(condata[,1]),]

---

# 1 SET VARIABLES --------------------
# VARIABLES FOR PLOTTING
{pF <- seq(-3, 6.8, length = 201)
h <- 10^pF
ticksatmin <- -2
tcllen <- 0.4
ticksat <- seq(ticksatmin,5,1)
pow <- ticksatmin:6

# VARIABLES FOR THE FITTING ALGORITHM
weightmethod = "range"
ivap = NULL
set.itermax = 3e1
LikModel = "rss" # ALTERNATIVE OPTION: LikModel = "-2logLik"
ALG = "DE"        # ALTERNATIVE OPTION: ALG = "modMCMC"
shpmodel.v <- c("01110", "01110FM")

plot.query = FALSE
no.shps <- length(shpmodel.v)
```

```
#  initialising lists
out.L <- vector("list", no.shps)
gof.L <- vector("list", no.shps)
}
# Run comparison
for (i in 1:2) {
    shpmodel = shpmodel.v[i]
    # INITIAL PARAMETERS, BOUNDS, and SELECTED PARAMETERS FOR FITTING
    switch(shpmodel,
    "01110" = {

        # van Genuchten-Mualem Model parameters
        parL<-list("p"=c("thr"=0.05,"ths"=0.45,"alf1"=0.01,"n"=2,"Ks"=100,"tau"=.5),
                   "psel" = c(1, 1, 1, 1, 1, 1),
                   "plo"= c(0.001 , 0.2 , 0.001 , 1.1, 1, -2),
                   "pup"= c(0.3 , 0.8 , .1, 11 , 1e4, 10)
        )
    },

    "01110FM" = {

        # van Genuchten-Mualem Model parameters + BRUNSWICK MODEL
        parL<-list("p"=c("thr"=0.05,"ths"=0.45,"alf1"=0.01,"n"=2,"Ksc"=100,
                        "tau"=.5,"Ksnc"=1e-4,"a"=1.5,"h0"=6.8),
                   "psel" = c( 1,1, 1 ,1 , 1,1,1, 0, 0),
                   "plo"= c(0.001 , 0.1 , 0.001 , 1.1, 1,0,1e-6 , 1, 6.5),
                   "pup"= c(0.35, 0.7 , .1, 11 , 1e4,10 ,1e0, 2, 6.9)
        )
    },
    stop("Enter a meaningful shpmodel")
    )

    out <- shypEstFun(shpmodel = shpmodel,
                parL = parL,
                retdata = retdata, condata = condata,
                ivap = ivap,
                hclip = FALSE,
                weightmethod = weightmethod,
                LikModel = LikModel,
                ALG = ALG,
                set.itermax = set.itermax
                ,lhs.query = FALSE)

    out$model <- shpmodel.v[i]
    out.L[[i]] <- out


    #  Calculate the soil hydraulic properties for the plot
    if(ALG == "DE"){
        p <- out$out$optim$phattrans
    }
```

```
if(ALG == "modMCMC"){
      p <- out$out$bestpartrans
}

if(weightmethod =="est1"){
      np <- length(p)
      p <- p[-c(np-1, np)]
      if(ALG =="modMCMC"){
            parL$p[which(parL$psel==1)] <- p
            p <- parL$p
      }
}

if(plot.query==TRUE){

      shyp.L<-shypFun(p,h,shpmodel=shpmodel.v[i],ivap.query=ivap)

      if(shpmodel == c("01110")){

            wrc<-shyp.L$theta
            hcc<-log10(shyp.L$Kh)

            # PLOT THE WATER RETENTION CURVE
            par(mfrow=c(1,2),tcl=tcllen)
            plot(retdata,ylim=c(.0,.50),xlim=c(0,6.8),ylab="",xlab="",
                  col="darkgrey",axes=FALSE,main="WaterRetentionCurve",cex=2)
            lines(log10(abs(h)),wrc,col="darkblue",lwd=2)
            legend("topright",c("observed","simulated"),pch=c(1,NA),
                  lty=c(NA,1),lwd=2,bty="n",cex=1.3,col=c("darkgrey","darkblue"))
            axis(1,at=pow,labels=parse(text=paste('10^',(pow),sep="")),tcl=tcllen)
            axis(2,tcl=tcllen)
            axis(4,labels=NA)
            axis(3,labels=NA)
            mtext("pressurehead|h|[cm]",1,cex=1.2,line=2.8)
            mtext("vol.watercontent[-]",2,cex=1.2,line=2.8)
            box()

            # PLOT THE MEASURED HYDRAULIC CONDUCTIVITY CURVE
            plot(condata,ylim=c(-8,2),xlim=c(0,6.8),ylab="",xlab="",col="darkgrey",
                  axes=FALSE,main="HydraulicConductivityCurve",cex=2)
            lines(log10(abs(h)),hcc,col="darkblue",lwd=2)
            legend("topright",c("observed","simulated"),pch=c(1,NA),
                  lty=c(NA,1),lwd=2,bty="n",cex=1.3,col=c("darkgrey","darkblue"))
            axis(1,at=pow,labels=parse(text=paste('10^',(pow),sep="")),tcl=tcllen)
            axis(2)
            axis(4,labels=NA)
            axis(3,labels=NA)
            mtext("log10K[cm/d]",2,cex=1.2,line=2.8)
            mtext("pressurehead|h|[cm]",1,cex=1.2,line=2.8)
            box()
            par(mfrow=c(1,1))

