

Supplement of Hydrol. Earth Syst. Sci., 19, 2101–2117, 2015
<http://www.hydrol-earth-syst-sci.net/19/2101/2015/>
doi:10.5194/hess-19-2101-2015-supplement
© Author(s) 2015. CC Attribution 3.0 License.



Supplement of

Virtual laboratories: new opportunities for collaborative water science

S. Ceola et al.

Correspondence to: S. Ceola (serena.ceola@unibo.it)

1 The TUW model

The TUWmodel is a lumped conceptual rainfall-runoff model, following the structure of the HBV model. The model consists of 3 routines: a snow routine, a soil moisture routine and a flow routing routine (for more details see Parajka *et al.* [2007]). The 15 model parameters, whose suggested ranges provided in parenthesis are valid when the model is run at the daily time step, as in this experiments in this paper, are:

1. SCF: snow correction factor (0.9-1.5);
2. DDF: degree day factor ($0.0-5.0 \text{ mm } ^\circ \text{C}^{-1} \text{ day}^{-1}$);
3. Tr: threshold temperature above which precipitation is rain ($1.0-3.0 \text{ } ^\circ \text{C}$);
4. Ts: threshold temperature below which precipitation is snow ($-3.0-1.0 \text{ } ^\circ \text{C}$);
5. Tm: threshold temperature above which snow melt starts ($-2.0-2.0 \text{ } ^\circ \text{C}$);
6. LPrat: parameter related to the limit for potential evaporation (0.0-1.0);
7. FC: field capacity, i.e., max soil moisture storage (0-600 mm);
8. BETA: the non linear parameter for runoff production (0.0-20.0);
9. k0: storage coefficient for very fast response (0.0-2.0 day);
10. k1: storage coefficient for fast response (2.0-30.0 day);
11. k2: storage coefficient for slow response (30.0-250.0 day);
12. lsuz: threshold storage state, i.e., the very fast response start if exceeded (1.0-100.0 mm);
13. cperc: constant percolation rate ($0.0-8.0 \text{ mm day}^{-1}$);
14. bmax: maximum base at low flows (0.0-30.0 day);
15. croute: free scaling parameter ($0.0-50.0 \text{ day}^2 \text{ mm}^{-1}$).

The model is available for R at <http://cran.r-project.org/web/packages/TUWmodel/>.

2 Protocol 2 settings: rationale from each research group

2.1 SMHI: Automatic calibration evaluation and optional re-calibration

We chose to start the evaluation with a set of additional diagnostic plots of the calibrated results for each catchment, besides the ones available from the

Protocol 1. The additional plots consist of time series plots over the automatic calibration period, i.e., the first half of each observation data set. Variables covered in these plots are:

- precipitation with snow/rain distribution
- air temperature
- potential and actual evapotranspiration
- snow water equivalent
- observed and simulated discharge
- runoff contributions from storage components
- accumulated volume error
- relative soil moisture

Furthermore, we compiled annual water balance bar charts with water balance components precipitation, stacked simulated discharge and actual evapotranspiration, and additionally a bar with measured discharge. In order to investigate parameter identifiability, we re-run the model in each catchment using 10000 parameter sets, sampled from the automatic calibration parameter space using a latin hypercube approach and evaluated against performance criteria KGE, R^2 , MSE, VE, and NSE. We evaluated the result plots and parameter sets in round-table discussions within the SMHI partner group, and made decisions on either acceptance of the original calibration or re-calibration using alternative parametrisation. We established the following options for evaluation:

- acceptance of the original calibration, with comments on performance details;
- automatic re-calibration on a limited sub-set of the original data (“hydrological expertise”);
- descriptive evaluation of model errors without changing parametrisation, where model structural or data errors were assumed.

We opted to keep the MSE as evaluation criterion for automatic re-calibration. For re-calibrations on limited sub-sets, we accepted performance penalties if we agreed that the conceptual behaviour got better in the process. In particular, we noticed behavioural model failure where the water balance in the input data was not closed and the optimisation forced unrealistic behaviour to compensate for this. We evaluated this behaviour mainly discussing time series plots in the group.