      }
```

```
            if(shpmodel == "01110FM"){

                    wrc<-shyp.L$theta
                    wrccap<-shyp.L$thetacap
                    wrcnc<-shyp.L$thetanc

                    hcc<-log10(shyp.L$Kh)
                    hcccap<-log10(shyp.L$Kcap)
                    hccnc<-log10(shyp.L$Knc)
                    hcvap<-log10(shyp.L$Kvap)

                    par(mfrow=c(1,2),tcl=tcllen)
                    plot(retdata,ylim=c(.0,.50),xlim=c(0,6.8),ylab="",xlab="",
                        col="darkgrey",axes=FALSE,main="WaterRetentionCurve",cex=2)
                    lines(log10(h),wrccap,col="brown",lwd=2)
                    lines(log10(h),wrcnc,col="brown",lwd=2,lty=2)
                    lines(log10(h),wrc,col="darkblue",lwd=2)

                    legend("topright",c("observed","simulated"),pch=c(1,NA),
                        lty=c(NA,1),lwd=2,bty="n",cex=1.3,col=c("darkgrey","darkblue"))
                    axis(1,at=pow,labels=parse(text=paste('10^',(pow),sep="")),tcl=tcllen)
                    axis(2,tcl=tcllen)
                    axis(4,labels=NA)
                    axis(3,labels=NA)
                    mtext("pressurehead|h|[cm]",1,cex=1.2,line=2.8)
                    mtext("vol.watercontent[-]",2,cex=1.2,line=2.8)
                    box()

                    #  PLOT THE HYDRAULIC CONDUCTIVITY CURVE
                    plot(condata,ylim=c(-8,max(max(condata[,1]),max(hcc)))
                        ,xlim=c(0,6.8),ylab="",xlab="",col="darkgrey",
                        axes=FALSE,main="HydraulicConductivityCurve",cex=2)
                    lines(log10(h),hcccap,col="brown",lwd=2)
                    lines(log10(h),hccnc,col="brown",lwd=2,lty=2)
                    lines(log10(h),hcc,col="darkblue",lwd=2)
                    lines(log10(h),hcvap,col="darkblue",lwd=2)
                    legend("topright",c("observed","simulated"),pch=c(1,NA),
                        lty=c(NA,1),lwd=2,bty="n",cex=1.3,col=c("darkgrey","darkblue"))
                    axis(1,at=pow,labels=parse(text=paste('10^',(pow),sep="")),tcl=tcllen)
                    axis(2)
                    axis(4,labels=NA)
                    axis(3,labels=NA)
                    mtext("log10K[cm/d]",2,cex=1.2,line=2.8)
                    mtext("pressurehead|h|[cm]",1,cex=1.2,line=2.8)
                    box()
                    par(mfrow=c(1,1))
            }
      }
      phattrans.m <- out$out$optim$phattrans
    gof.L[[i]]<-gofFun(phattrans.m,shpmodel=shpmodel.v[i],retdata=retdata,condata=condata,
                    out.L[[i]]$settings$weight,parL$psel,ivap.query=NULL,hclip.query=FALSE)
  }
```

```
statstab3 <- cbind("th_rmse" = unlist(lapply(lapply(gof.L, `[[`, "th"), '[[', "rmse")),
                   "log10Kh_rmse" = unlist(lapply(lapply(gof.L, `[[`, "log10Kh"), '[[', "rmse"))
)
}

## End(Not run)
```

---

shypFun   *Wrapper Function for all Supported Soil Hydraulic Property Models*

Description

This function allows to select soil hydraulic property models.

Usage