2.2 UNIBO: Data quality check

An adequate parameter estimate may be hampered or biased by data errors in the calibration period. The measurements and estimates of the hydro-meteorological time-series used in the parametrisation procedure is often affected by considerable uncertainty and errors. Such errors may be randomly distributed in time but a considerable part of them characterise long, continuous periods: typical examples are when one or more rain gages do not work for a period (so that the rainfall network density becomes insufficient), or when the rating curve is not updated following an important flood event (so that it does not account for changes in the roughness, the slope, the river section geometry). Under these circumstances, there may be weeks or months of unreliable data and these periods should not be kept into account in the calibration phase. Unfortunately, such uncertainties are rarely explicitly signalled in the data sources and in order to identify these periods, a simplified approach was applied, assuming that the simulation errors obtained in the first protocol experiment are mainly due to this kind of errors (rather than to model or parameters inadequacy). By using the parameter set that allowed the best results on the calibration set in the first protocol experiment, for each study watershed we computed the MSE for each separate year of the considered period. Hypothesising that the years where the model performs worst are those that are characterised by a higher fraction of data errors, around one quarter of the years corresponding to the highest MSE were identified as unreliable periods. These years were then excluded from the computation of the objective function in Protocol 2.

2.3 TUW: Alternative optimisation algorithm and objective function

The TUW motivation for selecting the calibration strategy in Protocol 2 was to repeat and compare the results with findings of Merz *et al.* [2011]. Merz *et al.* [2011] aimed to analyse the temporal stability of the TUWmodel parameters and to assess the implications for climate impact evaluations. The main objective was to introduce a penalty in the calibration objective function, which will account for the difference between calibrated model parameter values and the pre-define model parameter distribution. It is expected that such approach reduces parameter uncertainty and the values of model parameters calibrated in different periods are more similar. In this strategy, 4 model parameters (Tr, Ts, bmax and croute) were fixed and 11 model parameters (i.e., SCF, DDF, Tm, LPrat, FC, BETA, k0, k1, k2, lsuz and cperc) were calibrated by using the Shuffled Complex Evolution (SCE) optimisation algorithm. More details about the a-priori definition of model parameter distributions is given in Merz *et al.* [2011].

2.4 TUD: Alternative optimisation algorithm and objective function

The dynamically dimensioned search (DDS) algorithm is a relatively fast converging algorithm that reduces the dimensionality by probabilistically adjusting the set of parameters that need to be modified in the optimisation run (Tolson & Shoemaker [2007]). This algorithm was chosen in order to look at differences in modelling results between a relatively fast search algorithm and the algorithms applied by the other research partners. The calibration was repeated 10 times (just as in Protocol 1), with 1000 iterations. The Kling-Gupta efficiency (KGE) is a more balanced objective function compared with the Nash-Sutcliffe efficiency. It was shown by Gupta *et al.* [2009] that the Nash-Sutcliffe efficiency can be rewritten in a combination of bias, correlation coefficient and relative variation and it can be noted that the Nash-Sutcliffe efficiency is a not-equally weighed combination of the before mentioned statistical metrics. The relative variability is for example always underestimated by optimising for NSE. The KGE provides a more equally weighed combination of these same metrics, what would lead to improvements for especially bias and variability in model outcomes. After a very basic visual inspection for unexplained peaks in the hydrograph, the calibration period was kept the total first period, except for the Tanaro (ITA) catchment, where we started from 01/09/2001 to avoid the use of no data values in the calibration period. The warming period was kept at one year for all the catchments.

2.5 BRISTOL: Data quality check

Runoff coefficients were calculated on an event basis for the calibration time-series of each catchment. Those events which exceeded a runoff-ratio of 1 were deemed unreliable. To calculate runoff coefficients, the time-series for precipitation and discharge were divided into discrete events. Precipitation was converted into an effective precipitation to account for snow fall and subsequent melt using a simple snow routine based on that applied in the TUWmodel. A threshold of zero degrees was used to represent both the snowfall/rainfall threshold and the melt threshold, and the Degree Day Factor (DDF) used to control the rate of snowmelt. Accurate event discharge calculation requires the extrapolation of the hydrograph for each event to quantify the volume of discharge that would occur if no further rainfall had occurred. In order to derive this discharge, an approach similar to Beven *et al.* [2011] was applied, where an approximation of the master recession curve was derived by combining individual shorter recession periods (segments) together. Flow recession segments were extracted from the discharge time-series that were at least 3 days in length. For the three day period of each recession curve, if the rainfall over that period exceeded a small threshold set for each catchment, then the segment was discarded. In order to combine the segments into a single master recession curve, the gradient at each point on each segment was calculated and all gradients combined and plotted as a function of the discharge at the point at which the gradient