```
shypFun(p, h, shpmodel = "01110", ivap.query = NULL)
```

Arguments

p	Vector of the model parameters, order is sensitve cf respective model documentation.
h	Pressure heads [cm] for which the corresponding retention and conductivity values are calculated.
shpmodel	character

01110	unimodel van Genuchten-Mualem model, with the contraint of m = 1-1/n (van Genuchten 1980)
01210	bimodel van Genuchten-Mualem model, with the contraint of m_i = 1-1/n_i (Durner 1994)
01310	trimodal van Genuchten-Mualem model, with the contraint of m_i = 1-1/n_i (Durner 1994)
02110	unimodel Kosugi 2 parametric-Mualem model (Kosugi 1996)
03110	unimodel Fredlund-Xing-Mualem model, with the contraint of m = 1-1/n (Fredlund and Xing 1994)

NULL	no isothermal vapour conductivity will be calculated with Kvap
	Model type for isothermal vapour conductivity, see Details of function KvapFun for model codes

ivap.query

Value

a list with calculated soil hydraulic properties at specified h:

theta	calculated volumetric moisture content
Se	calculated saturation

Scap	effective saturation (of the capillary part if FM is specified
cap	specific water capacity function (of the capillary part if FM is specified
psd	pore size distribution (of the capillary part if FM is specified
Kh	total hydraulic conductivity

if FM specified, additionally:

thetacap	calculated volumetric moisture content of the capillary part
thetanc	calculated volumetric moisture content of the non-capillary part
Snc	effective saturation of the non-capillary part
Kcap	hydraulic conductivity of the capillary
Knc	hydraulic conductivity of the non-capillary
Kvap	isothermal vapour conductivity
Krcap	relative hydraulic conductivity of the capillary
Krnc	relative hydraulic conductivity of the non-capillary

Note

the function is used to assign a new function variable with a function which calculates the soil hydraulic properties according to sepcified shpmodel and model specified by ivap.query

Author(s)

Tobias KD Weber , <tobias.weber@uni-hohenheim.de>

References

Durner W (1994). "Hydraulic conductivity estimation for soils with heterogeneous pore structure." *Water Resources Research*, **30**(2), 211–223. ISSN 0043-1397.

Fredlund DG, Xing A (1994). "Equations for the soil-water characteristic curve." *Canadian Geotechnical Journal*, **31**(4), 521–532. ISSN 00083674 (ISSN), doi: 10.1139/t94061. Kosugi K (1996). "Lognormal distribution model for unsaturated soil hydraulic properties." *Water Resources Research*, **32**(9), 2697–2703. ISSN 0043-1397. van Genuchten MT (1980). "Closed-form equation for predicting the hydraulic conductivity of unsaturated soils." *Soil Science Society of America Journal*, **44**(5), 892–898. ISSN 03615995 (ISSN). Weber TK, Durner W, Streck T, Diamantopoulos E (2019). "A modular framework for modelling unsaturated soil hydraulic properties over the full moisture range." *Water Resources Research*. ISSN 0043-1397, doi: 10.1029/2018WR024584.

Examples

```
data("shpdata1")
retdata <- shpdata1$LFH1$wrc[!is.na(shpdata1$LFH1$wrc[,1]),]
condata <- shpdata1$LFH1$hcc

# assign auxiliary variables
```

```
pF <- seq(-3, 6.8, length = 501)
h <- 10^pF

# assign list of parameters for the van Genuchten-Mualem model
parL <- list("p" = c("thr"= 0.05, "ths" = 0.45, "alf1" = 0.01, "n" = 2, "Ks" = 100, "tau" = .5),
             "psel" = c(1, 1, 0, 1, 1, 1),
             "plo" = c(0.001 , 0.2, 0.001, 1.1, 1, -2),
             "pup" = c(0.3, 0.95, 1, 10, 1e4, 10))

# calculate soil hydraulic property function values

shyp.L <- shypFun(parL$p, h, shpmodel = "01110", ivap.query = NULL)
wrc <- shyp.L$theta
hcc <- log10(shyp.L$Kh)

# # PLOT THE MEASURED WATER RETENTION CURVE
ticksatmin <- -2
tcllen <- 0.4
ticksat <- seq(ticksatmin,5,1)
pow <- ticksatmin:6

par(mfrow = c(1,2), tcl = tcllen)
plot(retdata, ylim = c(.0,.50), xlim = c(0, 6.8), ylab = "", xlab = "",
     col = "darkgrey", axes = FALSE, main = "Water Retention Curve", cex = 2)
lines(log10(h), wrc, col = "darkblue", lwd = 2)
legend("topright", c("observed", "simulated"),pch = c(1,NA),
       lty = c(NA,1), lwd = 2, bty = "n", cex = 1.3,col = c("darkgrey", "darkblue"))
axis(1, at = pow, labels = parse(text = paste('10^',(pow), sep = "")), tcl = tcllen)
axis(2, tcl = tcllen)
axis(4, labels = NA)
axis(3, labels = NA)
mtext("pressure head |h| [cm]", 1, cex = 1.2, line = 2.8)
mtext("vol. water content [ - ]", 2, cex = 1.2, line = 2.8)
box()

# PLOT THE MEASURED HYDRAULIC CONDUCTIVITY CURVE
plot(condata, ylim = c(-8,2), xlim = c(0, 6.8), ylab = "", xlab = "", col = "darkgrey",
     axes = FALSE, main = "Hydraulic Conductivity Curve", cex = 2)
lines(log10(h), hcc, col = "darkblue", lwd = 2)
legend("topright", c("observed", "simulated"), pch = c(1,NA),
       lty = c(NA,1), lwd = 2, bty = "n", cex = 1.3, col = c("darkgrey","darkblue"))
axis(1, at = pow, labels = parse(text = paste('10^',(pow), sep = "")), tcl = tcllen)
axis(2)
axis(4, labels = NA)
axis(3, labels = NA)
mtext("log10 K [cm/d]", 2, cex = 1.2, line = 2.8)
mtext("pressure head |h| [cm]",1 , cex = 1.2, line = 2.8)
box()
par(mfrow = c(1,1))