was calculated. To the range of discharges for which the relationship between discharge and gradient was deemed reliable, a power function was fitted. For each event, this function was then used to extrapolate the flow recession curves to derive the total event discharge. Runoff coefficients were then calculated by dividing event runoff by event precipitation. The method for deriving both the event precipitation and the event discharge contain uncertainty. To account for this issue, a conservative approach to event exclusion was taken whereby runoff coefficients for each event were calculated over the whole calibration by varying the DDF over the range of the priors (1-5), and also by varying the parameters of the gradient-discharge function in order to represent the range of scatter in the relationship, and its effect on the derived recession curve, and in turn, the calculated event discharge. Only those events which exceeded a runoff coefficient of 1 on all realisations of the calculation of the runoff coefficient were excluded from the calibration.

References

- Beven, K., Smith, P. J., & Wood, A. 2011. On the colour and spin of epistemic error (and what we might do about it). *Hydrology and Earth System Sciences*, **15**(10), 3123–3133.
- Gupta, H. V., Kling, H., Yilmaz, K. K., & Martinez, G. F. 2009. Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. *Journal of Hydrology*, **377**(1-2), 80–91.
- Merz, R., Parajka, J., & Blöschl, G. 2011. Time stability of catchment model parameters: Implications for climate impact analyses. *Water Resources Research*, **47**(2).
- Parajka, J., Merz, R., & Blöschl, G. 2007. Uncertainty and multiple objective calibration in regional water balance modelling: case study in 320 Austrian catchments. *Hydrological Processes*, **21**(4), 435–446.
- Tolson, B. A., & Shoemaker, C. A. 2007. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. *Water Resources Research*, **43**(1).

```

# -----
#                                     #
# exercise in SWITCH-ON                #
#                                     #
# Baratti                             #
#                                     #
#                                     #
# -----
#                                     #
# This script is the tool for the collaborative experiment devised in WP3 of
# SWITCH-ON, that should focus on comparative application of
# a rainfall-runoff model on several catchments.

# PROTOCOL 1 as suggested by Alberto Montanari

# In the first protocol the same calibration-validation periods and same
# setting for calibrating the models are used by all partners.
# We expect that all partners independently converge to the same results.
# However, operational uncertainty may arise and it will be
# of interest to assess how it behaves in a context where it should be minimal,
# and how it evolves in a context where there is more
# flexibility. As we know, operational uncertainty is deemed to be an important
# issue in real-time hydrological applications.
#
# Hereafter the protocol (on which this script is designed):
#
# - Divide the observation period in 2 subsequent pieces of equal length. First
#   calibrate on the first period and validate on the
#   second and then revert the calibration and observation period, therefore
#   obtaining two calibration experiments.
#
# - Adopt a warm-up period of one year for both calibration and validation.
#
# - Use the same setting for the optimisation algorithm (here the Differential
#   Evolution optimisation algorithm is used) and run 10
#   calibration experiments with different random seeds. Keep the best
#   calibration run. Use the mean square error as objective function.
#
# - In validation compute the Nash efficiency, bias, mean absolute error and
#   mean log error. Also draw graphs of observed versus
#   simulated data (scatterplots) and observed versus simulated flow duration
#   curves.

# ----- #

# INTRODUCTION AND REQUIRED VARIABLES

# You should have created an ASCII file for each of your catchments, as the
# example file 'at4.asc', which is composed by a header
# and by 5 columns separated by one space indicating, respectively, day, month,
# year, catchment precipitation (mm/d), mean catchment
# temperature (deg C) and observed streamflow (m3/s).
# The file looks like the following:
#   # at4.asc
#   # Code: 204917
#   # Station: Furtmuehle
#   # River: Grosse Muehl
#   # Area: 253.3 km2
#   # Mean elevation: 723.4 m a.s.l.
#   # x: ???; y: ???
#   # Columns: day, month, year, P (mm/d), T (deg C), Q (m3/s)
#   # 1 1 1976 25.12777 0.87445 2.74
#   # 2 1 1976 13.28428 2.76742 14.4
#   # 3 1 1976 14.21458 3.01872 23.5
#   # 4 1 1976 13.38546 -1.46329 12.1
#   # 5 1 1976 15.95901 -3.57361 6.87
#   # ...
#   # ...