## Not run:
# HOW TO WRITE A MATER.IN FOR HYDRUS-1D
```

```
    mater_out <- cbind(shyp.L[['theta']], h, shyp.L[['Kh']], abs(shyp.L[['cap']]))

    materWriteFun <- function(mater_out.L, path = getwd(), sample) {

            # Function to write a Mater.in

            # ARGUMENTS

            # mater_outdata frame of 4 columns of calculated SHP values at h and length n.
            # 1. Column: THETA
            # 2. Column: H(negative pressure heads)
            # 3. Column: K
            # 4. Column: C(positive)
            # path character specifying the path where the MATER.IN should be saved
            # sample optional chr for sample name: NULL = no name given

            n <- dim(mater_out)[1]
            sink(file.path(path, paste(sample, "MATER.IN", sep = "")))
            cat(c("iCap", "\n", "1", "\n", "NTab", "\n", n, "\n"))
            cat(c("\t","THETA", "\t\t\t","H","\t\t\t","K","\t\t\t","C"))
            cat("\n")

            write.table(format(mater_out, justify = "right"),
                        row.names = FALSE, col.names = FALSE, quote = FALSE)
            sink()
     }

  ## End(Not run)
```

---

shypFun.01110	*van Genuchten-Mualem Soil Hydraulic Proptery Model*

Description

van Genuchten-Mualem functions for the retention and hydraulic conductivity curves (van Genuchten 1980).

Usage

```
shypFun.01110(p, h)
```

Arguments

p   vector of the 6 van Genuchten-Mualem model parameters, the order is sensitve and has to be given as:

thr	residual water water content [cm cm-3]
ths	saturated water water content [cm cm-3]
alf1	van Genuchten alpha [cm-3]
n1	van Genuchten n [-]

Ks   saturated conductivity [cm d-1]
tau   exponent of Se in the capillary conductivity model, sometimes denoted in the literature as l [-]

h   pressure heads [cm] for which the corresponding retention and conductivity values are calculated.

Details

The function solves analytically the spec. water capacity function and integral to the capillary bundle model. It can be extended by the Brunswick model to account for non-capillary storage and conductivity (Weber et al. 2019).

Value

returns a `list` with calculations at specified h:

theta   calculated volumetric moisture content

Se   calculated saturation

cap   specific water capacity function

psd   pore size distribution

Kh   Hydraulic conductivity values

Author(s)

Tobias KD Weber , `<tobias.weber@uni-hohenheim.de>`

References

van Genuchten MT (1980). "Closed-form equation for predicting the hydraulic conductivity of unsaturated soils." *Soil Science Society of America Journal*, **44**(5), 892–898. ISSN 03615995 (ISSN). Weber TK, Durner W, Streck T, Diamantopoulos E (2019). "A modular framework for modelling unsaturated soil hydraulic properties over the full moisture range." *Water Resources Research*. ISSN 0043-1397, doi: 10.1029/2018WR024584.

Examples

```
p      <- c(0.1, 0.4, 0.01, 2, 100, .5)
h      <- 10^seq(-2, 6.8, length = 197)
shyp.L <- shypFun.01110(p, h)
```

---

shypFun.01210   *van Genuchten-Mualem bimodal Soil Hydraulic Propterty Model*

Description

bimodal van Genuchten-Mualem functions (Durner model) for the retention and hydraulic conductivity curves (Durner 1994).

Usage

```
shypFun.01210(p, h)
```

Arguments

p   vector of the 9 van Genuchten-Mualem model parameters, order is sensitve and has to be given as:

thr	residual water water content [cm cm-3]
ths	saturated water water content [cm cm-3]
alf1	van Genuchten alpha [cm-3]
n1	van Genuchten n [-]
w1	fraction of the first modality [-], w2 is internally computed as w2 = 1-w1
alf2	van Genuchten alpha of the second modality[cm-3]
n2	van Genuchten n of the second modality [-]
Ks	saturated conductivity [cm d-1]
tau	exponent of Se in the capillary conductivity model, sometimes denoted in the literature as l [-]

h   pressure heads [cm] for which the corresponding retention and conductivity values are calculated.

Details

The function solves analytically the spec. water capacity function and integral to the capillary bundle model.

Value

returns a list with calculations at specified h:

theta   calculated volumetric moisture content

Se   calculated saturation

cap   specific water capacity function

psd   pore size distribution

Kh   Hydraulic conductivity values

Author(s)

Tobias KD Weber , `<tobias.weber@uni-hohenheim.de>`

References

Durner W (1994). "Hydraulic conductivity estimation for soils with heterogeneous pore structure." *Water Resources Research*, **30**(2), 211–223. ISSN 0043-1397.

Examples

```
p <- c("thr" = 0.1, "ths" = 0.4, alf1 = 0.5, "n1" = 3,
       "w1" = .6, "alf2" = 0.01, "n2" = 1.6,
       "Ks" = 100, "tau" = .5)
h <- 10^seq(-2, 6.8, length = 197)
shyp.L <- shypFun.01210(p, h)
```

---

shypFun.01310   *van Genuchten-Mualem trimodal Soil Hydraulic Propterty Model*

Description

trimodal van Genuchten-Mualem functions for the retention and hydraulic conductivity curves (Durner 1994).

Usage