```

```

#
#      ...
#      27 12 2010 0.28481 -8.57164 3.73
#      28 12 2010 0.00675 -5.83401 3.66
#      29 12 2010 0 -9.7422 3.59
#      30 12 2010 0 -6.5834 3.55
#      31 12 2010 0.09979 -5.65648 3.52

# For the script to work the following variables should be defined: data, area,
SI.
# 'data' contains the table with 5 columns: day, month, year, catchment
precipitation (mm/d), mean catchment temperature (deg C) and
#      observed streamflow (m3/s);
# 'area' contains the catchment area;
# 'SI' is a vector of 12 values of the SI ratio. As in Parajka et al. (2003),
the Blaney-Criddle method modified by Schroedter (1985)
#      is used to calculate the potential evapotranspiration:
#      EP = -1.55 + 0.96 * (8.128 + 0.457 * T) * SI
#      with the constrain that EP >= 0. The values for the SI ratio are
calculated using ????? for all catchments and are reported in
#      the file ?????.

# An example of how to define the variables (data, area and SI) in R follows:
# > data <- read.table("at4.asc")
# or, eventually adding the path,
# > data <- read.table("/home/alviglio/00TEMP/R_work/SWITCH-ON_work/at4.asc")
# > area <- 253.3 # km2
# > SI <- c(0.19, 0.23, 0.27, 0.31, 0.35, 0.37, 0.36, 0.33, 0.29, 0.25, 0.20,
0.18)
# which is the SI ratio (one x month from JAN to DEC).
# Just for the output files, also define this character variable:
# > datafile <- "at4.asc"

# In order for the script to work, the following libraries should be installed:
# > install.packages("zoo") # to work with time series
# > install.packages("TUWmodel") # which is the rainfall-runoff model
(Parajka et al., 2007)
# > install.packages("DEoptim") # for evolutionary global optimization via
the Differential Evolution algorithm

# ----- #

# SCRIPT

# The following code loads libraries
library(zoo)
library(TUWmodel)
library(DEoptim)

# The following code creates timeseries
days <- as.Date(strptime(paste(data[,1], data[,2], data[,3]), format="%d %m
%Y"))
P <- zoo(data[,4], order.by=days) # daily catchment precipitation (mm/d)
T <- zoo(data[,5], order.by=days) # mean daily catchment temperature (deg C)
Q <- zoo(data[,6], order.by=days) # mean daily observed streamflow (m3/s)

# The following code calculates the potential evapotranspiration with the
Blaney-Criddle method modified by Schroedter (1985)
repeatedSI <- SI[as.numeric(format(index(P), '%m'))]
EP <- -1.55 + 0.96*(8.128 + 0.457*T) * repeatedSI
EP[EP < 0] <- 0 # daily potential evapotranspiration
(mm/d)

# The following code defines the two subsequent pieces of the time series of
equal length
years <- unique(as.numeric(format(days, "%Y")))
nyr <- length(years)

p1 <- window(P, end=as.Date(strptime(paste("31 12", years[floor(nyr/2)]),
format="%d %m %Y"))