```
shypFun.01310(p, h)
```

Arguments

p   vector of the 9 van Genuchten-Mualem model parameters, order is sensitve and has to be given as:

thr	residual water water content [cm cm-3]
ths	saturated water water content [cm cm-3]
alf1	van Genuchten alpha [cm-3]
n1	van Genuchten n [-]
w1	fraction of the first modality [-], w2 is internally computed as w2 = 1-w1
alf2	van Genuchten alpha of the second modality[cm-3]
n2	van Genuchten n of the second modality [-]
w2	fraction of the second modality [-], w3 is internally computed as w3 = 1-w1-w2, in resFun ensures w3 >=0
alf3	van Genuchten alpha of the third modality[cm-3]
n3	van Genuchten n of the third modality [-]
Ks	saturated conductivity [cm d-1]
tau	exponent of Se in the capillary conductivity model, sometimes denoted in the literature as l [-]

| h | pressure heads [cm] for which the corresponding retention and conductivity values are calculated. |

Details

The function solves analytically the spec. water capacity function and integral to the capillary bundle model.

Value

returns a `list` with calculations at specified h:

theta	calculated volumetric moisture content
Se	calculated saturation
cap	specific water capacity function
psd	pore size distribution
Kh	Hydraulic conductivity values

Author(s)

Tobias KD Weber , `<tobias.weber@uni-hohenheim.de>`

References

Durner W (1994). "Hydraulic conductivity estimation for soils with heterogeneous pore structure." *Water Resources Research*, **30**(2), 211–223. ISSN 0043-1397.

See Also

Weber TK, Durner W, Streck T, Diamantopoulos E (2019). "A modular framework for modelling unsaturated soil hydraulic properties over the full moisture range." *Water Resources Research*. ISSN 0043-1397, doi: 10.1029/2018WR024584.
Weber TKD, Iden SC, Durner W (2017). "Unsaturated hydraulic properties of Sphagnum moss and peat reveal trimodal pore-size distributions." *Water Resources Research*, **53**(1), 415–434. ISSN 0043-1397, doi: 10.1002/2016WR019707.
Weber TKD, Iden SC, Durner W (2017). "A pore-size classification for peat bogs derived from unsaturated hydraulic properties." *Hydrology and Earth System Sciences*, **21**(12), 6185–6200. ISSN 1607-7938, doi: 10.5194/hess2161852017.

Examples

```
p <- c("thr" = 0.1, "ths" = 0.4, alf1 = .5, "n1" = 3,
       "w1" = .5, "alf2" = 0.01, "n2" = 2,
       "w2" = .3, "alf3" = 0.01, "n3" = 1.6,
       "Ks" = 100, "tau" = .5)
h <- 10^seq(-2, 6.8, length = 197)
shyp.L <- shypFun.01310(p, h)
```

---

shypFun.02110   *Unimodal Kosugi-Mualem Model (2 Parameter Model)*

Description

Calculates the soil hydraulic property function values based on given pressure heads (Kosugi 1996).

Usage

```
shypFun.02110(p, h)
```

Arguments

p   vector of the 6 Kosugi-Mualem model parameters, order is sensitve and has to be given as:

thr	residual water water content [cm cm-3]
ths	saturated water water content [cm cm-3]
hm	air entry pressure head [cm]
sigma	width of pore size distribution [-]
Ks	saturated conductivity [cm d-1]
tau	exponent of Se in the capillary conductivity model, sometimes denoted in the literature as l [-]

h   pressure heads [cm] for which the corresponding retention and conductivity values are calculated.

Details

The function solves analytically the spec. water capacity function and integral to the capillary bundle model.

Value

returns a list with calculations at specified h:

theta	calculated volumetric moisture content
Se	calculated saturation
cap	specific water capacity function
psd	pore size distribution
Kh	Hydraulic conductivity values

References

Kosugi K (1996). "Lognormal distribution model for unsaturated soil hydraulic properties." *Water Resources Research*, **32**(9), 2697–2703. ISSN 0043-1397.

Examples

```
p        <- c(0.1, 0.4, 100, 2, 100, .5)
h        <- 10^seq(-2, 6.8, length = 197)
shyp.L <- shypFun.02110(p, h)
```

---

shypFun.03110   *Unimodal Fredlund-Xing - Mualem Model*

Description

Calculates the soil hydraulic property function values based on given pressure heads. The function calculates the base function of Fredlund and Xing (Fredlund and Xing 1994).

Usage