```



```

T1 <- window(T, end=as.Date(strptime(paste("31 12", years[floor(nyr/2)]),
format="%d %m %Y")))
EP1 <- window(EP, end=as.Date(strptime(paste("31 12", years[floor(nyr/2)]),
format="%d %m %Y")))
Q1 <- window(Q, end=as.Date(strptime(paste("31 12", years[floor(nyr/2)]),
format="%d %m %Y")))

P2 <- window(P, start=as.Date(strptime(paste("1 1", years[nyr - floor(nyr/2) +
1]), format="%d %m %Y")))
T2 <- window(T, start=as.Date(strptime(paste("1 1", years[nyr - floor(nyr/2) +
1]), format="%d %m %Y")))
EP2 <- window(EP, start=as.Date(strptime(paste("1 1", years[nyr - floor(nyr/2)
+ 1]), format="%d %m %Y")))
Q2 <- window(Q, start=as.Date(strptime(paste("1 1", years[nyr - floor(nyr/2) +
1]), format="%d %m %Y")))

# The following function is the objective function that we want to minimise,
i.e., the mean square error. It embeds the function
# 'TUWmodel', which is a lumped conceptual rainfall-runoff model, following the
structure of the HBV model. The model runs on a daily
# time step and consists of a snow routine, a soil moisture routine and a flow
routing routine (Parajka et al., 2007).
# For more details use the R help:
# > help(TUWmodel)
MSE <- function (param, precip, temp, potevap, runoff, area) {
  simu <- TUWmodel(prec=as.numeric(precip), airt=as.numeric(temp),
ep=as.numeric(potevap), area=area, param)$q
  simu[is.nan(simu)] <- -999
  simu <- simu[-c(1:365)] # remove the warming period (1 year)
  obse <- runoff[-c(1:365)] # remove the warming period (1 year)
  mobs <- mean(obse, na.rm=TRUE)
  mean((simu - obse)^2, na.rm=TRUE) # mean square error
}

# The following code optimise the model parameters with the Differential
Evolution optimization algorithm.
# For more details use the R help:
# > help(DEoptim)
# ATTENTION: THIS TAKES TIME!!
calibrate_period1 <- DEoptim(fn=MSE, lower=c(0.9, 0.0, 1.0, -3.0, -2.0, 0.0,
0.0, 0.0, 0.0, 2.0, 30.0, 1.0, 0.0, 0.0, 0.0),
upper=c(1.5, 5.0, 3.0, 1.0, 2.0, 1.0,
600.0, 20.0, 2.0, 30.0, 250.0, 100.0, 8.0, 30.0, 50.0),
control=DEoptim.control(NP=NA, itermax=600,
reltol=1e-4, steptol=50, trace=10, parallelType=0),
precip=P1, temp=T1, potevap=EP1, runoff=Q1,
area=area)

calibrate_period2 <- DEoptim(fn=MSE, lower=c(0.9, 0.0, 1.0, -3.0, -2.0, 0.0,
0.0, 0.0, 0.0, 2.0, 30.0, 1.0, 0.0, 0.0),
upper=c(1.5, 5.0, 3.0, 1.0, 2.0, 1.0,
600.0, 20.0, 2.0, 30.0, 250.0, 100.0, 8.0, 30.0, 50.0),
control=DEoptim.control(NP=NA, itermax=600,
reltol=1e-4, steptol=50, trace=10, parallelType=0),
precip=P2, temp=T2, potevap=EP2, runoff=Q2,
area=area)
# Alberto Montanari suggests to run 10 calibration experiments with different
random seeds and keep the best calibration run.

# The following code uses the rainfall-runoff model with the optimised
parameters 'calibrate_period$optim$bestmem' and simulates on the
# calibration and validation periods
simulation1_cal1 <- TUWmodel(prec=as.numeric(P1), airt=as.numeric(T1),
ep=as.numeric(EP1), area=area,
param=calibrate_period1$optim$bestmem)
simulation2_cal2 <- TUWmodel(prec=as.numeric(P2), airt=as.numeric(T2),
ep=as.numeric(EP2), area=area,
param=calibrate_period2$optim$bestmem)
simulation2_val1 <- TUWmodel(prec=as.numeric(P1), airt=as.numeric(T1),
ep=as.numeric(EP1), area=area,