```
shypFun.03110(p, h)
```

Arguments

p   vector of the 6 van Genuchten-Mualem model parameters, order is sensitve and has to be given as:

thr	residual water water content [cm cm-3]
ths	saturated water water content [cm cm-3]
alf1	inverse of the air entry pressure head [cm]
n1	width of pore size distribution [-]
Ks	saturated conductivity [cm d-1]
tau	exponent of Se in the capillary conductivity model, sometimes denoted in the literature as l [-]

h   pressure heads [cm] for which the corresponding retention and conductivity values are calculated.

Details

The function numerically solves the spec. water capacity function and integral to Mualem's conductivity model.

Value

returns a list with calculations at specified h:

theta	calculated volumetric moisture content
Se	calculated saturation
cap	specific water capacity function
psd	pore size distribution
Kh	Hydraulic conductivity values

Author(s)

Tobias KD Weber , <tobias.weber@uni-hohenheim.de>

References

Fredlund DG, Xing A (1994). "Equations for the soil-water characteristic curve." *Canadian Geotechnical Journal*, **31**(4), 521–532. ISSN 00083674 (ISSN), doi: 10.1139/t94061.

Examples

```
p       <- c(0.1, 0.4, 0.01, 2, 100, .5)
h       <- 10^seq(-2, 6.8, length = 197)
shyp.L <- shypFun.03110(p, h)
```

---

shypFun.04110   *Unimodal Brooks-Corey Model*

Description

Calculates the soil hydraulic property function values based on given pressure heads (Kosugi 1996).

Usage

```
shypFun.04110(p, h)
```

Arguments

p   vector of the 6 Brooks-Corey model parameters, order is sensitve and has to be given as:

thr	residual water water content [cm cm-3]
ths	saturated water water content [cm cm-3]
alf	air entry pressure head [cm^-1]
bet	effective model parameter [-]
Ks	saturated conductivity [cm d-1]
tau	exponent of Se in the capillary conductivity model, sometimes denoted in the literature as l [-]

h   pressure heads [cm] for which the corresponding retention and conductivity values are calculated.

Details

The function solves analytically the spec. water capacity function and integral to the capillary bundle model.

Value

returns a `list` with calculations at specified h:

theta	calculated volumetric moisture content
Se	calculated saturation
cap	returns NA; not supported
psd	returns NA; not supported
Kh	Hydraulic conductivity values

Note

The Muealm integral is solved numerically

References

Brooks RH, Corey AT (1964). *Hydraulic Properties of Porous Media*. Hydrology Papers No 3, Colorado State University, Fort Collings, Colorado.

Examples

```
p      <- c(0.1, 0.4, .01, .3, 100, .5)
h      <- 10^seq(-2, 6.8, length = 197)
shyp.L <- shypFun.04110(p, h)
```

---

sncFun	*Non-capillary Saturation Function to Extend Other Functions*

Description

The general purpose method to calculate numerically the effective non-capillary saturation is directly obtained from any arbitrary expression for the rescaled capillary saturation function as described by (Weber et al. 2019). Examples of capillary saturation functions are the well known (van Genuchten 1980), (Kosugi 1996), (Fredlund and Xing 1994) functions.

Usage

```
sncFun(h, scap)
```

Arguments

h	A vector of n pressure head values for which `scap` was calculated
scap	vector of n monotonically decreasing capillary saturation function values calculated by `shypFun`, rescaled between 0 and 1.

Value

A vector of n elements with calculated saturation content of the non-capillary part.

Author(s)

Tobias KD Weber , `<tobias.weber@uni-hohenheim.de>`

References

van Genuchten MT (1980). "Closed-form equation for predicting the hydraulic conductivity of un-saturated soils." *Soil Science Society of America Journal*, **44**(5), 892–898. ISSN 03615995 (ISSN).

Kosugi K (1996). "Lognormal distribution model for unsaturated soil hydraulic properties." *Water Resources Research*, **32**(9), 2697–2703. ISSN 0043-1397.

Fredlund DG, Xing A (1994). "Equations for the soil-water characteristic curve." *Canadian Geotechnical Journal*, **31**(4), 521–532. ISSN 00083674 (ISSN), doi: 10.1139/t94061.

Examples

```
# set variables
p <- c(0.1, 0.4, 0.01, 2, 100, .5)
h <- 10^seq(-2, 6.8, length = 197)

# Calculate the capillary and non-capillary saturation function.
Se <- shypFun(p, h, shpmodel = "01110")$Se
Snc <- sncFun(Se)
```

---

sncFun.01110 *Unimodal van Genuchten Non-Capillary Saturation Model*

Description

Analytical implementation of the non-capillary saturation function (van Genuchten 1980).

Usage

```
sncFun.01110(p_snc, h)
```

Arguments

p_snc   vector of the 2 van Genuchten Mualem model and h0, the order is sensitve and has to be given as:

alf1	van Genuchten alpha [cm-3]
n1	van Genuchten n [-]
h0	pressure head representing oven dryness given in pF, i.e. log[10](|pressure head| [cm])

h   pressure heads [cm] for which the corresponding retention and conductivity values are calculated.

Details

The function is Eq. Table 1-C1 in insertRefStreck.2020spsh using eq 21 which can be used to accelerate the convergence of the sum. The analytical solution presented in `sncFun.01110` only requires the Brooks-Corey model parameters

Value

returns a `list` with calculations at specified h:

`snc`   non-capillary saturation

References

van Genuchten MT (1980). "Closed-form equation for predicting the hydraulic conductivity of un-saturated soils." *Soil Science Society of America Journal*, **44**(5), 892–898. ISSN 03615995 (ISSN). Weber TK, Durner W, Streck T, Diamantopoulos E (2019). "A modular framework for modelling unsaturated soil hydraulic properties over the full moisture range." *Water Resources Research*. ISSN 0043-1397, doi: 10.1029/2018WR024584. Streck T, Weber TKD (2020). "Analytical expressions for noncapillary soil water retention based on popular capillary retention models, accepted." *Vadose Zone Journal*. ISSN 15391663 (ISSN).

Examples