```

```

simulation1_val2 <-      param=calibrate_period2$optim$bestmem)
TUWmodel(prec=as.numeric(P2),      airt=as.numeric(T2),
ep=as.numeric(EP2), area=area,
      param=calibrate_period1$optim$bestmem)

# The following code writes a file 'param' with the optimised parameters
tt <- format(Sys.time(), "%Y%m%d%H%M")
output <- data.frame(simulation1_cal1$par, simulation2_cal2$par)
names(output) <- c("cal1", "cal2")
write.table(output, file=paste("param", tt, "_", datafile, sep=""))
# which produces the attached file 'param201402121216_at4.asc'

# The following function calculates efficiency measures
EMS <- function (sim, obs, warmup=365) {
# obs = observed runoff in mm/d (class numeric)
# sim = simulated runoff in mm/d (class numeric)
# warmup = warm-up period in d
simu <- sim[-c(1:warmup)]
obse <- obs[-c(1:warmup)]

# RMSE = root mean square error (mm/d)
RMSE <- sqrt(mean((simu - obse)^2, na.rm=TRUE))
# NE = Nash efficiency ()
mobse <- mean(obse, na.rm=TRUE)
NE <- 1 - sum((simu - obse)^2, na.rm=TRUE)/sum((obse - mobse)^2, na.rm=TRUE)
# lNE = log Nash efficiency ()
mlobse <- mean(log(obse), na.rm=TRUE)
lNE <- 1 - sum((log(simu) - log(obse))^2, na.rm=TRUE)/sum((log(obse) -
mlobse)^2, na.rm=TRUE)
# B = bias (mm/d)
B <- mean(simu - obse, na.rm=TRUE)
# MAE = mean absolute error (mm/d)
MAE <- mean(abs(simu - obse), na.rm=TRUE)
# MALE = mean absolute log error (mm/d)
MALE <- exp(mean(abs(log(simu) - log(obse))), na.rm=TRUE))
# VE = volume error (%/%)
VE <- (sum(simu[!is.na(obse)]) - sum(obse, na.rm=TRUE))/sum(obse, na.rm=TRUE)

output <- c(RMSE, NE, lNE, B, MAE, MALE, VE)
names(output) <- c("RMSE (mm/d)", "Nash efficiency ()", "log Nash efficiency
()", "bias (mm/d)",
      "mean absolute error (mm/d)", "mean absolute log error
(mm/d)", "volume error (%/%)")
return(output)
}

# The following code returns a matrix with the efficiencies
efficiencies <-      rbind(EMS(3.6*24*as.numeric(simulation1_cal1$q)/area,
3.6*24*as.numeric(Q1)/area),
      EMS(3.6*24*as.numeric(simulation2_cal2$q)/area,
3.6*24*as.numeric(Q2)/area),
      EMS(3.6*24*as.numeric(simulation2_val1$q)/area,
3.6*24*as.numeric(Q1)/area),
      EMS(3.6*24*as.numeric(simulation1_val2$q)/area,
3.6*24*as.numeric(Q2)/area))
rownames(efficiencies) <- c("cal1", "cal2", "val1", "val2")
print(signif(efficiencies, 3))

# The following code writes a file 'param' with the optimised parameters
write.table(t(efficiencies), file=paste("eff", tt, "_", datafile, sep=""))
# which produces the attached file 'eff201402121216_at4.asc'

# The following function plots observed vs. simulated runoff as scatterplot
scatterplot <- function (sim, obs, warmup=365, xlab="Observations",
ylab="Simulations", ...) {
# obs = observed runoff (class numeric)
# sim = simulated runoff (class numeric)
# warmup = warm-up period in d
simu <- sim[-c(1:warmup)]
obse <- obs[-c(1:warmup)]

```

```

plot(obse, simu, xlab=xlab, ylab=ylab, ...)
abline(0, 1, lty=2)
}

# The following code does the scatterplots
scatterplot(simulation1_cal1$q, Q1)
scatterplot(simulation1_cal1$q, Q1, log="xy", cex=.5)

limits <- range(c(range(Q, na.rm=TRUE), range(simulation1_cal1$q,
simulation2_cal2$q, simulation2_val1$q, simulation1_val2$q)))
par(mar=c(3,3,1,1)+0.03, mgp=c(1.1,0.1,0), tcl=.2, xaxs="r", yaxs="r", las=0)
layout(matrix(1:4, nrow=2, byrow=TRUE))
scatterplot(simulation1_cal1$q, Q1, log="xy", cex=.5, xlim=limits,
ylim=limits,
xlab="Obs period 1 (m3/s)", ylab="Sim (calibrated on period 1)")
scatterplot(simulation2_cal2$q, Q2, log="xy", cex=.5, xlim=limits,
ylim=limits,
xlab="Obs period 2 (m3/s)", ylab="Sim (calibrated on period 2)")
scatterplot(simulation2_val1$q, Q1, log="xy", cex=.5, xlim=limits,
ylim=limits,
xlab="Obs period 1 (m3/s)", ylab="Sim (calibrated on period 2)")
scatterplot(simulation1_val2$q, Q2, log="xy", cex=.5, xlim=limits,
ylim=limits,
xlab="Obs period 2 (m3/s)", ylab="Sim (calibrated on period 1)")

# The following function plots observed vs. simulated flow duration curves
flowdurations <- function (sim, obs, warmup=365, xlab="F(x > X)", ylab="x",
...) {
# obs = observed runoff (class numeric)
# sim = simulated runoff (class numeric)
# warmup = warm-up period in d
simu <- sim[-c(1:warmup)]
obse <- obs[-c(1:warmup)]

quantili <- c(1,.999,.995, seq(.99,.01, by=-.01), .005,.001,0)
obs_fdc <- quantile(obse, prob=quantili, na.rm=TRUE)
sim_fdc <- quantile(simu, prob=quantili, na.rm=TRUE)

plot(1 - quantili, obs_fdc, type="l", xlab=xlab, ylab=ylab, ...)
lines(1 - quantili, sim_fdc, col="red")
}

par(mar=c(3,3,1,1)+0.03, mgp=c(1.1,0.1,0), tcl=.2, xaxs="r", yaxs="r", las=0)
layout(matrix(1:4, nrow=2, byrow=TRUE))
flowdurations(simulation1_cal1$q, Q1, log="y", ylim=limits, ylab="Cal period
1")
flowdurations(simulation2_cal2$q, Q2, log="y", ylim=limits, ylab="Cal period
2")
flowdurations(simulation2_val1$q, Q1, log="y", ylim=limits, ylab="Val period
1")
flowdurations(simulation1_val2$q, Q2, log="y", ylim=limits, ylab="Val period
2")

# The following code writes a file 'sim' with the simulations for the two
parametrisations
simulation_all_cal1 <- TUWmodel(prec=as.numeric(P), airt=as.numeric(T),
ep=as.numeric(EP), area=area,
param=calibrate_period1$optim$bestmem)
simulation_all_cal2 <- TUWmodel(prec=as.numeric(P), airt=as.numeric(T),
ep=as.numeric(EP), area=area,
param=calibrate_period2$optim$bestmem)
output <- data.frame(data, as.numeric(simulation_all_cal1$q),
as.numeric(simulation_all_cal2$q))
names(output) <- c("day", "month", "year", "P", "T", "Q", "simQcal1",
"simQcal2")
write.table(output, file=paste("sim", tt, "_", datafile, sep=""),
row.names=FALSE)
# which produces the attached file 'sim201402121216_at4.asc'