```
p        <- c(0.1, 0.4, .01, 2, 100, .5)
# add h0
p_snc   <- c(p[3:4], 6.8)
h        <- 10^seq(-2, 6.8, length = 197)
Se       <- shypFun.01110(p, h)$Se
snc      <- sncFun.01110(p_snc, h)
```

---

sncFun.02110   *Unimodal Kosugi Non-Capillary Saturation Model*

Description

Analytical implementation of the non-capillary saturation function for the (Kosugi 1996).

Usage

```
sncFun.02110(p_snc, h)
```

Arguments

`p_snc`   vector of the 2 Kosugi saturation model parameters, and h0 sensitve and has to
                         be given as:

`hm`	air entry pressure head [cm]
`sigma`	width of pore size distribution [-]
`h0`	pressure head representing oven dryness given in pF, i.e. log[10](|pressure head| [cm])

h	pressure heads [cm] for which the corresponding retention and conductivity values are calculated.

Details

The function is Eq. Table 1-B in insertRefStreck.2020spsh The analytical solution presented in `sncFun.02110` only requires the Kosugi specific model parameters and `h0`

Value

returns a `list` with calculations at specified h:

snc	non-capillary saturation

References

Kosugi K (1996). "Lognormal distribution model for unsaturated soil hydraulic properties." *Water Resources Research*, **32**(9), 2697–2703. ISSN 0043-1397. Streck T, Weber TKD (2020). "Analytical expressions for noncapillary soil water retention based on popular capillary retention models, accepted." *Vadose Zone Journal*. ISSN 15391663 (ISSN).

Examples

```
p       <- c(0.1, 0.4, 100, 2, 100, .5)
# add h0
p_snc   <- c(p[3:4], 6.8)
h       <- 10^seq(-2, 6.8, length = 197)
Se      <- shypFun.02110(p, h)$Se
snc     <- sncFun.02110(p_snc, h)
```

---

sncFun.04110	*Unimodal Brooks-Corey Non-Capillary Saturation Model*

Description

Analytical implementation of the non-capillary saturation function from the Brooks-Corey function (Brooks and Corey 1964).

Usage

```
sncFun.04110(p_snc, h)
```

Arguments

p_snc	vector of the 2 Brooks-Corey model parameters, order is sensitve and has to be given as:

alf	air entry pressure head [cm^-1]
bet	effective model parameter [-]
h0	pressure head representing oven dryness given in pF, i.e. log[10](|pressure head| [cm])

h pressure heads [cm] for which the corresponding retention and conductivity values are calculated.

Details

The function is Eq. Table 1-A in insertRefStreck.2020spsh The analytical solution presented in sncFun.04110 only requires the Brooks-Corey model specific parameters and h0.

Value

returns a `list` with calculations at specified h:

snc non-capillary saturation

References

Brooks RH, Corey AT (1964). *Hydraulic Properties of Porous Media*. Hydrology Papers No 3, Colorado State University, Fort Collings, Colorado. Streck T, Weber TKD (2020). "Analytical expressions for noncapillary soil water retention based on popular capillary retention models, accepted." *Vadose Zone Journal*. ISSN 15391663 (ISSN).

Examples

```
p     <- c(0.1, 0.4, .02, 2, 100, .5)
p_snc <- c(p[3:4], 6.8)
h     <- 10^seq(-2, 6.8, length = 197)
Se    <- shypFun.04110(p, h)$Se
snc   <- sncFun.04110(p_snc, h)
```

---

transBoundFun *Creates Parameter Transformation and Backtransformation Rules for the Estimation Procedure*

Description

Creates Parameter Transformation and Backtransformation Rules for the Estimation Procedure

Usage

```
transBoundFun(parL, shpmodel, weightmethod)
```

Arguments

parL a list with 4 numeric vectors specifying:

p   Vector of model parameters, has to coincide with the chosen soil hydraulic property model. If weightmethod == est1
psel vector identifying which parameters are to be estimatedmodel parameters, has to coincide with the chosen soil hydraul
plo   vector of lower bounds (non-transformed parameter boundaries)
pup   vector of upper bounds (non-transformed parameters boundaries)

shpmodel	A string specifying the selected soil hydraulic property model.
weightmethod	A string specifying the selected weigthing method, if weightmethod == "est1" is TRUE, then parL is modified to account for nuisance parameters).

Details

This function is intended for the function shypEstFun so that lists with set rules for the transformation and back-transformation of the soil hydraulic model parameters are enabled. In general, the following rules apply log10 transformation for the model parameters α_i, n_i-1, K_s, K_sc, K_snc.

The function is meant for internal use in shypEstFun.

Value

a list of two lists. One of the sub-lists is parL but with transformed parameters, and the second, transL with model specific transformation and back-transformation rules.

Author(s)

Tobias KD Weber , <tobias.weber@uni-hohenheim.de>

Examples

```
parL <- list("p" = c("thr"= 0.05, "ths" = 0.45, "alf1" = 0.01, "n" = 2, "Ks" = 100, "tau" = .5),
             "psel" = c(1, 1, 0, 1, 1, 1),
             "plo" = c(0.001 , 0.2, 0.001, 1.1, 1, -2),
             "pup" = c(0.3, 0.95, 1, 10, 1e4, 10))

# transformation and back-transformation of parameter vectors
for(k in c("p", "plo", "pup")){
    for (j in c("none")){
        parL.trans <- transBoundFun(parL, shpmodel = "01110", weightmethod = j)

        p_trans <- transFun(parL[[k]], parL.trans$transL$ptrans)
        p_retrans <- transFun(p_trans, parL.trans$transL$pretrans)

        stopifnot(sum(p_retrans != parL[[k]])==0)
    }
}
```

---

transFun   *Parameter Transformation and Back-transformation*

Description

Enables the transformation and backtransformation of parameters. This is widely considered advantageous during parameter estimation as the parameter space in the transformed is well-behaved, e.g. with normally distributed posteriors.

Usage

```
transFun(par.vec, trans.L)
```

Arguments

par.vec   Vector of n model parameters.

trans.L   list of n transformation/backtransformation operators, transformation and back-
               transformatio rules have to be antonyms and position in vector has to coincide
               with that in par.vec.

Details

Transformation rules are:

$$log10\alpha_i, log10n_i - 1, log10Ks, log10\omega, log10Ksc, and log10Ksnc$$

.

Value

Returns transformed parameters as specificef by trans.L.

Note

The function is used to transform the parameter space and enabling optimisation or MCMC sam-
pling to be more efficient.

Author(s)

Tobias KD Weber , <tobias.weber@uni-hohenheim.de>

Examples

```
# van Genuchten-Mualem Model parameters
parL <- list("p" = c("thr"= 0.05, "ths" = 0.45, "alf1" = 0.01, "n" = 2, "Ks" = 100, "tau" = .5),
             "psel" = c(1, 1, 0, 1, 1, 1),
             "plo" = c(0.001 , 0.2, 0.001, 1.1, 1, -2),
             "pup" = c(0.3, 0.95, 1, 10, 1e4, 10)
)
# Two lists, one with function to transform, the other to back-transform model parameters
ptransfit <- c(function(x)x, function(x)x,log10,function(x)log10(x-1),log10, function(x)x)
pretransfit <- c(function(x)x, function(x)x,function(x)10^x,
                 function(x)10^x+1,function(x)10^x,function(x)x)
# Transform
p_trans <- transFun(parL$p, ptransfit)
```

---

weightFun   *Specification of Weights for the Data Groups Retention Data and Conductivity Data.*

Description

Weights can be fixed to suggested standards, fixed by the user, or estimated as additional nuisance parameters.

Usage

```
weightFun(weightmethod = "fix1", retdata, condata, parL = NA)
```

Arguments

weightmethod	Character specifying the method of selecting model weights, cd Details.
retdata	A dataframe or matrix with 2 columns. The first with pressure head values in [cm] and the second with volumetric water contents in [cm cm-3].
condata	A dataframe or matrix with 2 columns. The first with pressure head values in [cm] and the second with hydraulic conductivity values log10[cm d-1].
parL	Defaults to NA has to be provided if weightmethod == "est1" . See Details of ([shypEstFun](#) for explanation of parL).

Details

Character specifying weightmethod

user	user defined weights
none	no weights are considered, i.e. no measurement error assumed
range	rescaling (normalization of observations to the intervall [0,1]
fix1	fixed scalar weight for THETA is 0.05^2 and weight for log10K is 1
fix2	vector with the length of number of observations as given in retdata and condata are given, fixed to weight for THI
est1	Two scalar model weights (sigma^-2) are treated as free parameters to be estimated by inversion, one for THETA and c

If weightmethod is set to est1 and parL is given as an extra argument, the function returns a list wich is concatenated to the parL used in shypEstFun providing extra information on the nuisance parameters. Alternatively, parL can be passed as an argument to shypEstFun directly, accounting for the two additional nuisance parameters at the end of the respective vectors.

Value

The function returns a list of weights as specified through weightmethod.

Examples

```
# Example 1 | fixed weights
```

```
weight.fix.L <- weightFun("fix1")

# Example 2 | range of measure data
data(shpdata1)

wrc <- shpdata1$TS1$wrc
hcc <- shpdata1$TS1$hcc
# Remove NAs
hcc <- shpdata1$TS1$hcc[!is.na(shpdata1$TS1$hcc[,1] ),]
weight.fix.L <- weightFun("range", wrc, hcc)
```

Index