```

```

# Optionally, some plots of simulated vs. observed streamflow can be produced
# in R using the function in the file 'plot_comparison.R':
# > source("plot_comparison.R") # must be in the current directory
# > pdf(file=paste("check", tt, "_", sub(".asc", "_cal1", datafile), ".pdf",
# sep=""), height=7.6, width=6.4, pointsize=9)
# > par(mar=c(3.5,2.5,1.2,1)+0.03, mgp=c(1.1,0.1,0), tcl=.2, xaxs="r",
# yaxs="r", las=0)
# > layout(matrix(c(1,1,2,2:8), ncol=2, byrow=TRUE))
# > plot_comparison(3.6*24*Q/area,
# zoo(3.6*24*as.numeric(simulation_all_cal1$q)/area, time(Q)))
# > dev.off()
# > pdf(file=paste("check", tt, "_", sub(".asc", "_cal2", datafile), ".pdf",
# sep=""), height=7.6, width=6.4, pointsize=9)
# > par(mar=c(3.5,2.5,1.2,1)+0.03, mgp=c(1.1,0.1,0), tcl=.2, xaxs="r",
# yaxs="r", las=0)
# > layout(matrix(c(1,1,2,2:8), ncol=2, byrow=TRUE))
# > plot_comparison(3.6*24*Q/area,
# zoo(3.6*24*as.numeric(simulation_all_cal2$q)/area, time(Q)))
# > dev.off()
# which produces the attached file 'check201402111908_at4.pdf'

# ----- #

# References

# Nash, J. and Sutcliffe, J. (1970). River flow forecasting through conceptual
# models part I - A discussion of principles, Journal of
# Hydrology, 10(3):282-290, doi:10.1016/0022-1694(70)90255-6.
#
# Parajka, J., Merz, R., and Bloeschl, G. (2003). Estimation of daily potential
# evapotranspiration for regional water balance modeling
# in Austria. In Transport of Water, Chemicals and Energy in the Soil - Crop
# Canopy - Atmosphere System, 11th International Poster Day
# and Institute of Hydrology Open Day, pages 299-306.
#
# Parajka, J., Merz, R., and Bloeschl, G. (2007). Uncertainty and multiple
# objective calibration in regional water balance modelling:
# case study in 320 Austrian catchments. Hydrological Processes, 21(4):435-
# 446, doi:10.1002/hyp.6253.
#
# Schroedter, H. (1985). Verdunstung - Anwendungsorientierte Messverfahren und
# Bestimmungsmethoden, Springer, 186 pages, ISBN:
# 3540153551, 9783540153559.

```